密码科学技术丛书

公钥加密的设计方法

作者：陈宇 & 秦宝东

组织：山东大学 & 西安邮电大学

版本：0.1

微言大义 高屋建瓴

前言

本书是在 “2017-2019 中国科学院大学研究生暑期课程” 和 “2022 北京大学应用数学专题讲习班” 的讲义基础上, 结合多年在公钥加密方面的研究成果编写而成. 目的是尽快引导读者到达公钥加密这一极为重要的现代密码学领域, 力求从高屋建瓴的视角介绍公钥加密的设计方法, 对重要的思想剥丝抽茧、对关键的技术条分缕析. 写作过程中根据教学和研究经历感悟对部分内容进行了精简和重构, 参阅了国际顶级会议期刊的前沿论文, 也融入了作者的独立思考. 陈宇编写了本书的第1、2、3、4和7章, 秦宝东编写了本书的第5、6章.

计算机网络技术的飞速发展引发了人类社会组织形态的根本性变革, 从集中式迁移为分布式, “海内存知己, 天涯若比邻” 的诗歌意象成为现实世界. 面向分布式环境下的隐私保护需求, 1976 年 Diffie 和 Hellman 开创了现代密码学的新方向—公钥密码学, 自此以后的半个多世纪, 公钥密码学一直处于最活跃的前沿, 引领驱动了密码学的研究进展, 极大丰富了密码学的学科内涵. 公钥加密作为公钥密码学最重要的分支, 在理论方面孕育了可证明安全方法、引入了各类数学困难问题作为安全基础、启发了一系列密码原语和重要概念, 已有多项历史性成果获得 Turing 奖和 Gödel 奖; 在应用方面成为各类网络通信安全协议的核心密码组件, 时刻保护着公开信道上信息传输的机密性.

当前, 公钥加密仍处于快速发展阶段, 在安全性方面, 各类超越传统语义安全的高级安全属性研究已经日趋成熟, 基于复杂性弱假设的细粒度安全的研究正在兴起; 在功能性方面, 函数加密的研究方兴未艾, 全同态加密的研究如火如荼. 我们已经有幸见证了公钥加密之旅的美妙风景, 但还有更广袤深邃的领域待探索征服.

公钥加密历经多年发展, 各类方案层出不穷, 概念定义繁多复杂, 因此想深入学习的读者往往会感觉陷入书山文海, 难识庐山真面目. 本书试图指引读者快速登高俯瞰, 将公钥加密的设计方法尽收眼底, 达到万变不离其宗的认知. 为此, 本书的内容偏重基于一般假设的通用构造. 此般选择有诸多好处, 从理论角度, 通用构造剥除了旁枝细节、凸显了核心要素, 从而更容易洞察研究对象的本质; 从实际角度, 通用构造能够启发更多的具体方案, 敏捷满足各类安全和应用需求.

本书的第1章简述了公钥加密的发展历程, 第2章介绍了准备知识, 为后续章节做好铺垫. 第3章回顾经典的公钥加密方案, 帮助读者先获得具象的感性认识, 为理解抽象的通用构造积累一些重要的例子. 第4章是核心部分, 展示如何从各类密码组件出发构造公钥加密, 并在最后获得更高阶的抽象, 与对称加密的构造相互呼应、完美契合. 第5章和第6章分别从安全性增强和功能性扩展两个维度介绍公钥加密的重要成果和前沿进展. 第7章简介了公钥加密的标准化与工程实践, 打通理论与实践的最后一公里.

书中很多看似不起眼的注记恰恰是点睛之笔, 它们大多来源于作者科研过程中的心得体会, 期待读者在领悟其中蕴含的思辨方式之后能会心一笑, 见到更美的风景. 总的来说, 切实掌握本书的内容之后, 可使读者熟练掌握公钥加密的可证明安全技术、养成抽象思维习惯, 为进一步的研究打好基础.

密码科学技术国家实验室对本书的出版给予了极大支持, 作者深表感谢. 作者也借此机会感谢清华大学王小云院士、中国科学院林东岱研究员、邓燚研究员和上海交通大学的郁昱教授、刘胜利教授多年以来给予的鼓励和指点. 最后, 感谢家人们的默默支持, 没有你们的全情支持, 我们不可能完成此书.

由于水平有限, 时间紧迫, 定有许多不当之处. 诚恳欢迎批评指正.

陈宇 & 秦宝东

2023 年夏

目录

第一章 公钥加密的概述

1.1 背景与起源

密码几乎与文字一样古老, 并随着时代的变革、技术的进步而持续不断地发展, 其发展历程大致可以分为以下两个阶段:

- 古典阶段: 该阶段密码的内涵局限于加解密, 主要应用于军事行动中的保密通信. 在古典密码早期, 加密方案的安全性依赖于对方案本身的保密, 代表性的方案有公元 110 年前后出现的 Caesar 密码, 其本质是将字母循环后移三位进行简单的单表代换加密. 在古典密码中期, 加密方案的安全性摆脱对于系统保密性的依赖, 转向对密钥的保密, 代表性的方案是 1586 年出现的多表代换加密的 Vigènre 密码, 其不断重复密钥并与明文进行相加. 在古典密码后期, Caesar、Vigènre 等密码中蕴含的代换/置换设计思想得到进一步发展, 集大成者是一战时期出现的 Enigma 密码, 其代换规则是动态的, 每加密一个字母的消息后, 代换规则随内部诸多转子的位置变化动态的改变, 从而使得代换、置换关系更为复杂. 总的来说, 古典阶段的密码缺少系统的设计与分析方法, 更类似于一种 “艺术” 而非 “科学”, 一旦加密方案暴露后, 容易受到密码分析而被攻破, 陷入 “攻破-修复” 的循环.
- 现代阶段: 随着信息化进程的加速, 密码逐渐从军用扩展到商用和民用. 1945 年, Shannon 在著名论文《密码学的数学理论》[1] 首次使用概率论的方法对密码系统的安全性进行了精准的刻画, 奠定了现代密码学的基础, 使得密码学从 “艺术” 逐渐走向 “科学”. 1973 年, Feistel [2] 提出了赫赫有名的 Feistel 网络, 催生出对称密码标准 (data encryption standard, DES), 对称密码的设计与分析逐渐系统化. 20 世纪 70 年代后, 计算设备的算力遵循摩尔定律持续提升、计算环境由集中式迁移到分布式, 信息技术的迅猛发展对密码学提出了新的挑战. 1976 年, Diffie 与 Hellman 在划时代的论文《密码学的新方向》 [3] 中将非对称的思想引入密码学, 开创了公钥密码学. 1978 年, Rivest, Shamir 和 Adelman [4] 基于数论中的困难问题设计出首个公钥加密方案和数字签名方案. Goldwasser 和 Micali [5] 给出了公钥加密的合理安全性定义——语义安全, 同时开创了可证明安全的方法 (即使用计算复杂性理论中的归约技术将密码方案的安全性严格归结为计算困难问题的复杂性), 基于二次剩余假设构造出了首个满足语义安全的概率公钥加密方案–Goldwasser-Micali PKE. 此后, 密码学蓬勃发展, 内涵不断丰富, 从加密、签名和密钥交换扩展到零知识证明 [6]、安全多方计算 [7] 等. 总的来说, 区别于古典阶段密码设计的随意, 现代密码学与复杂性理论结合紧密, 具备严格的可证明安全, 各类密码组件/方案与困难问题之间以归约为桥梁, 形成精密的归约网络.

我们再把视线聚焦回公钥加密. 21 世纪信息化技术进一步高速发展, 面对多样的应用需求和更强的攻击手段, 函数加密、全同态加密等高功能加密方案和抗泄漏、抗篡改等高等级安全相继出现 [8]. 以下, 本文将按照安全性增强和功能性扩展这两条并行的线索对公钥加密的发展历程和前沿进展做系统的概述. 在密码学的发展历程中, 当具有新的功能性或安全性的方案被提出后, 密码学家通常先给出若干具体方案, 然后随着研究的深入再抽象出更基本的密码原语, 进而以它们为底层组件给出通用构造. 通用构造不仅能够总结一般的方法对既有方案进行分类, 还能凝练关键的思想从而加深对密码方案本质的认识、启发新的构造, 因此极具价值. 为了引领读者从更高的观点回顾公钥加密, 本概述将注重对通用构造的介绍, 其中所涉及的底层密码原语将有助于读者拓展知识面、加深理解认知.

1.2 公钥密码学发展的两条主线

自 Diffie 和 Hellman 的划时代论文 [3] 后, 公钥密码学的发展一日千里、日新月异, 热潮持续至今, 始终是现代密码学的核心和重要技术的摇篮.

公钥密码的发展有两条主线: 一条是安全性的增强, 从最初的直觉安全演进到严格健壮的语义安全, 再到不可区分选择密文安全和各类超越传统安全模型的高级安全性, 如选择打开攻击安全、抗泄漏安全、抗篡改安全和消息依赖密钥安全; 另一条是功能性的丰富, 从最初不具有密钥委派功能的一对一加解密到能委派身份密钥的身份加密, 再到具有细粒度访问控制功能的属性加密乃至极致泛化的函数加密以及对可对密文进行公开计算的 (全) 同态加密.

本章, 我们将从这两条主线出发, 梳理公钥密码学的发展历程, 概述重要研究成果.

1.3 安全性增强

对于密码方案, 给出恰当的安全性定义至关重要, 安全性定义必须足够强以刻画现实中存在的攻击. 但是通常很难直接构造满足强安全性的密码方案, 强安全性的公钥加密方案的构造往往建立在弱安全性的公钥加密方案之上. 对应到公钥加密的发展历程, 公钥加密的安全性定义亦是从弱到强逐渐演化的.

1982 年, Goldwasser 和 Micali [5] 发展了可证明安全的技术框架: 建立安全模型以准确刻画敌手的攻击行为和攻击效果, 将密码方案的安全性归约到计算困难问题的困难性上. Goldwasser 和 Micali 指出单向安全并不是加密方案的合理安全性, 他们提出了语义安全性 (semantic security) 以及等价的另一个定义–选择明文攻击下的不可区分性 (indistinguishability against chosen-plaintext attack, IND-CPA). IND-CPA 安全要求密文在计算意义下不泄漏明文的任何一比特信息. 相比基于模拟方式定义的语义安全, IND-CPA 安全基于安全游戏定义, 在归约证明中更容易使用, 因此应用广泛. 随着密码分析技术的发展, 研究人员发现 IND-CPA 安全无法应对更强的攻击手段, 增强的安全性被陆续被提出. 以下简介公钥加密的几种重要的增强安全性质.

选择密文安全. IND-CPA 安全仅考虑被动敌手, 即敌手只窃听信道上的密文. 1990 年, Naor 和 Yung [9] 指出敌手有能力发起一系列主动攻击, 比如重放密文、修改密文等, 进而提出选择密文攻击 (chosen-ciphertext attack, CCA) 刻画这一系列主动攻击行为, 即敌手可以自适应地获取指定密文对应的明文. 在 CCA 刻画的攻击行为下的安全性 (在不可区分实验下) 可对应到选择密文攻击下的不可区分性 (indistinguishability against chosen-ciphertext attack, IND-CCA), 即 IND-CCA 安全. 1998 年, Bleichenbacher [10] 展示了针对 PKCS#1 标准中公钥加密方案的有效选择密文攻击, 实证了关于 IND-CCA 安全的必要性. 从此, IND-CCA 安全成为了公钥加密方案的事实标准. 获得 IND-CCA 安全的公钥加密方案有以下主流路线:

- 随机谕言机模型下:
  - 基于单向陷门置换构造 IND-CCA 安全的 PKE. Bellare 等 [11] 提出了随机谕言机模型 (RO, Random Oracle), 而后在文献 [12] 提出了一种基于单向陷门置换构造 IND-CCA 安全的 PKE 方案的通用转换, 即 OAEP 通用转换. Shoup [13] 指出 OAEP 转换并不是对所有的单向陷门置换成立. Fujisaki 等 [14] 则进一步指出了 OAEP 转换应该满足的条件, 并提出了基于 RSA 类型单向陷门置换的 RSA-OAEP 方案.
  - 基于 IND-CPA 安全的 PKE 构造 IND-CCA 安全的 PKE. Fujisaki 和 Okamoto [15] 提出了在 RO 模型下利用混合加密将 IND-CPA 安全的 PKE 方案提升为 IND-CCA 安全的 PKE 方案的通用转换, 其核心思想是引入哈希函数绑定密文的各部分使得密文不可延展. Fujisaki 和 Okamoto 随后 [16] 进一步弱化了底层 PKE 方案的安全性, 将其从 IND-CPA 安全减弱为单向安全, 这一通用转换也被称为 FO 转换 [16, 17]. 类似的通用转换还有许多, 但就蕴含的思想而言, 其核心思想均与 FO 转换相似, 如 Okamoto 和 Pointcheval [18] 提出的 REACT 通用转换. 后续工作着力于提升 FO 转换的效率 [19, 20, 21] 及扩展应用范围 [22].
- 标准模型下:

- 使用非交互零知识证明将任意 IND-CPA 安全的 PKE 强化为 IND-CCA 安全. Naor 和 Yung [9] 首次提出双密钥加密策略结合非交互零知识证明 (non-interactive zero-knowledge proof, NIZK) 将任意 IND-CPA 安全的 PKE 方案强化为 IND-CCA1 安全; Dolev, Dwork 和 Naor [23] 设计出矩阵式加密结构, 结合一次性不可伪造签名 (one-time signature, OTS) 和 NIZK 将任意 IND-CPA 安全的 PKE 方案提升至 IND-CCA 安全; 受上述两个工作启发, Sahai [24] 展示如何用 OTS 将普通 NIZK 强化为具有模拟稳固 (simulation soundness, SS) 性质的 NIZK, 以此为工具应用 Naor-Yung 双重加密范式, 将任意 IND-CPA 安全的 PKE 方案强化为 IND-CCA 安全. Biagioni 等 [25] 和 Cramer 等 [26] 分别给出了 Naor-Yung 双重加密范式的两个变体, 前者对加密随机数进行了安全的重用, 后者使用了新的加密重复策略和消息一致性证明方法.
- 基于弱化的非交互证明系统构造 IND-CCA 安全的 PKE. Cramer 和 Shoup [27] 提出了一种特殊的指定验证者非交互式零知识证明 (designated-verifier NIZK, DV-NIZK)—- 哈希证明系统 (hash proof system, HPS), 并基于 HPS 给出了一类基于判定性困难问题的 IND-CCA 安全的 PKE 方案的通用构造. Wee [28] 提出了一种特殊的指定验证者非交互式零知识的知识证明—-可提取哈希证明系统 (extractable HPS, EHPS), 并基于 EHPS 给出了一类基于搜索性困难问题的 IND-CCA 安全的 PKE 方案的通用构造.
- 基于结构更丰富的加密方案构造 IND-CCA 安全的 PKE. Boneh 等 [29] 以身份加密 (identity-based encryption, IBE) 为底层方案, 结合 OTS 或消息验证码 (MAC, Message Authentication Code) 构造出 IND-CCA 安全的 PKE 方案. Kiltz [30] 将 IBE 弱化为基于标签的加密 (tag-based encryption, TBE), 结合 OTS 构造出 IND-CCA 安全的 PKE 方案.
- 基于更强的单向陷门函数. Peikert 等 [31] 提出了有损陷门函数及其构造 IND-CCA 安全的 PKE 的方法. 后续工作进一步减弱了有损陷门函数的结构性质 [32, 33]. 最近的突破性结果是 Hohenberger 等 [34] 展示了如何仅基于单射陷门函数构造 IND-CCA 安全的 PKE.
- 基于 $i\mathcal{O}$ 的编译器. Sahai 和 Waters [35] 引入了可穿孔伪随机函数, 借助不可区分混淆 (indistinguishability obfuscation, $i\mathcal{O}$) 构造出 IND-CCA 安全的 PKE 方案. 该构造实现了 Diffie 和 Hellman 自 1970s 起的愿景, 展示了如何将对称加密编译为公钥加密.
- 基于可公开求值伪随机函数. Chen 等 [36] 提出了可公开求值伪随机函数, 并基于此给出了 PKE 方案的通用构造. 该构造不仅从更抽象的层次阐释了经典的 GM 加密方案 [5] 和 ElGamal 加密方案 [37] 的设计思想, 还统一了基于 (可提取) 哈希证明系统、单向陷门函数、可穿孔伪随机函数与 $i\mathcal{O}$ 的构造.

选择打开安全. IND-CPA 安全模型和 IND-CCA 安全模型主要考虑一对一的两方通信场景, 但是 PKE 方案常常被应用到多对多的分布式协议当中. 例如, $n-1$ 个发送者用某用户的公钥对某轮协议中 (相关联) 的消息进行加密, 并发送给该用户. 此时存在一个强力的敌手, 其腐化了该轮协议一部分消息发送方的计算机, 不仅获得了他们密文对应的明文, 还获得了加密用的随机数. 类似于两方通信下的 PKE 方案的语义安全, 仍然希望敌手不会获得除了条件分布以外的关于未腐化的用户的明文的有效知识. Bellare 等 [38] 首先考虑了上述的情景, 将这样的安全性称为选择打开 (selective opening, SO) 安全, 并从模拟和不可区分两个角度分别给出了 SO 安全的 SIM-SO 和 IND-SO 两种形式化定义. 同时, 他们也提出了有损加密 (lossy encryption) 这一新概念. 简单来说, 有损加密有两种加密模式, 单射模式下利用私钥可以解密密文得到明文, 而有损模式下则是多个密文对应到一个明文, 从而无法正确解密, 他们也证明了有损加密方案本身为 IND-SO-CPA 安全的, 而具有某种高效打开算法的有损加密方案本身为 SIM-SO-CPA 安全的. 但是上述的场景——多用户根据同一个公钥加密消息发送给某一用户 (敌手腐化发送方取得明文与加密随机数) 与真实的多方协议相去甚远. 因此, Bellare 等 [39] 提出了一个新的场景——某一用户根据多个用户不同的公钥加密发送不同的加密消息给不同的用户 (敌手腐化接收方取得私钥). 为了区分, 他们称第一种场景下的安全性为发送者 SO(sender SO, SSO) 安全, 第二种场景下的安全性为接受者 SO(receiver SO, RSO) 安全. SO 安全的主要研究工作主要集中在如何构造 SSO 安全和 RSO 安全的 PKE 方案上.

- SSO 安全的 PKE 的构造: Bellare 等 [38] 利用有损加密给出了 SSO 安全的 PKE 方案的构造, 后续一系列研究延续 Bellare 等的构造框架, 围绕着有损加密方案的构造展开. Hemenway 等 [40] 从多种密码原语出发 (例如同态加密、哈希证明系统等) 给出了有损加密的多种构造方法, 结合 All-But-N 有损陷门函数进一步构造

出了 IND-SSO-CCA 安全的 PKE 方案. Hofheinz [41] 将 All-But-N 有损陷门函数拓展为 All-But-Many 有损陷门函数, 构造出了 SIM-SO-CCA 安全的 PKE 方案. Fehr 等 [42] 则采取了另外一条有别于 Bellare 的路径, 他们先证明了发送者模糊性 (sender equivocable, NC)CCA 安全蕴含了 SIM-SSO-CCA 安全, 然后给出一种基于哈希证明系统和交叉验证码构造 NC-CCA 安全的 PKE 方案的通用方法, 从而构造出 SIM-SSO-CCA 安全的 PKE 方案. Huang 等 [43] 指出了 Fehr 等 [42] 的证明中的问题, 并给出一个可行的修正方法——将交叉验证码的性质加强. 格上的 SSO 安全的 PKE 方案的构造主要基于格基 All-But-Many 有损陷门函数 [44, 45].

- RSO 安全的 PKE 的构造: Hazay 等 [46] 首先利用接受者非承诺加密 (non-committing encryption for receiver, NCER) 构造了 SIM-RSO-CPA 安全的 PKE 方案, 并用 NCER 的变体构造了 IND-RSO-CPA 的 PKE 方案. Jia 等 [47] 基于 Naor-Yung 双重加密范式给出了 IND-RSO-CCA 安全的 PKE 的通用构造, 该构造将 IND-RSO-CPA 安全的 PKE 方案和 CCA 安全的 PKE 方案相结合, 并通过 NIZK 将其转换为 IND-RSO-CCA 安全的 PKE 方案. Hara 等 [48] 将类似的方法推广到了 SIM-RSO-CCA 安全的 PKE 方案的构造中, 并额外基于 DDH 假设给出了一个较为高效的具体构造.

孤立的 SSO 安全或者是 RSO 安全仍然与现实多对多分布式协议的安全性要求并不十分吻合, 后续有些研究开始考虑更为广泛意义下的 SO 安全性. 例如, Lai 等 [49] 首次在一个模型下同时考虑 SSO 和 RSO 两种安全性, 形式化的提出了基于模拟的 SIM-(w)Bi-SO-CCA 安全性, 并基于密钥模糊性哈希证明系统这一新的密码原语给出了 SIM-wBi-SO-CCA 安全的 PKE 方案的通用构造. 这些研究成果使得 SO 安全的 PKE 方案可以有效的保障多对多分布式协议的安全.

另外, 除了关于 SO 安全的 PKE 方案具体构造的研究外, SO 安全性与其他安全性间的强弱联系等理论问题也得到了深入的研究. 例如, Bellare 等 [38, 50] 证明了 SIM-SSO-CPA 安全性在黑盒意义上蕴涵 IND-SSO-CPA 安全性. 又例如, Bellare 等 [39] 证明了标准的 IND-CPA 安全性不蕴涵 SIM-SSO-CPA 安全性等.

消息依赖密钥安全. IND-CPA 安全模型下敌手可以获得消息对应的密文, 而 IND-CCA 安全模型下敌手还可以额外获得选定密文对应的明文, 除此之外, 敌手不能获得其他与 sk 有关的信息. 然而传统的 IND 安全无法全面覆盖复杂的实际应用场景. 以 Windows 操作系统下最为知名的文件加密程序 BitLocker [51] 为例, 其不仅将保密文件经过加密放在硬盘上, 还将相应的密钥对加密存储在硬盘上, 从而使得敌手不仅可以获得机密文件对应的密文, 还获得了私钥的密文, 类似的场景还有匿名凭证 [52] 等. 此类攻击显然无法被 IND-CPA 安全和 IND-CCA 安全所刻画, 为了刻画敌手可以获得私钥密文的能力, Black 等 [53] 首先定义了消息依赖密钥 (key-dependent message, KDM) 安全, 要求即使敌手获得与密钥相关的密文, 加密方案依然安全. 正式的, 令 $\{(sk_i, pk_i)\}_{i\in[n]}$ 为 n 组密钥对, $\mathcal{F}$ 是定义在 SK^n 上的密钥相关函数集, 如果敌手可以任意选择 $f \in \mathcal{F}$ 并观察到 $f(sk_1, \cdots, sk_n)$ 在所有 pk_i 下加密的密文后仍然无法打破 PKE 方案的语义安全, 那么就称该 PKE 方案为 $\mathcal{F}$-KDM 安全的. 显然, KDM 安全由密钥函数集 $\mathcal{F}$ 刻画, $\mathcal{F}$ 越大, 对应的 $\mathcal{F}$-KDM 安全性越强. 因此, 消息依赖密钥安全研究的一个核心问题就在于如何扩大密钥函数集 $\mathcal{F}$.

Boneh 等 [51] 基于 DDH 假设构造出首个 KDM-CPA 安全的 PKE 方案, 其支持的密钥相关函数为仿射函数 $\mathcal{F}_{\mathsf{aff}}$. Applebaum 等 [54] 给出基于 LWE 假设构造出另一个 $\mathcal{F}_{\mathsf{aff}}$ 的 $\mathcal{F}_{\mathsf{aff}}$-KDM-CPA 安全的 PKE 方案. 之后, Malkin 等 [55] 基于 DCR 假设给出了一个相关密钥函数集为次数有界多项式函数集 $\mathcal{F}_{\mathsf{poly}}$ 的 KDM-CPA 安全的 PKE 方案. Wee [56] 基于同态的哈希证明系统给出了一个通用的 KDM-CPA 安全的 PKE 方案设计框架, 统一了诸多已有的具体构造 [51, 57].

后续工作致力于构造更强的 KDM-CCA 安全的 PKE 方案, Camenisch 等 [58] 指出 Naor-Yung 双重加密范式可将 $\mathcal{F}$-KDM-CPA 安全的 PKE 方案强化为 $\mathcal{F}$-KDM-CCA 安全的 PKE 方案. Han 等 [59] 提出了一个密文紧致的 $\mathcal{F}_{\mathsf{aff}}$-KDM-CCA 安全的 PKE 方案, 并进一步拓展为 $\mathcal{F}_{\mathsf{poly}}$-KDM-CCA 安全的 PKE 方案. Kitagawa 等 [60] 基于 DDH 假设构造出了 $\mathcal{F}_{\mathsf{aff}}$-KDM-CCA 安全的 PKE 方案, 密钥尺寸为 $O(\kappa)$ 数量级. Kitagawa 等 [61] 提出了对称密钥封装这一新的密码原语, 并利用此原语改进之前的方案, 得到了比 Han 等 [59] 更为高效 $\mathcal{F}_{\mathsf{aff}}$-KDM-CCA 安全的 PKE 方案.

类似于 SO 安全的研究, 对 KDM 安全的研究除了关于 KDM 安全的 PKE 方案的具体构造, KDM 安全的相对强度以及与其他安全性之间的联系等理论问题也得到了深入的研究. 例如, Kitagawa 等 [62] 中证明了 KDM-CPA

安全和 KDM-CCA 安全在黑盒意义上是等价的; Waters 等 [63] 证明了对于 PKE 方案, 单密钥循环安全在黑盒意义上蕴含了相关函数集为任意有限深度电路的 KDM 安全等.

抗泄漏安全. 2004 年, Micali 和 Reyzin [64] 提出了物理可观测安全密码学, 拉开了从理论上防御泄漏攻击的研究序幕. 2008 年, Dzimbowski 和 Pietrzak [65] 提出了泄漏模型, 开启抗泄漏密码学的系统研究. 简言之, 泄漏模型通过增加泄漏谕言机 $\mathcal{O}_{\text{leak}}(\cdot)$ 强化传统安全模型. 敌手可以自适应的向泄漏谕言机发起一系列泄漏询问 $f_i : sk \to \{0,1\}^{\ell_i}$ 并获得相应的结果, 其中 f_i 是关于私钥的函数, 称为泄漏函数. 不同的泄漏模型通过对泄漏函数施加不同的限制获得. Micali 和 Reyzin [64] 首先给出唯计算泄漏模型, 即要求泄漏函数的输入只能是私钥参与计算的部分. 然而冷启动攻击表明该模型无法全面刻画现实攻击, 未参与计算的存储单元同样可能发生泄漏. 之后出现的模型对发生泄漏的存储单元不再加以限制, 即所有的存储单元均可能存在泄漏. 其又可进一步分为相对泄漏模型 (relative memory leakage model) 和连续泄漏模型 (continual memory leakage model), 前者对密码系统在整个生命周期内的泄漏量存在一定限制; 后者考虑密码系统可进行连续更新, 只对状态更新之间的泄漏量存在一定限制, 而对整个生命周期内的泄漏量没有限制.

- 相对泄漏模型中, 攻击者能够通过访问泄漏函数获取关于私钥的信息. 对泄漏函数施加的细粒度限制进一步细分了此类模型. Akavia 等 [66] 提出的有界泄漏模型 (bounded leakage model) 限制 $\sum \ell_i < \ell$, 其中 ℓ_i 为敌手自适应选择的泄漏询问 f_i 的输出长度, ℓ 为泄漏量上界, 泄漏率被定义为比值 $\ell/|sk|$, 用于衡量泄漏量的多少. Akavia 等在有界泄漏模型下基于 LWE 假设给出了抗泄漏安全的公钥加密方案和身份加密方案. 随后, Naor 和 Segev [67] 基于哈希证明系统给出了有界泄漏模型下抗泄漏公钥加密的通用构造, 并给出多个实例化方案, 其中基于 Cramer-Shoup 加密方案变体的方案泄漏率为 $1/6 - o(1)$. 此外, Naor 和 Segev [67] 还将有界泄漏模型放宽为噪声泄漏模型 (noisy leakage model), 该模型不再对泄漏函数输出长度总和进行限制, 仅要求泄漏后私钥的最小熵仍大于预设的下界. Dodis 等 [68] 猜想完全基于哈希证明系统的选择密文安全构造泄漏率不可能大于 $1/2 - o(1)$. 后续的工作致力于提升抗泄漏选择密文安全公钥加密方案的泄漏率. Liu 等 [69] 基于 Cramer-Shoup 加密方案的新变体给出了泄漏率为 $1/4 - o(1)$ 的构造. Qin 等 [70] 结合哈希证明系统和一次有损过滤器给出了选择密文安全公钥加密方案的新构造, 并在 [71] 中通过精心的实例化达到了最优泄漏率 $1 - o(1)$. Chen 等 [72] 提出正则有损函数 (regular lossy function, RLF), 以此为工具证实了仅基于 HPS 即可构造出具有最优泄漏率的 IND-CCA 安全的 PKE 方案, 否证了 Dodis 等人 [68] 的猜想. Chen 等 [73] 综合使用 $i\mathcal{O}$ 和有损函数开辟了构造最优泄漏率的抗泄漏 PKE 方案的非黑盒新路线.
  有界泄漏模型和噪声泄漏模型的泄漏容忍上界正比于私钥长度, 泄漏上界的提升将导致密码系统私钥长度的增加和运行效率的下降. 为了在泄漏上界极大 (如 $O(2^{\lambda})$) 的情况下仍具有较好的运行效率, Alwen 等 [74] 提出有界获取模型 (bounded retrieval model). 该模型允许敌手获取关于私钥高达 $\ell = O(2^{\lambda})$ 比特量级的信息, 同时要求密码算法运行时间与泄漏量 ℓ 无关, 仍为关于安全参数 ℓ 的多项式. Alwen 等 [75] 利用身份哈希证明系统基于不同困难假设构造了一系列有界获取模型下抗泄漏的身份加密方案.
  Dodis 等 [76, 77] 观察到上述相对泄漏模型的共同点是要求泄漏后私钥仍有一定的统计意义下的最小熵. 他们将统计意义进一步放宽到计算意义, 得到了更具一般性的辅助输入模型, 并在该模型下构造了抗泄漏的 PKE 方案.
- 持续泄漏模型: 该类模型中针对密钥具有连续更新机制的密码系统定义. 攻击者每个更新周期内都可以获得任意内存单元的泄漏. 在该模型下, Brakerski 等 [78] 利用线性代数的技巧, 基于双线性群中的线性假设构造出抗泄漏的 PKE 方案和 IBE 方案. Lewko 等 [79] 利用双系统加密技术, 给出了的更强意义下的抗泄漏的 (H)IBE 方案和 ABE 方案, 其中泄漏不仅可以针对一个身份/属性的多个私钥发生, 还可针对主私钥发生. Lewko 等 [80] 基于合数阶群中的判定性假设, 给出抗泄漏的 PKE 方案, 且允许更新过程中的泄漏量超过指数级别. Yuen 等 [81] 将连续内存泄漏模型和辅助输入模型结合, 利用双系统加密技术给出了抗泄漏的 (H)IBE 方案. 以上持续泄漏模型的局限是仅允许泄漏发生在每个私钥生命周期中, 不允许在私钥更新过程中发生泄漏. Dachman-Soled 等 [82] 利用 $i\mathcal{O}$ 构造了一个编译器, 可对持续泄漏模型下的公钥加密方案进行强化, 容忍私钥更新过程中的泄漏.

抗篡改安全. 2004 年, Gennaro 等 [83] 提出了抗篡改密码学, 要求即使敌手可访问篡改函数获取秘密状态的额外

信息, 密码方案仍然安全. 抗篡改的安全模型可以细分为计算篡改 (tampering with computation) 模型和唯内存篡改 (memory-only tampering) 模型.

计算篡改模型中, 拥有篡改能力的敌手可以同时篡改算法对应的电路 $\mathcal{C}$ 和内部秘密状态 st(例如包含私钥 sk), 即对 $(\mathcal{C}, st)$ 实施任何的篡改函数得到 $(\mathcal{C}', st') = f(\mathcal{C}, st)$. 若原始模型中的敌手可访问某个密码算法谕言机 $\mathcal{C}(st, \cdot)$, 那么计算篡改模型中的敌手就可访问篡改后的密码算法谕言机 $\mathcal{C}'(st', \cdot)$. 显然, 如果不给 f 施加限制, 那么可以使篡改后的密码算法对应的电路 $\mathcal{C}'$ 直接输出内部状态 st, 从而彻底攻破密码方案. 为了避免定义平凡, 抗篡改安全模型通常均会限制 f 来自某个篡改函数集 Φ. 因此, 篡改安全模型的强弱取决于篡改函数集 Φ 的大小, 如何扩大可容忍的篡改函数集是抗篡改密码学的核心问题. Ishai 等 [84] 在计算篡改模型下研究了这个问题, 他们采取的方法是构造通用的电路编译器, 将不抗篡改攻击的密码电路编译为抗篡改的. 后续研究 [85, 86, 87] 致力于扩展 [84] 中篡改函数集的限制.

唯内存篡改的模型中, 敌手只能对内部秘密状态 st 进行篡改, 而不能对电路本身进行篡改. 内部秘密状态主要可分为密钥 sk 和随机数, 但是 [88] 指出即使篡改少量的随机数, 对整个密码系统的影响都是毁灭性的, 故目前关于抗唯内存篡改的研究通常假设篡改攻击的敌手仅能篡改私钥 sk, 即实施相关密钥攻击 (related-key attack, RKA). 具体到公钥加密, 敌手可以询问解密谕言机 (ϕ, c) 以获得 c 在密钥 $\phi(sk)$ 下的解密值. 公钥加密方案若可抵抗篡改函数集为 Φ 的相关密钥攻击, 则称其是 Φ-RKA 安全的. RKA 安全的 PKE 方案研究可以分为两类, 一类关注可行的构造, 另一类关注不可能结果.

- 具体方案: Bellare 等在 [89] 首次将 RKA 安全的理论模型具体地推广到 PKE 中. Wee [90] 基于适应性陷门关系给出了 RKA-CCA 的通用构造, 篡改函数集为仿射函数集 Φ_{aff}. Qin 等 [91] 将 [92] 中提出的不可延展密钥导出函数 (non-malleable key derivation functions, NM-KDF) 扩展为连续不可延展密钥导出函数 (cNM-KDF), 并以此为工具将一系列密码方案 (包括公钥加密) 提升为对函数集 $\Phi^{\mathsf{hoe}}_{\mathsf{iocr}}$ (高输出熵且输入-输出抗碰撞的函数集, 该函数集包含次数有上界的多项式函数) 的 $\Phi^{\mathsf{hoe}}_{\mathsf{iocr}}$-RKA 安全的方案. Chen 等 [93] 提出不可延展函数 (non-malleable function, NMF), 涵盖并优化了 Qin 等 [91] 的工作. Dziembowski 等 [94] 提出了不可延展编码 (non-malleable code) 这一密码原语, 并基于该原语构造了一个能将不具有 RKA 安全的密码算法编译为具有 RKA 安全的密码算法的通用编译器. 后续的工作主要围绕不可延展编码的构造展开, 而其核心问题仍然是如何扩大篡改函数集的大小 [95, 96].
- 不可能结果: Wang 等 [97] 证明了在有界篡改模型下, 即使敌手的篡改次数被限制为一次, 也无法通过黑盒归约基于任意的计算困难问题证明多种具有唯一性的密码原语的 RKA 安全性. 具体包括: 可验证随机函数、单射单向函数、具有唯一性的签名和具有唯一消息性的加密, 涵盖了 Cramer-Shoup 加密 [27]、RSA 签名 [4]、Waters 签名 [98] 等方案.

紧归约安全. 与抗泄漏、抗篡改安全对传统安全模型的增强不同, 紧归约安全关注某一特定安全模型下的归约效率. 令 $\mathcal{R}$ 是安全性证明中的归约算法, 将敌手 $\mathcal{A}$ 转换为算法 $\mathcal{R}^{\mathcal{A}}$ 解决底层某个困难问题, 记敌手 $\mathcal{A}$ 和归约算法 $\mathcal{R}$ 的运行时间分别为 $t_{\mathcal{A}}(\kappa)$ 和 $t_{\mathcal{R}}(\kappa)$, 优势分别为 $\epsilon_{\mathcal{A}}(\kappa)$ 和 $\epsilon_{\mathcal{R}}(\kappa)$, 定义 $\ell(\kappa) = (t_{\mathcal{R}}(\kappa)/\epsilon_{\mathcal{R}}(\kappa))/(t_{\mathcal{A}}(\kappa)/\epsilon_{\mathcal{A}}(\kappa))$, 衡量归约损失的度量, $\ell(\kappa)$ 越大, 则归约损失越大, 归约越低效. 在密码背景下, 归约算法和敌手的运行时间通常均为多项式级别, 因此密码学归约更加关注比值 $L(\kappa) = \epsilon_{\mathcal{A}}(\kappa)/\epsilon_{\mathcal{R}}(\kappa)$. 若 $L(\kappa) = O(1)$(或者 $L(\kappa) = O(\kappa)$), 则称对应的密码方案为 (几乎) 紧归约安全的. 通常来说, 如果采用一般的混合论证技术 (hybrid argument), $L(\kappa)$ 往往与敌手控制的参数相关, 例如用户的数量 n、加密查询的次数 Q_e、解密查询的次数 Q_d 等, 约为 2^{30} 数量级 [99, 100], 对归约效率影响较大. 密码方案若安全归约低效, 则需增大安全参数以保证方案的安全性, 进而会影响方案效率, 而具有紧归约安全的密码方案无须调整安全参数. 综上, 对密码方案紧归约安全的研究不仅具有理论价值, 也具有实际意义.

公钥密码的紧归约安全的研究发源于 [101] 中对多挑战多用户模型下归约效率的讨论, 但是其并没有得到 $O(\kappa)$ 或者是 $O(1)$ 的紧归约 CCA 安全的 PKE 方案. 而后, 关于如何在标准模型下构造多挑战多用户的紧归约 CCA 安全的 PKE 方案这一论题, 密码学界展开了广泛且深入的研究. 目前设计紧归约 CCA 安全的 PKE 方案主要有两种方式, 第一种基于 Naor-Yung 双重加密范式, 第二种基于哈希证明系统:

- 基于 Naor-Yung 双重加密范式: 由于 Naor-Yung 双重加密范式的转换是紧的, 即在紧归约安全的 NIZK 下,

紧归约 CPA 安全的 PKE 方案可转换为紧归约 CCA 安全的 PKE 方案, 同时紧归约 CPA 安全的 PKE 方案容易构造 [101], 所以该路线的研究主要集中在如何构造紧归约安全的 NIZK 上. Hofheinz 和 Jager [102] 设计了第一个紧归约安全的结构保持的签名方案 (structure-preserving signature, SPS), 并利用 Groth-Sahai 证明 [103] 得到了第一个紧归约安全的无界模拟稳固 (unbounded simulation soundness, USS) 的 NIZK, 从而得到第一个基于标准假设的紧归约 CCA 安全的 PKE 方案. 但是 [102] 的开销过大, 后续的工作主要集中在如何保证紧归约的同时使 NIZK 更为高效 [104, 105, 106, 107]. 目前在标准模型、标准假设下最好结果是 Abe 等 [108] 构造的紧归约 USS 的 QANIZK 与 SPS.

- 基于哈希证明系统: 传统的 HPS [109, 27] 的固有结构难以实现紧归约 CCA 安全, 故相关工作主要集中在如何设计 HPS 的变体, 并基于此来设计紧归约 CCA 安全的 PKE 方案. Gay 等 [110] 基于标签对挑战密文、解密谕言机询问进行划分, 得到了第一个不基于配对的紧归约 CCA 安全的 PKE 方案; 但是 [110] 中由于私钥和公钥的数量与安全参数成线性关系而开销过大, 文献 [111] 则进一步改造 HPS 为合法证明系统 (qualified proof system, QPS), 借助于或证明 (OR proof) 的技术将私钥、公钥的数量缩减到常数, 得到了非常高效的紧归约 CCA 安全的 PKE 方案. 后续的工作主要围绕这二者进一步展开, 并将类似技术推广到更强的安全模型下 [112, 113, 100].

除此之外, 还有许多基于未充分研究的假设的紧归约工作, 其主要可以分为基于非标准的 q 类 (q-type) 假设 [41], 以及基于合数阶群上的计算困难假设 [114, 115], 这里不一一阐述.

1.4 功能性丰富

更广阔的应用场景要求 PKE 方案具有更为多样化的功能, 而不仅仅局限于加解密操作. 下面将介绍身份加密、属性加密、函数加密、可搜索加密、(全) 同态加密的研究概况.

身份加密. 传统公钥加密的密钥产生算法得到的公钥不具备语义特性, 因此公钥与用户的身份信息必须通过可信第三方签发的证书进行绑定, 在实际应用中, 公钥必须通过 PKI 体系的认证才能使用. 为了摆脱对 PKI 体系的依赖, Shamir [116] 提出了身份加密方案 (identity-based encryption, IBE). 相比传统公钥加密, 身份加密许用户使用任意字符串 id(如手机号码、Email、地址) 作为公钥, 同时增加了密钥派生算法, 私钥生成中心 (private key generator, PKG) 可通过主私钥 msk 派生出针对身份 id 的身份密钥 sk_{id}. IBE 规避了复杂的证书管理与公钥分发验证过程, 不再依赖 PKI. 不过, IBE 亦有一些缺点, 例如对 PKG 的固有依赖使得密钥托管 (key escrow) 问题难以避免.

Shamir 最初的工作只给出了 IBE 的概念, 并没有给出具体方案. 直到 2001 年前后, 三组密码科学家几乎同时独立的给出了 IBE 的具体构造 [117, 118, 119]. Gentry 和 Silverberg [120] 与 Horwitz 和 Lynn [121] 提出了层级身份加密 (HIBE, Hierarchical IBE). 以上方案均在随机谕言机模型下可证明安全. 为了在标准模型下构造安全的 IBE 方案, Canetti 等 [122] 提出了较弱的选择身份模型 (selective-identity model), 即敌手必须在游戏的开始阶段承诺挑战身份 id^*, 而非自适应选择. Boneh 和 Boyen [123] 在选择身份模型下基于 DBDH 假设给出了不依赖随机谕言机的 IBE 方案. 随后, Boneh 和 Boyen [124] 展示了如何使用可容许哈希函数 (admissible hash function) 实例化随机谕言机, 得到标准模型下的 IBE 方案, 然而该方案效率低下无法应用于实际. Waters [98] 通过巧妙的设计身份到群元素的映射实例化随机谕言机, 基于 DBDH 假设得到了首个标准模型下高效的 IBE 方案. Hohenberger 等 [125] 给出了基于不可区分程序混淆和可容许哈希函数实例化全域哈希类范式中随机谕言机的方法, 应用该方法可将 Boneh-Frankin IBE [118] 的安全性从随机谕言机模型提升到标准模型. 上述的所有 IBE 方案在证明中均采用的是分割策略 (partition strategy), 即归约算法将身份空间 I 切分为 I_0 和 I_1, 期望敌手选择 I_0 中的身份进行私钥询问, 选择 I_1 中的身份作为挑战. 分割策略成功的概率决定了归约的松紧. Gentry [126] 基于判定性可增双线性 Diffie-Hellman 指数 (DABDHE) 假设给出了标准模型下紧归约的高效 IBE 方案. 此外, Gentry 在该工作中还首次为 IBE 引入了匿名性. Gentry 的工作是 IBE 发展的一个里程碑, 后续的研究方向转向如何基于格基假设构造 IBE.

基于格的 IBE 与基于双线性映射的 IBE 演进路线基本一致. Gentry 等 [127] 首先构造出对偶的 Regev PKE 方案, 再利用随机谕言机将其升级为 IBE 方案. Agrawal 等 [128] 设计了格基采样算法, 以此在选择身份模型下设计出首个不依赖随机谕言机的 IBE 方案. Cash 等 [129] 设计了格基代理算法, 设计出首个完全安全的标准模型下的

IBE 方案. 虽然 [129] 达到了适应性安全, 但是由于其格基代理算法的固有结构使得其公钥、私钥、密文长度都是 $O(\kappa)$ 线性的, 其效率远不及 [128] 中选择身份模型下安全的公私钥、密文长度仅为 $O(1)$ 常数的 IBE 方案, 故后续的工作主要集中在如何兼顾完全的适应性安全以及较高的效率 [130, 131, 132, 133, 134]. 目前为止, 最优的结果由 [135] 给出, 其进一步改进了 Yamada [136] 中的划分技术, 在保持安全性和私钥、密文长度几乎不变的情况下, 将公钥降低到了 $\omega(1)$ 量级, 具体的方案下可为 $O(\log\log\kappa)$.

在 IBE 的各种具体构造百花齐放后, 密码学者开始探索构造的不可能结果. Boneh 等 [137] 指出 IBE 无法以黑盒的方式由陷门置换或 CCA 安全的 PKE 构造得出. Papakonstantinou 等 [138] 指出 IBE 无法以黑盒的方式基于 DDH 假设构造得出. Döttling 和 Garg [139] 使用基于混淆电路 (garbled circuits) 的非黑盒技术绕过上述不可能结果, 首次基于计算性 Diffie-Hellman 假设和大整数分解假设构造出 IBE.

在通用构造方面, Döttling 和 Garg [140] 展示了如何基于任意选择安全的 IBE 构造完全安全的 IBE. Hofheinz 和 Kiltz [141] 抽象出离散对数群上的可编程哈希函数 (programmable hash function, PHF), 以此解释了一类标准模型下基于分割证明策略的 IBE 方案. Zhang 等 [134] 提出并设计出格基可编程函数, 以此为工具设计出高效的 IBE 方案. Alwen 等 [75] 和 Chen 等 [142] 提出了身份哈希证明系统, 阐释了一系列基于判定性假设的 IBE 方案的设计思想 [126, 143, 127, 144]. Chen 等 [145] 提出了身份可提取哈希证明系统, 阐释了一系列基于搜索性假设的 IBE 方案的设计思想 [146, 147, 148, 149, 150].

最近关于 IBE 的工作主要是集中在如何增强 IBE 的功能性和安全性, 构造 IBE 的变体以应对更为丰富的实际场景、更加契合实际使用要求. 在功能性上, 例如密钥撤销 [151, 152, 153, 154, 155], IBE 的计算效率、身份匹配模式等性能性、灵活性等 [156, 157, 158].

属性加密. 如果把身份加密看作一种访问控制的策略, 那么 IBE 只有当解密者拥有加密者所期望的身份才能进行正确解密, 这属于一对一的简单访问控制结构, 但是对于现实应用而言, 通常希望一个加密的文件对于拥有某种权限的人都能解密, 即实现一对多的访问控制结构, 从而 IBE 这种简单的访问控制结构是远远不够的, 需要新的密码原语以提供更强大的访问控制结构. Sahai 和 Waters [159] 拓展了 IBE 的内涵, 结合秘密共享方案, 提出第一个基于门限访问控制的属性加密 (attribute-based encryption, ABE) 方案, 具体来说, 他们将基于身份加密下的身份看作由不同的属性构成, 这种 ABE 只要解密者的属性密钥与加密者所期望的属性集合足够相似即可以进行正确解密. 相比于基于身份加密下实现的一对一的访问控制结构, ABE 的主要优点是支持更为丰富、强大的控制策略, 具有一对多的访问控制结构, 更加灵活.

Goyal 等在 [160] 中正式给出了 ABE 的语义, 他们考虑到密文和密钥在控制与被控制关系上的对称性, 将 ABE 进一步划分为密钥策略的 ABE (key-policy ABE, KP-ABE) 以及密文策略的 ABE (ciphertext-policy ABE, CP-ABE) 以适用于不同的应用场景, 顾名思义, KP-ABE 就是将控制策略与委派的属性密钥相绑定, 而将属性集与密文相绑定, 能够正确解密当且仅当密文的属性集符合密钥的控制策略, CP-ABE 亦作类似解释, 除此之外, 他们利用双线性映射、秘密共享, 给出了第一个选择属性模型下的单调访问控制结构 (即与门和或门构成的布尔电路) 的 KP-ABE 方案, 但需要注意的是, 不同于秘密共享, 他们定义的 ABE 还需要额外地满足抗合谋性. 但是也正是为了满足抗合谋性, CP-ABE 的构造显得较为困难, Bethencourt 等 [161] 在一般群模型 (generic group model, GGM) 下通过给不同的属性密钥引入随机数以防止其重组与合谋实现了第一个支持单调访问控制结构的 CP-ABE. 利用类似的思想, Cheung 和 Newport [162] 在标准模型下构造了只支持与门构成的访问控制结构的 CP-ABE. Waters [163] 构造了在标准模型和标准假设下的支持单调访问控制结构的 CP-ABE, 密文大小仅与访问控制结构对应的单调电路大小成线性相关. 另一方面, 由于单调访问控制结构对应的布尔电路仅包含与门和或门, 并不完备, Ostrovsky 等 [164] 将 KP-ABE 推广到了非单调访问控制结构. Okamoto 和 Takashima [165] 则进一步将 CP-ABE 也推广到了非单调访问控制结构.

在安全性方面, 上面论述的早期 ABE 方案一般都只能达到选择模型下的安全性, 如果简单地使用 IBE 中的分割策略, 当属性空间较大时, 安全归约将指数级的松弛, 从而失效. Lewko 等 [203] 以合数阶群的双线性映射为代数工具, 结合双系统加密技术 [166] 构造出首个适应性安全的 ABE. Okamoto 等 [165] 引入对偶配对向量空间, 进而利用素数阶群的双线性映射模拟合数阶群的双线性映射, 同样结合双系统加密技术构造除了适应性安全的 ABE.

格上的构造与群上构造的发展路线几乎一致. Agrawal 等 [157] 利用类似于 [159] 中的技术, 构造出了格上第一个基于 LWE 问题的门限访问控制结构的属性加密. 同年, Zhang 等 [167] 则将类似的结果推广到了 CP-ABE 下. Boyen [168] 基于 LWE 假设构造了支持单调访问控制结构的 KP-ABE, 大大扩大了格上构造的 ABE 的控制策略的表达能力. Gorbunov 等 [169] 则更进一步, 基于 LWE 假设, 将格上 KP-ABE 的构造推广到了全电路 (即任意多项式深度), 其密钥大小和电路大小有关, 密文大小则与电路的深度成线性相关. Boneh 等 [170] 结合同态加密 [171] 的思想, 构造了支持访问控制结构为全电路的 KP-ABE, 相较于 [169], 其最大的效率提升在于密钥的大小仅与电路的深度有关. Datta 等 [172] 基于 LWE 假设给出了一个支持访问控制为 NC^1 电路的 CP-ABE 方案.

但是, ABE 的诸多构造主要强调的是加密消息的语义安全, 即具有载荷隐藏 (payload-hiding) 的性质, 而属性集是公开的, 但在一些场合下, 属性集也希望对外保密, 即希望 ABE 方案具有属性隐藏 (attribute-hiding) 的性质. 为了达到这样的安全要求, Katz 等 [173] 首先正式提出了谓词加密 (predicate encryption) 的概念, 其安全模型在不可区分实验下精准刻画了属性隐藏的含义, 他们基于合数阶群的双线性配对构造了选择性安全的内积谓词加密, 并基于内积谓词加密, 给出了针对多项式、析取逻辑等谓词族的谓词加密. 与 ABE 的研究路线相同, 谓词加密研究的主要问题在于如何兼顾安全性的同时扩大谓词族. Okamoto 等 [165] 利用对偶配对向量空间技术, 在素数阶群上基于 DLIN 假设亦构造出适应性弱属性隐藏安全的内积谓词加密. Okamoto 等 [174] 开发了对偶配对向量空间下的新技术, 他们基于 DLIN 假设构造了第一个适应性属性隐藏安全的内积谓词加密. Gorbunov 等 [175] 将 ABE 与同态加密相结合, 基于 LWE 假设构造了选择性安全但谓词族为有限深度布尔电路族的谓词加密. 一个自然的问题是能否构造谓词加密方案, 其在满足较强的适应性安全的同时亦可支持谓词族为更为广泛的电路族 (如 NC^1) 的谓词加密? Bitansky 和 Vaikuntanathan [176] 指出适应性安全且支持的谓词族为广泛的电路族的谓词加密将直接蕴含不可区分混淆 $i\mathcal{O}$. Wee [177] 为了避开上述结果, 提出了部分属性隐藏这一新的安全性质, 构造了半适应性 (semi-adaptive) 部分属性隐藏安全的谓词加密方案, 其在公开属性上支持的谓词族为算术分支程序 (arithmetic branching program, ABP), 在私有属性上支持的谓词族为内积函数, 首次得到了 “双赢” 的谓词加密构造. Datta 等 [178] 利用对偶配对向量空间的方法进一步将上述的结果推广到适应性安全下.

其他关于属性加密的研究主要是集中在如何进一步增强属性加密功能的多样性, 以便适用于更广泛的应用场景, 除此之外, 还有关于 ABE 各种变体的研究, 比如多权威 ABE [172, 179, 180]、注册化 ABE [181, 182] 等.

函数加密. 传统公钥加密的解密为完全或无 (all or nothing) 的方式, 即拥有私钥能够恢复全部信息, 否则得不到任何信息. 随着云计算时代的到来和隐私保护技术的发展, 要求公钥加密体制支持任意细粒度访问控制机制的需求愈发迫切. 考察如下的应用场景: Alice 想通过社交网站与好友们分享合照, 希望每个好友只能看到他本人和 Alice, 看不到其他人, 为了保护隐私信息, Alice 计划将合照加密后上传. 显然, 经典的公钥加密体制无法满足这一需求. 在此背景下, O’Neill [183] 和 Boneh 等 [184] 分别独立地提出函数加密 (functional encryption, FE) 的概念. 简单来说, 函数加密相较于传统的公钥加密增加了函数密钥委派算法, 其可以针对函数族中的合法 f(例如某个访问控制策略) 委派出函数密钥 sk_f, 函数密钥 sk_f 的拥有方可以调用函数加密方案的解密算法从密文中计算出 $f(m)$, 而不一定是数据 m 本身, 从而超越了传统密码的完全或无的加解密方式. 从而针对于上述分享合照实际场景, Alice 可以作为函数加密方案的主私钥持有方, 并且根据一定的委派策略为不同好友分享不同的函数密钥 sk_{f_i}, 这样不同的好友即仅能从加密的照片中计算出 $f_i(m)$, 即为他本人和 Alice 的合照. 函数加密极大地拓宽了公钥加密的内涵, 是传统公钥加密 (PKE)、身份加密 (IBE) [116, 118]、可搜索公钥加密 (public-key encryption with keyword searching, PEKS) [185]、模糊身份加密 [159]、属性加密 (ABE) [160, 186]、谓词加密 (predicate encryption, PE) [187, 173] 逐步升华泛化的结果. 就研究的函数族而言, 函数加密主要可以划分为两大类, 一类主要研究针对通用的函数族的函数加密方案, 其针对的函数族可为多项式深度布尔电路对应的函数族等, 这类研究虽然得到的密码方案功能性强大, 但往往会依赖于较沉重的密码学组件, 例如不可区分混淆 (indistinguishability obfuscation, $i\mathcal{O}$) 等, 通常其理论意义大于现实意义. 另外一类则主要研究针对特定且常用的函数族的函数加密方案, 例如内积函数、二次函数等, 其不像第一类研究中针对通用函数族的函数加密方案那样有强大的功能性, 但其构造的方案的效率、安全性却有大幅的提升, 具有更大的现实意义.

- 针对通用函数族的函数加密方案: 由于抗无界 (unbounded) 函数密钥腐化攻击的通用 FE 难以直接构造, 早期的研究主要是在有界函数密钥腐化模型下进行考虑, Sahai 和 Seyalioglu [188] 首次在单次函数密钥腐化

模型下, 利用混淆电路和 PKE 作为基本组件构造出了针对全电路的 FE. 但是其密钥大小与表示函数密钥的电路大小相关. 另外一方面, Garg 等 [189] 在无界函数密钥腐化模型下, 基于 $i\mathcal{O}$ 构造的选择性安全的通用 FE 方案. Boyle 等 [190] 则进一步基于推广的差分输入 $i\mathcal{O}$ 构造了适应性安全的通用 FE 方案. 而差分输入 $i\mathcal{O}$ 的构造可分别由 $i\mathcal{O}$ [191] 或多线性配对 [192] 给出.

- 针对特定函数族的函数加密方案: Abdalla 等 [193] 首先研究了针对较为简单的内积函数的函数加密方案, 他们利用类似于并行 ElGamal 加密的结构基于 DDH 假设构造了选择性安全的内积函数加密方案, 利用类似的结构, 他们也得到了基于 LWE 假设的选择性安全的内积函数加密方案. Agrawal 等 [194] 将 [193] 中的密文结构改为类似于 HPS 的密文结构, 并分别基于 DDH、DCR、LWE 假设给出了适应性安全的内积函数加密方案. Agrawal 等 [195] 则进一步改造 [194] 中的内积函数加密方案, 将其 IND-CPA 安全性提升为 SIM-CPA 安全性. 理论上, 拥有了内积函数加密, 很容易朴素地构造任意次数多项式的函数加密, 但是这种朴素构造密文尺寸过大, 比如二次函数的构造至少是平方的密文长度. 另一方面, 受限于目前的密码组件多是线性, 模拟二 (高) 次多项式时常会出现交叉项而难以处理, 从而二次函数加密的构造更显得困难. 目前的处理方式是用内积函数加密方案封装交叉项, 从而用线性组件模拟二次组件 [196, 197, 198, 199]. 但是, 到目前为止, 能否在密文为线性的长度下实现适应性安全的二次函数加密仍然是一个长期的公开问题.

其他关于函数加密的研究主要集中在如何进一步增强函数加密方案功能的多样性, 以便适用于更广泛的应用场景, 例如 Goldwasser 等 [200] 为了实现 n 个输入源下 n 个不同输入源索引的消息 $m_i (i \in [n])$ 的独立加密以及协同的函数解密, 提出了多输入函数加密这一新的密码概念, 而多输入函数加密在内积函数 [201, 202, 203]、二次函数 [204, 205] 上均有深入的研究成果. 除此之外, 函数加密还有诸如无界消息 [206, 207, 208]、多权威 [209]、去中心化 [210, 211]、多层次 [212] 等更丰富功能性下的扩展变体.

可搜索加密. 函数加密强大的功能性使得函数密钥的拥有方能够做指定的函数运算以知道消息的函数值, 其安全性则保证了函数密钥的拥有方除了消息的函数值外一无所知. 这样的特点使得函数加密具有广泛的应用场景, 例如, 考虑如下云存储的场景: 医疗机构将巨量的加密后的病历及其对应的关键词索引等数据存储在云服务器上, 除此之外, 医疗机构也希望能够通过与云服务器交互, 在加密的数据上检索任何一位病人的病历数据, 且不会泄漏关于病例数据的明文的任何有效知识给云服务器和第三方. 显然, 理论上来说, 针对全电路/图灵机的函数加密可以完美地实现上述的场景, 但是由于其存在性依赖于沉重的密码组件, 从而其极大的计算开销和查询复杂度使得其很难应用到现实生产环境中. 另一方面, 在加密数据上进行检索又是现实中非常普遍的、亟需解决的问题, Song 等 [213] 针对这样的场景, 首次提出了可搜索加密 (searchable encryption, SE) 的概念, 并利用对称加密构造了非常高效的可搜索加密方案. 从此之后, 可搜索加密飞速发展, 其上研究主要可以分为基于对称加密的对称可搜索加密 (symmetric searchable encryption, SSE), 以及基于公钥加密的非对称可搜索加密 (asymmetric searchable encryption, ASE), 下面将主要介绍 ASE 方面的研究成果.

Boneh 等 [185] 首次将可搜索加密引入了公钥密码学的领域, 为其建立了基本的语义和形式化的语义安全模型, 并分别基于双线性映射和陷门置换函数构造了第一个基于公钥加密的关键词可搜索方案 (public-key encryption with keyword searching, PEKS). Abdalla 等 [214] 则注意到身份加密中的身份和搜索关键词的关联性, 给出了匿名 IBE 方案到 PEKS 方案的通用转换. 但是以上关于 ASE 的工作都是针对单关键词的可搜索加密, 现实中的搜索应该支持更为丰富的高级搜索选项, 例如多个关键词、区间搜索、模糊搜索等 [215, 187, 216]. 在安全模型上, Baek、Zhang、Bellare 等 [216, 217, 218] 注意到一般在讨论可搜索加密的时候 [185, 214] 都是单独考虑关键词域的安全性, 他们认为这样安全模型并不完整, 因此将关键词域部分的 PEKS 和对应的数据域部分的 PKE 作为一个整体, 同时考虑其安全性, 定义了 PKE-PEKS 的安全模型, 并给出了满足增强安全性后的 PKE-PEKS 方案. Chen 等 [219] 进一步改进了 Bellare 等 [218] 的构造, 他们给出了从匿名 HIBE 到 PKE-PEKS 的通用构造, 其密钥大小约为 [218] 中的一半. 后续的工作则是进一步细化关键词的搜索匹配模式 [220, 221, 222, 223], 以及将可搜索加密推广到更广泛的使用场景下 (例如多用户等) [224, 225].

以上构造可搜索加密的底层公钥加密方案几乎都是概率性加密算法, 虽然得到的安全性较高, 但是云存储服务器的查询复杂度一般都与数据库中的数据量成线性关系, 针对存储着大量加密数据的云服务器, 这样的查询复杂度是无法接受的. 另外一条实现可搜索加密的路径是使用确定性加密 (deterministic encryption) 方案. Bellare

等 [226] 提出了确定性公钥加密, 形式化了其 PRIV 安全模型, 并基于符合 PRIV 安全性的确定性加密方案构造高效的可搜索方案, 其查询复杂度仅为数据量的对数级别. 最开始满足 PRIV 安全性的确定性加密方案是在随机谕言机 (RO) 下进行构造的, 而 Bellare 等 [227] 利用 [31] 中的技术在较弱的 PRIV1 安全模型下给出了标准模型和标准假设下的构造, 并在 [228] 中详细讨论了确定性加密不同安全性的等价关系, 给出了更多的构造. 事实上, 如果关键字域的最小熵较小, 那么基于确定性加密的可搜索加密会遭受暴力遍历攻击, 所以如何为确定性加密定义更强的安全模型并给出相应的构造是后续研究的主要关注点, 一系列的工作均围绕此展开 [229, 230, 231, 232, 233, 234].

(全) 同态加密. 身份加密、属性加密、函数加密、可搜索加密的解密过程可以统一地理解为用私钥对密文进行秘密计算, 计算的结果为明文. Rivest 等 [235] 提出的同态加密则允许对密文进行公开的计算, 得到的密文恰好与对明文施加同样的计算对应. 同态加密的方案可分为部分同态加密和全同态加密 (fully homorphic encryption, FHE), 其中全同态的加密方案支持任意深度的电路, 而部分同态的加密方案仅支持非常受限类型的电路. 事实上, 部分同态的 PKE 方案很早就存在 [236, 4, 237, 238]. 2009 年 Gentry [239] 基于理想格提出了真正意义上的全同态加密, 这是全同态加密领域的一个里程碑式的研究成果, 此后, 各种全同态加密方案被陆续提出, 按照核心技术大致可以分为四代.

- 第一代全同态加密: Gentry [239] 提出了第一个全同态加密方案, 其中的关键为自举 (bootstrapping) 技术, Gentry 基于理想格上的困难问题构造了满足自举性质的部分同态加密方案, 然后将其转换全同态加密方案. Gentry 的全同态加密方案是理论上的重大突破, 但是实际运行效率并不高, 每比特运算需要耗时 30 分钟 [240]. Dijk 等 [241] 将 [239] 底层的基于理想格的部分同态加密方案换为基于整数的部分同态加密方案, 并沿用 Gentry 的技术路线, 得到了第二个全同态加密方案.
- 第二代全同态加密: Brakerski 和 Vaikuntanathan [242, 243] 引入了重线性化技术和模约简技术, 在标准格上基于 (R)LWE 假设以及循环安全假设 (circular security assumption) 构造了两个新的全同态加密方案. Brakerski 等 [244] 则进一步提出支持有限深度电路的层次化的全同态加密的概念, 使得全同态加密在一定程度上摆脱了计算开销较大的自举技术, 大幅提高了效率. 后续的工作 [245, 246, 247, 248] 通过引入并行化、打包、批处理等优化技术, 进一步提升效率.
- 第三代全同态加密: Gentry 等 [171] 利用新的技术–近似特征向量方法 (approximate eigenvector method), 避免了重线性化给乘法带来的较大噪声. Brakerski 和 Vaikuntanathan [249] 则注意到对于特殊的电路, GSW 方案的噪声增长速度更慢, 从而给出了更高效和更安全的实现, 该全同态加密方案基于多项式近似的 GapSVP 假设. Sheriff 和 Peikert [250] 进一步基于此观察, 同时结合快速傅里叶 (fast fourier transform, FFT) 算法等优化方法, 得到了具有非常高效的自举过程的全同态加密方案. Chillotti 等 [251] 基于 [250] 的思想, 利用另外一种自举方式 [252] 得到了极其高效的实现, 其自举一个比特仅需要 12ms.
- 第四代全同态加密: Cheon 等 [253] 提出了可以支持某些浮点运算的分层次全同态加密方案, 而浮点运算是诸多实际应用 (例如训练神经网络) 所需要的, 这使得分层次全同态加密具备了大范围地应用到工业的可能. 次年, 他们 [254] 又将类似结果推广到了全同态加密下. 后续工作 [255, 256, 257, 258] 致力于 CKKS 方案的高效安全实现.

回顾发展历程, 自从 Gentry [239] 在 2009 年提出第一个全同态加密方案后, 全同态加密的研究主要是对自举技术不断改进, 将自举开销从分钟量级 [240] 最终降低到毫秒量级 [259], Rivest 等 [235] 的愿景终于成为现实.

第二章 准备知识

章前概述

万丈高楼平地起, 一砖一瓦皆根基.

— 中国成语

内容提要

- 符号与记号
- 可证明安全方法简述
- 困难问题
- 复杂性理论
- 信息论工具
- 密码组件

本章介绍必须的准备知识, 为本书展开后续内容做铺垫. 2.1节规定了本书所使用符号、记号与术语, 2.2节简要介绍了可证明方法, 2.3节介绍了常见的计算困难问题, 2.4节介绍了计算复杂性最为基本的一些概念, 2.5节介绍最基本的信息论概念, 2.6节介绍了本书中涉及的密码组件.

2.1 符号、记号与术语

集合. 对于正整数 n, 用 $[n]$ 表示集合 $\{1,\dots,n\}$. 对于集合 X, $|X|$ 表示其大小, $x \xleftarrow{R} X$ 表示从 X 中均匀采样 x, U_X 表示 X 上的均匀分布.

基本算术. 对于实数 $x \in \mathbb{R}$, 令 $\lfloor x \rfloor$ 表示 x 的下取整, $\lfloor x \rceil := \lfloor x + 1/2 \rfloor$ 表示与 x 最接近的整数 (x 的就近取整). 对于向量 $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|$ 表示 $\mathbf{x}$ 的 2-范数. 整数集合定义为 $\mathbb{Z} \stackrel{\text{def}}{=} \{\dots,-2,-1,0,1,2,\dots\}$. 自然数集合定义为 $\mathbb{N} \stackrel{\text{def}}{=} \{0,1,2,\dots\}$.

字符串. 令 $\{0,1\}^n$ 表示 n-bit 二进制字符串的集合, $\{0,1\}^*$ 表示所有 (长度有限) 的二进制字符串集合. 令 x 为二进制字符串, $|x|$ 表示其 bit 长度, $\bar{x}$ 表示 x 取反. 0^n 和 1^n 分别表示长度为 n 的全 0 串和全 1 串.

算法与函数. 若一个概率算法的运行时间是关于输入规模 n 的多项式函数 $\mathsf{poly}(n)$, 则称其是概率多项式时间的算法, 简记为 (probabilistic polynomial time, PPT). 令 $\mathcal{A}$ 是一个随机算法, $z \leftarrow \mathcal{A}(x;r)$ 表示 $\mathcal{A}$ 在输入为 x 和随机带为 r 时输出 z, 当上下文明确时, 常隐去随机带 r 简记为 $z \leftarrow \mathcal{A}(x)$. 令 $f(\cdot)$ 是关于 n 的函数, 如果对于任意的多项式 $p(\cdot)$ 均存在常数 c 使得 $n > c$ 时总有 $f(n) < 1/p(n)$ 成立, 则称 f 是关于 n 的可忽略函数, 记为 $\mathsf{negl}(n)$, 另外, 称 $1-\mathsf{negl}(n)$ 为压倒性函数; 如果存在多项式 $q(\cdot)$ 以及常数 d 使得 $n > d$ 总有 $f(n) > 1/q(n)$, 则称 f 是关于 n 的可察觉函数. 本书中使用 $\kappa \in \mathbb{N}$ 表示计算安全参数, $\lambda \in \mathbb{N}$ 表示统计安全参数. 令 F 是带密钥的函数, $F_k(x)$ 表示函数 F 在密钥 k 控制下对 x 的求值, 也常记作 $F(k,x)$.

统计距离. 令 X 和 Y 是定义在 Ω 上的两个分布, 两者之间的统计距离定义为 $\Delta(X,Y) = \frac{1}{2}\sum_{\omega\in\Omega}|\Pr[X=\omega] - \Pr[Y=\omega]|$. 令 $\mathcal{X} = \{X_\kappa\}_{\kappa\in\mathbb{N}}$ 和 $\mathcal{Y} = \{Y_\kappa\}_{\kappa\in\mathbb{N}}$ 是两个由 κ 索引的分布簇, 则可考察两者之间渐进意义下的统计距离. 如果 $\mathcal{X}$ 和 $\mathcal{Y}$ 之间的统计距离为 0, 则称 $\mathcal{X}$ 和 $\mathcal{Y}$ 完美不可区分. 如果 $\mathcal{X}$ 和 $\mathcal{Y}$ 之间的统计距离是关于 κ 的可忽略函数, 则称 $\mathcal{X}$ 和 $\mathcal{Y}$ 统计不可区分, 记为 $\mathcal{X} \approx_s \mathcal{Y}$; 如果任意 PPT 敌手区分 $\mathcal{X}$ 和 $\mathcal{Y}$ 的优势函数为 $\mathsf{negl}(\kappa)$, 则称 X 和 Y 计算不可区分, 记为 $\mathcal{X} \approx_c \mathcal{Y}$.

方案和协议. 已有文献中并没有对密码方案 (scheme) 和密码协议 (protocol) 的清晰定义, 常常交换使用两个名词. 本书使用密码方案特指若干实体通过运行系列算法完成某项密码操作的全流程, 实体之间不存在交互, 仅存在单向的消息传递, 如加密和签名. 与密码方案相比, 密码协议则允许实体之间存在交互, 即存在双向的消息传递, 如密钥协商、安全多方计算和零知识证明等. 当协议中不存在消息传递或仅存在单向的消息传递时, 协议退化为非交互式版本.

表 2.1: 缩略词及其含义对照表

缩略词	英文表达	中文含义
CPA	chosen-plaintext attack	选择明文攻击
CCA	chosen-ciphertext attack	选择密文攻击
RKA	related-key attack	相关密钥攻击
KDM	key-dependent message attack	消息依赖密钥攻击
PKE	public-key encryption	公钥加密方案
IBE	identity-based encryption	身份加密方案
PEKS	public-key encryption with keyword search	可搜索公钥加密方案
–	hardcore function	硬核函数
–	hardcore predicate	硬核谓词
–	oracle	谕言机
TDF	one-way trapdoor function	单向陷门函数

2.2 可证明安全方法

Proving it is what makes it sciences.

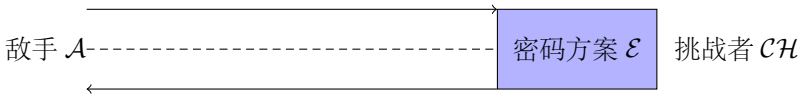
— Indiana Jones and the Dial of Destiny

长久以来, 密码方案的安全性分析缺乏统一规范, 通常是由密码分析者遍历各类攻击来检验密码方案的安全性. 容易看出, 传统的分析方式存在以下局限: (1) 分析结果严重依赖分析者的个人能力 (细致的观察、敏锐的直觉和积累的经验); (2) 分析者难以穷尽所有可能的攻击. 因此, 绝大多数古典密码陷入 "设计—攻破—修补—攻破" 的循环往复怪圈, 难以称之为真正的科学.

上世纪八十年代, Goldwasser 和 Micali [5] 借鉴计算复杂性理论的归约技术, 开创了可证明安全方法. 从此, 密码方案的安全性分析手段由遍历攻击转为严格的数学证明, 安全性由 "声称安全" 变为 "可证安全", 密码学也从此由艺术蝶变为真正的科学.

简言之, 可证明安全方法的核心是以下三要素组成:

- **精确的安全模型:** 通常由攻击者和挑战者之间的交互式游戏进行刻画, 如图 2.1所示, 包括:
  - 敌手的计算能力: 常见的有概率多项式时间和指数时间
  - 敌手能够获取的信息, 包括:
    1. 固定信息: 如方案的公开参数等公开信息
    2. 非固定信息: 在攻击过程中获得的信息, 形式化为访问相应谕言机获得的输出
  - 敌手的攻击效果: 以加密方案为例, 敌手恢复密钥和恢复明文是不同的攻击效果



$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S] - t|$$

图 2.1: 安全模型

令事件 S 表示敌手攻击成功这一事件, t 表示目标基准优势 (如区分类游戏定义为 $1/2$, 搜索性游戏定义为 0), 定义 $\mathcal{A}$ 的优势函数为 $\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S] - t|$, 其中 S 所在的概率空间由 $\mathcal{A}$ 和 $\mathcal{CH}$ 的随机带确定. 后续行文中为了表述简洁, 常省略优势函数的绝对值符号. 如果对于所有 PPT 敌手 $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 均是关于 κ 的可忽略函数, 则称密码方案 $\mathcal{E}$ 在既定的安全模型下是安全的.

- **清晰的困难性假设**
- **严格的归约式证明:** 通过反证式论证将方案的安全性归结到困难性假设.

图 2.2是归约式证明的交换图表. 归约式证明的步骤如下:

1. 假设存在 PPT 的敌手 $\mathcal{A}$ 在既定安全模型下针对密码方案 $\mathcal{E}$ 具有不可忽略的优势 $\epsilon_1(\kappa)$;
2. 利用 $\mathcal{A}$ 的能力, 构建 PPT 的算法 $\mathcal{R}$ 以不可忽略的优势 $\epsilon_2(\kappa)$ 打破困难问题. 这里 $\mathcal{R}$ 通常以扮演敌手 $\mathcal{A}$ 的挑战者的方式调用 $\mathcal{A}$, 因此也常称 $\mathcal{R}$ 是模拟算法或归约算法. $\mathcal{R}$ 调用敌手 $\mathcal{A}$ 的方式又可细分为两类:
   - 黑盒方式: $\mathcal{R}$ 以黑盒的方式调用 $\mathcal{A}$, 从算法的角度理解就是 $\mathcal{R}$ 将 $\mathcal{A}$ 作为子程序调用. 黑盒方式实质上证明了比所需更强的结果, 即 $\exists\mathcal{R}\ \forall\mathcal{A}$. 此类证明最为常见, 被称为黑盒归约 (black-box reduction) 或者一致归约 (universal reduction).
   - 非黑盒方式: $\mathcal{R}$ 以非黑盒的方式调用 $\mathcal{A}$, 充分利用了 $\mathcal{A}$ 的个体信息, 如算法结构、运行时间等. 这类证明是完全契合可证明安全思想的, 即 $\forall\mathcal{A}\ \exists\mathcal{R}$. 此类证明相对少见, 被称为个体归约 (individual reduction) [260], 常可以突破黑盒归约下的安全性下界.

上述两步归约式论证的逻辑是: 构造出的算法 $\mathcal{R}$ 与困难假设相矛盾, 因此不存在算法 $\mathcal{R}$, 进而得出 $\mathcal{A}$ 不存在的结论.

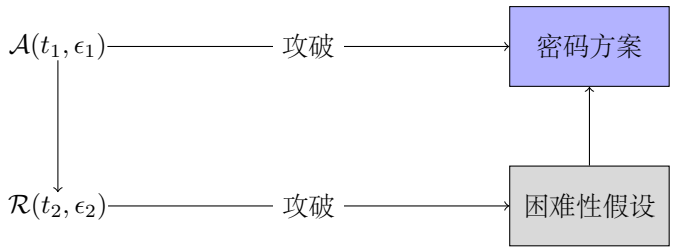


图 2.2: 归约式证明的交换图表

安全性强弱. 在明确可证明安全方法后, 密码方案的强弱可以根据三要素的强弱定性分析:

- 安全模型的强弱: 敌手的计算能力越强大、获得的信息越多、攻击的效果越弱, 则所确定的安全模型越强, 反之越弱.
- 困难假设的强弱: 通常搜索类假设强于判定类假设, 平均情形 (average-case) 假设强于最坏情形 (worst-case) 假设.
- 归约质量的优劣: 笼统地说, (t_2, ϵ_2) 越接近 (t_1, ϵ_1), 归约的质量越高. 归约算法和敌手运行时间均为多项式级别, 因此在考察归约算法质量时更关注优势函数这一指标. 定义归约松紧因子 $r = \epsilon_2/\epsilon_1$, 如果 r 是一个常数, 称归约是紧的; 如果 r 是一个可察觉的函数 (noticeable function), 称归约是多项式松弛的, 归约有效; 如果 r 是一个可忽略函数, 称归约是超多项式松弛的, 归约无效.

注记 2.1

阿基米德曾说过:"给我一个支点, 我能撬起地球!" 可证明安全方法与这句名言有共通之处, 地球可以理解为待证明方案的安全性, 支点和杠杆可以理解为归约式证明方法, 而施加在杠杆上的力可以理解为困难性假设. 如果支点在困难性假设和方案安全性正中, 代表归约最优, 方案的安全性可以紧归约到假设困难性上; 如果力臂过短, 则代表归约松弛, 困难性假设无法有意义的保证密码方案的安全性.

2.2.1 如何书写安全性证明

很多初学者对方案/协议的安全性有隐约的直觉, 但是很难写出严格精确的证明. 密码学中的安全性证明如同吉他中的大横按, 是横亘在所有初学者面前的一个障碍.

本小节将以极为精炼的方式归纳总结安全性证明的构建方式. 给出安全性证明大致有两种方式, 分别是单一归约和游戏序列.

单一归约适用于密码方案/协议仅依赖单一困难问题的简单情形. 拟基于惟一困难假设 $\mathcal{P}$ 证明密码方案/协议

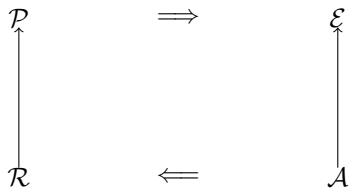


图 2.3: 单一归约

$\mathcal{E}$ 的安全性时, 证明的方式是构建如图 2.3所示的交换图表, 即首先假设存在 PPT 的敌手 $\mathcal{A}$ 打破 $\mathcal{E}$ 的安全性, 再利用 $\mathcal{A}$ 构造算法 $\mathcal{R}$ 打破困难假设 $\mathcal{P}$. 构造 $\mathcal{R}$ 的方法通常是令 $\mathcal{R}$ 在方案 $\mathcal{E}$ 的安全游戏中模拟挑战者的角色, 模拟的方式是将困难问题的实例直接嵌入到安全游戏的参数中. 此时, 基于 $\mathcal{P}$ 的困难性——$\mathsf{Adv}^{\mathcal{P}}_{\mathcal{R}}(\kappa)$——便可得到任意 PPT 敌手针对 $\mathcal{E}$ 的优势函数 $\mathsf{Adv}^{\mathcal{E}}_{\mathcal{A}}(\kappa) \leq \mathsf{negl}(\kappa)$ 的结论.

注记 2.2

需要特别指出, 单一归约的适用范围有限, 仅适用于困难问题单一且能够直接嵌入密码方案安全游戏的场景, 这就要求安全游戏的目标和困难问题的类型必须相同 (同为计算性或者判定性), 如基于 DLOG 假设的单向函数和基于 DDH 假设的伪随机数发生器.

对于基于多个困难问题的密码方案/协议, 即待证明的定理形如 $\mathcal{P}_1 + \cdots + \mathcal{P}_n \Rightarrow \mathcal{E}$, 就难以使用单一归约进行证明了. Shoup [261] 针对该情形, 系统地提出了 "游戏序列" 的方式组织证明. 游戏序列的证明框架如下:

1. 引入一系列游戏, 记为 $\mathsf{Game}_0, \ldots, \mathsf{Game}_m$. 敌手在 Game_i 中成功的事件记作 S_i, 优势基准为 t, 则敌手在 Game_i 的优势函数为:
$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_i} = |\Pr[S_i] - t|$$
通常情况下 Game_0 刻画原始真实的安全游戏, Game_m 刻画最终游戏且 $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_m} = |\Pr[S_m] - t| = \mathsf{negl}(\kappa)$, 即敌手在 Game_m 中的优势函数是可忽略的.
2. 证明对于所有的 $i \in [m]$ 均有 $|\Pr[S_i] - \Pr[S_{i-1}]| \leq \mathsf{negl}(\kappa)$;
3. 通过混合论证 (hybrid argument) 得出 $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Game}_m} = |\Pr[S_0] - t| = \mathsf{negl}(\kappa)$ 的结论.

对于同一密码方案/协议, 在使用游戏序列进行证明时存在多种可能的游戏序列组织方式. 尽管游戏序列的设定没有严格的规定, 但有以下两个经验准则:

- 相邻游戏的差异需最小化, 下一个游戏与上一个游戏仅有一个差异为宜;
- 差异应易于分析.

相邻游戏之间的差异通常有以下三种类型:

1. 差异源于不可区分的分布;
2. 差异基于某特定事件是否发生;
3. 差异仅是概念上调整, 为后续分析做铺垫.

对于第一类差异, Game_i 和 Game_{i+1} 的变化可以归结为分布的不可区分性, 如 $Z_0 \approx Z_1$, 其中 Z_0 和 Z_1 是两个分布. 换言之, 存在归约算法 B, 以 Z_0 为输入时, 可以完美模拟敌手在 Game_i 中的视图; 以 Z_1 为输入时, 可以完美模拟敌手在 Game_{i+1} 中的视图. 我们令 View_i 表述敌手在 Game_i 中的视图. 在上下文清晰没有歧义时, 也常用 Game_i 直接代指敌手的视图.

- 当 $Z_0 \approx_s Z_1$ 时, 利用复合引理 (composition lemma) 可以立刻得出任意敌手在两个游戏中的输出统计不可区分, 进而得出 $\Pr[S_0] - \Pr[S_1] \leq \mathsf{negl}(\kappa)$
- 当 $Z_0 \approx_c Z_1$ 时, 如果敌手成功这一事件 $\mathcal{R}$ 可准确判定, 则同样可以得出 $\Pr[S_0] - \Pr[S_1] \leq \mathsf{negl}(\kappa)$, 论证过程如下图 2.4所示, $\mathcal{R}$ 在事件 S 发生时输出 "1", 否则输出 "0". 根据游戏定义, 有 $\Pr[\mathcal{R}(Z_0)] = \Pr[S_i]$, $\Pr[\mathcal{R}(Z_1)] = \Pr[S_{i+1}]$, 因此有:
$$|\Pr[S_i] - \Pr[S_{i+1}]| = |\Pr[\mathcal{R}(Z_0) = 1] - \Pr[\mathcal{R}(Z_1) = 1]| \leq \mathsf{negl}(\kappa)$$

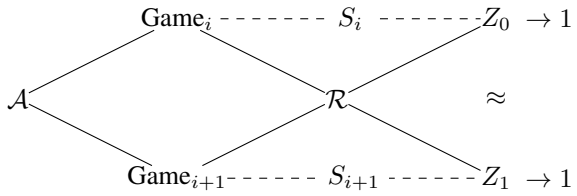


图 2.4: 基于分布不可区分的游戏演变

注记 2.3

许多学术论文在证明的过程中为了简便往往先证明敌手在两个相邻游戏中的视图不可区分，再据此得出敌手优势函数的差是可忽略这一结论. 在多数情形，这种论证并无问题，但需要特别小心的是在视图计算不可区分时，必须同时确保归约算法能够准确判定敌手成功事件才能够确保证明严谨. 在有些特殊情形，归约算法无法有效判定敌手是否在游戏中成功，此时归约算法无法利用敌手成功概率差异打破底层区分性假设，从而导致归约失效. 请读者参考文献 [59] 加深对该证明技术细节的理解和掌握.

第二类差异取决于某个特定事件是否发生，即定义在同一概率空间的两个相邻游戏 Game_i 和 Game_{i+1} 仅在某特定事件 F 发生时存在差异，在 F 不发生时完全一致，概率描述如下：

$$S_i \wedge \overline{F} = S_{i+1} \wedge \overline{F}$$

为了分析敌手在相邻游戏 Game_i 和 Game_{i+1} 中的优势函数差，需要以下的 “差异引理”(difference lemma)：

引理 2.1 (Difference Lemma)

令 A, B, F 是定义在同一概率空间中的事件，如果 $A \wedge \overline{F} = B \wedge \overline{F}$，那么则有 $|\Pr[A] - \Pr[B]| \leq \Pr[F]$.

证明 差异引理的证明如下，仅需使用古典概率中简单的缩放技巧：

$$\begin{aligned}
|\Pr[A] - \Pr[B]| &= |\Pr[A \wedge F] + \Pr[A \wedge \overline{F}] - \Pr[B \wedge F] - \Pr[B \wedge \overline{F}]| \text{//全概率展开} \\
&= |\Pr[A \wedge F] - \Pr[B \wedge F]| \text{//化简} \\
&\leq \max\{\Pr[A \wedge F], \Pr[B \wedge F]\} \leq \Pr[F] \text{//缩放}
\end{aligned}$$

证毕！ □

根据差异引理，若需证明 $|\Pr[S_i] - \Pr[S_{i+1}]| \leq \mathsf{negl}(\kappa)$，仅需证明 $\Pr[F] \leq \mathsf{negl}(\kappa)$，证明细分为以下两种情形：

- F 发生的概率取决于敌手的计算能力，如敌手找到哈希函数的碰撞或者成功伪造消息认证码. 该情形需要建立安全归约，即若 F 发生，则存在敌手打破困难问题 X_i.
- F 发生的概率与敌手的计算能力无关. 该情形仅需纯粹的信息论论证 (information-theoretic argument).

第三类差异称为桥接差异. 在分析游戏序列之间的差异时，有时需要引入桥接步骤对某个变量的生成方式以等价的方式重新定义，以确保差异分析的良定义. 桥接步骤引入的差异仅是挑战者侧的概念性变化，敌手侧的视图完全相同，因此 $\Pr[S_i] = \Pr[S_{i+1}]$. 桥接步骤看似可有可无，实则必要，若不引入必要的桥接步骤，则会使得证明跳跃难以理解、游戏序列间的差异无法精确分析.

2.3 困难问题

密码学中常见的困难问题可大致分为数论类和格类, 其中前者是代数问题, 后者是数的几何问题, 这也正是格类困难问题具备抗量子攻击的原因之一.

数论类的假设又可进一步分为整数分解类和离散对数类两个分支. 本章首先介绍常见的数论类假设, 再介绍常见的格类困难问题.

2.3.1 整数分解类假设

整数分解类假设定义在群 $\mathbb{Z}_N^*$ 上. 其中:

$$\mathbb{Z}_N^* \stackrel{\text{def}}{=} \{b \in \{1, \ldots, N_1\} \mid \gcd(b, N) = 1\}$$

也即 $\mathbb{Z}_N^*$ 是整数集合 $\{1, \ldots, N-1\}$ 中所有与 N 互质元素构成的子集, 在模乘运算 $ab \stackrel{\text{def}}{=} [ab \bmod N]$ 下构成交换群.

以下令 GenModulus 是 PPT 算法, 其以安全参数 1^κ 为输入, 输出 $(N = pq, p, q)$, 其中 p 和 q(以压倒性概率) 是两个 κ-bit 的素数.

定义 2.1 (整数分解假设)

整数分解问题指分解大整数在平均意义下是困难的. 我们称整数分解假设相对于 GenModulus 成立当且仅当对于任意 PPT 敌手:

$$\Pr[\mathcal{A}(N) = (p', q') \text{ s.t. } p'q' = N] \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$ 和 $\mathsf{GenModulus}(1^\kappa) \to (N, p, q)$ 的随机带上.

♣

尽管整数分解假设历经百年的分析攻击仍然健壮, 但是它并不直接蕴含高效的密码系统. 这就启发密码学工作者研究与整数分解问题困难性相关问题的研究. 1978 年, Rivest, Shamir 和 Adleman 提出了 RSA 问题.

令 GenRSA 是 PPT 算法, 其以安全参数 1^κ 为输入, 输出两个 κ-bit 素数的乘积 N 作为模数, 同时输出正整数 (e, d) 满足 $ed = 1 \bmod \phi(N)$.

定义 2.2 (RSA 假设)

RSA 问题指在平均意义下求解 $\mathbb{Z}_N^*$ 的 e 次方根是困难的. 我们称 RSA 假设相对于 GenModulus 成立当且仅当对于任意 PPT 敌手:

$$\Pr[\mathcal{A}(N, e, y) = x \text{ s.t. } x^e = y \bmod N] \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$、$\mathsf{GenRSA}(1^\kappa) \to (N, e, d)$ 和随机选取 $x \in \mathbb{Z}_N^*$ 的随机带上.

♣

注记 2.4

RSA 问题刻画的是在不知晓 $\phi(N)$ 的情况下计算 $\mathbb{Z}_N^*$ 中随机元素的 e 次方根是困难的. 容易看出, 如果敌手能够打破整数分解问题, 则可以通过分解 N 求出 $\phi(N)$, 进而通过 Fermat 小定理计算 e 次方根. 因此, 整数分解问题难于 RSA 问题, 整数分解假设弱于 RSA 假设. 两个假设是否等价仍然未知.

♠

给定群 $\mathbb{G}$, 称 $y \in \mathbb{G}$ 是一个二次剩余当且仅当 $\exists x \in \mathbb{G}$ s.t. $x^2 = y$. x 称为 y 的平方根. 如果一个元素不是二次剩余则称其为二次非剩余. 在 abelian 群中, 二次剩余构成子群.

首先考察群 $\mathbb{Z}_p^*$ 中的二次剩余, 其中 p 是素数. 定义函数 $\mathsf{sq}_p : \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ 为 $\mathsf{sq}_p(x) \stackrel{\text{def}}{=} [x^2 \bmod p]$. 当 p 是大于 2 的素数时, sq_p 是 2-对-1 函数, 因此立刻可知 $\mathbb{Z}_p^*$ 中恰好一半元素是二次剩余. 记模 p 的二次剩余集合为 $\mathcal{QR}_p$, 模 p 的二次非剩余集合为 $\mathcal{QNR}_p$, 我们有:

$$|\mathcal{QR}_p| = |\mathcal{QNR}_p| = \frac{|\mathbb{Z}_p^*|}{2} = \frac{p-1}{2}$$

定义元素 $x \in \mathbb{Z}_p^*$ 模 p 的 Jacobi 符号如下:

$$\mathcal{J}_p(x) \stackrel{\text{def}}{=} \begin{cases} +1 \text{ if } x \in \mathcal{QR}_p \\ -1 \text{ if } x \in \mathcal{QNR}_p \end{cases}$$

再考察群 $\mathbb{Z}_N^*$ 中的二次剩余, 其中 N 是两个互异素数 p 和 q 的乘积. 由中国剩余定理可知: $\mathbb{Z}_N^* \simeq \mathbb{Z}_p^* \times \mathbb{Z}_q^*$, 令 $y \leftrightarrow (y_p, y_q)$ 表示上述同构映射给出的分解, 易知 y 是模 N 的二次剩余当且仅当 y_p 和 y_q 分别是模 p 和模 q 的二次剩余. 定义函数 $\mathsf{sq}_N : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ 为 $\mathsf{sq}_N(x) \stackrel{\text{def}}{=} [x^2 \bmod N]$, 当 N 为互异素数乘积时, sq_N 是 4–对–1 函数. 记模 N 的二次剩余集合为 $\mathcal{QR}_N$, 由 $\mathcal{QR}_N$ 与 $\mathcal{QR}_p \times \mathcal{QR}_q$ 之间的一一对应关系可知:

$$\frac{|\mathcal{QR}_N|}{|\mathbb{Z}_N^*|} = \frac{|\mathcal{QR}_p| \cdot |\mathcal{QR}_q|}{|\mathbb{Z}_N^*|} = \frac{1}{4}$$

从二次剩余的角度可以对 $\mathbb{Z}_N^*$ 中的元素进行如下的划分: (i) $\mathbb{Z}_N^*$ 可以划分为相同大小的 $\mathcal{J}_N^{+1}$ 和 $\mathcal{J}_N^{-1}$(Jacobi 符号分别为 1 和-1); (ii) $\mathcal{J}_N^{+1}$ 可以划分为 $\mathcal{QR}_N$ 和 $\mathcal{QNR}_N^{+1}$, 其中 $\mathcal{QNR}_N^{+1} \stackrel{\text{def}}{=} \{x \in \mathbb{Z}_N^* \mid x \notin \mathcal{QR}_N \wedge \mathcal{J}_N(x) = +1\}$.

定义 2.3 (模二次剩余假设)

模二次剩余 (QR, quadratic residue) 假设指 $\mathcal{QR}_N$ 上的均匀分布与 $\mathcal{QNR}_N^{+1}$ 上的均匀分布计算不可区分. 我们称模二次剩余假设相对于 GenModulus 成立当且仅当对于任意 PPT 敌手:

$$|\Pr[\mathcal{A}(N, y_0) = 1] - \Pr[\mathcal{A}(N, y_1) = 1]| \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$、$\mathsf{GenModulus}(1^\kappa) \to (N, p, q)$ 和随机选取 $y_0 \in \mathcal{QR}_N, y_1 \in \mathcal{QNR}_N^{+1}$ 的随机带上.

与 QR 假设应用紧密相关的技术细节是如何对 $\mathcal{QR}_N$ 和 $\mathcal{QNR}_N^{+1}$ 进行高效的均匀采样.

- 对 $\mathcal{QR}_N$ 进行均匀采样较为简单: 仅需随机选取 $x \in \mathbb{Z}_N^*$ 再令 $y := x^2 \bmod N$ 即可. 注意到 $x^2 \bmod N$ 是一个 4-to-1 的正则函数, 因此当 $x \xleftarrow{R} \mathbb{Z}_N^*$ 时, 输出 y 服从 $\mathcal{QR}_N$ 上的均匀分布.
- 对 $\mathcal{QNR}_N^{+1}$ 进行均匀采样稍显复杂, 当 N 的分解未知时如何均匀采样未知. 我们可以借助辅助信息 $z \in \mathcal{QNR}_N^{+1}$ 完成采样, 即随机选取 $x \in \mathbb{Z}_N^*$, 输出 $y := z \cdot x^2 \bmod N$. 可以验证, 当 $x \xleftarrow{R} \mathbb{Z}_N^*$ 时, 输出 y 服从 $\mathcal{QNR}_N^{+1}$ 上的均匀分布.

注记 2.5

显然, 整数分解问题难于二次剩余判定问题, 因此整数分解假设弱于模二次剩余判定假设. 两个假设是否等价仍然未知.

定义 2.4 (模平方根假设)

模平方根 (SQR, square root) 假设指对 $\mathcal{QN}_N$ 中的随机元素求平方根是困难的. 我们称模平方根假设相对于 GenModulus 成立当且仅当对于任意 PPT 敌手:

$$\Pr[\mathcal{A}(N, y) = x \text{ s.t. } x^2 = y \bmod N] \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$、$\mathsf{GenModulus}(1^\kappa) \to (N, p, q)$ 和随机选取 $y \in \mathcal{QR}_N$ 的随机带上.

令 p 和 q 是两个互异的模 4 余 3 的素数, 则称 $N = pq$ 是 Blum 整数. 我们有以下推论:

命题 2.1

当 N 是 Blum 整数时, 则每个模 N 的二次剩余有且仅有一个平方根是二次剩余.

上述推论保证了当 N 是 Blum 整数时, 函数 $f_N \stackrel{\text{def}}{=} [x^2 \bmod N]$ 构成 $\mathcal{QR}_N$ 上的置换. 这一性质在构造加密方案时至关重要.

注记 2.6

模平方根假设等价于整数分解假设, 即在未知 N 分解的情况下求模平方根与分解 N 一样困难.

综上, 若 $A \succeq B$ 表示问题 A 难于问题 B, 则整数分解类问题的困难性关系如图 2.5所示:

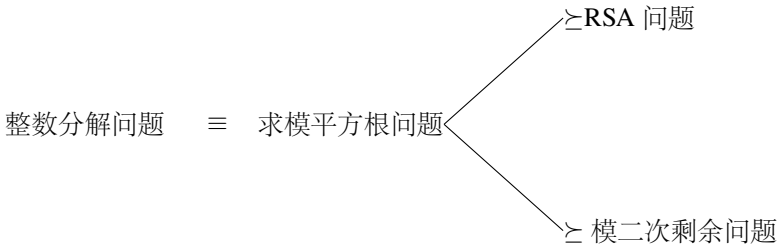


图 2.5: 整数分解类问题的困难性关系

2.3.2 离散对数类假设

离散对数类假设定义在循环群 $\mathbb{G}$ 中. 令 GenGroup 是 PPT 算法, 其以安全参数 1^κ 为输入, 输出 q 阶循环群 $\mathbb{G} = \langle g \rangle$ 的描述, 其中, q 是 κ-bit 的整数, 简记为 $(\mathbb{G}, q, g) \leftarrow \mathsf{GenGroup}(1^\kappa)$. 为了行文方便, 本书中假设 $\mathbb{G}$ 为加法群, 用 "$\cdot$" 表示群运算. 由循环群的定义可知, $\mathbb{G}$ 中的元素为 $\{g^0, g^1, \ldots, g^{q-1}\}$. 因此, 对于任意 $h \in \mathbb{G}$ 存在唯一的 $x \in \mathbb{Z}_q$ 使得 $g^x = h$, 我们称 x 是 h 相对于生成元 g 的离散对数并记为 $x = \log_g h$, 这里称其为离散对数强调其取值均为非负整数, 有别于标准算术对数的取值为实数.

定义 2.5 (离散对数假设)

离散对数 (DLOG) 问题指在平均意义下求解群元素的离散对数是困难的. 我们称离散对数假设相对于 GenGroup 成立当且仅当对于任意 PPT 敌手:

$$\Pr[\mathcal{A}(\mathbb{G}, q, g, h) = \log_g h] \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$、$\mathsf{GenGroup}(1^\kappa) \to (\mathbb{G}, q, g)$ 和随机采样 $h \in \mathbb{G}$ 的随机带上.

显然, 离散对数假设说明了 $x \mapsto g^x$ 是从 $\mathbb{Z}_q$ 到 $\mathbb{G}$ 的单向函数. 单向函数能够蕴含的密码方案有限, 下面介绍与离散对数假设相关的其它假设, 它们能够作为更多密码方案的安全基础. 这类困难假设的起源于 Diffie 和 Hellman [3] 在 1976 年的划时代论文, 后来被称为 Diffie-Hellman 假设. 为了叙述方便, 我们首先定义 DH 函数 $\mathsf{DH}_g : \mathbb{G}^2 \to \mathbb{G}$,

$$\mathsf{DH}_g(h_1, h_2) \stackrel{\text{def}}{=} g^{\log_g h_1 \cdot \log_g h_2}$$

Diffie-Hellman 类假设可细分为两类, 一类是计算性 Diffie-Hellman (CDH) 问题, 一类是判定性 Diffie-Hellman (DDH) 问题. 下面依次介绍.

定义 2.6 (CDH 假设)

CDH 问题指在平均意义下计算 DH_g 函数是困难的. 我们称 CDH 假设相对于 GenGroup 成立当且仅当对于任意 PPT 敌手:

$$\Pr[\mathcal{A}(\mathbb{G}, q, g, g^a, g^b) = g^c] \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$、$\mathsf{GenGroup}(1^\kappa) \to (\mathbb{G}, q, g)$ 和随机采样 $a, b \in \mathbb{Z}_q$ 的随机带上.

定义 2.7 (DDH 假设)

对于四元组 (g, g^a, g^b, g^c), 如果 $g^c = \mathsf{DH}_g(g^a, g^b)$ 也即 $ab = c \bmod q$, 则称其为 DH 元组. DDH 假设刻画的是随机 DH 元组和随机四元组是计算不可区分的. 我们称 DDH 假设相对于 GenGroup 成立当且仅当对于任意 PPT 敌手:

$$|\Pr[\mathcal{A}(\mathbb{G}, q, g, g^a, g^b, g^{ab}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, q, g, g^a, g^b, g^c) = 1]| \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$、$\mathsf{GenGroup}(1^\kappa) \to (\mathbb{G}, q, g)$ 和随机采样 $a, b, c \in \mathbb{Z}_q$ 的随机带上.

离散对数类问题的困难性关系如图 2.6所示:

离散对数问题 $\succeq$ CDH 问题 $\succeq$ DDH 问题

图 2.6: 离散对数类问题的困难性关系

注记 2.7

注意到任何 q 阶循环群 $\mathbb{G}$ 均与 $\mathbb{Z}_q$ 是同构的, 而 $\mathbb{Z}_q$ 上的离散对数问题是容易的. 因此在实例化循环群 $\mathbb{G}$ 必须小心审慎, 这也从一个方面说明离散对数类问题的困难性与底层代数结构的具体特性 (如群的表示) 紧密相关. 对于 $\mathbb{G}$ 的实例化, 通常既可以选择 $\mathbb{F}_{p^k}^*$ 的素数阶乘法子群, 也可以选择椭圆曲线上的素数阶乘法群. 另外强调一点, 存在这样的循环群 $\mathbb{G}$(如双线性映射群) 使得离散对数、CDH 假设成立, 而 DDH 假设不成立.

离散对数类假设还可延伸至具备双线性映射 (bilinear map) 的代数结构上.

定义 2.8 (双线性映射)

令 GenBLGroup 是 PPT 算法, 其以安全参数 1^κ 为输入, 输出 $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, g_1, g_2, e)$, 其中 $\mathbb{G}_1$、$\mathbb{G}_2$ 和 $\mathbb{G}_T$ 是三个循环群, 群阶均为素数 $q = \Theta(2^\kappa)$, g_1 和 g_2 分别是 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 的生成元, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ 可高效计算 (非退化) 的双线性映射. 令 $g_T = e(g_1, g_2)$, 则 g_T 是 $\mathbb{G}_T$ 的生成元. e 也常被称为配对 (pairing), 通常有以下三种类型.

- Type-I: $\mathbb{G}_1 = \mathbb{G}_2$;
- Type-II: $\mathbb{G}_1 \neq \mathbb{G}_2$ 且存在可高效计算的同构映射 $\psi : \mathbb{G}_2 \to \mathbb{G}_1$;
- Type-III: $\mathbb{G}_1 \neq \mathbb{G}_2$ 且 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 之间不存在可高效计算的同构映射.

根据文献 [262] 的总结, Type-I 是 "对称双线性映射", 因其结构精简、假设较弱, 学术论文中偏好使用这种类型的配对描述并证明方案; Type-II 和 Type-III 是 "非对称双线性映射", 其中 Type-III 因其效率优势明显, 是工程实现中的首选.

下面介绍判定性 Diffie-Hellman (DBDH) 问题在非对称双线性映射群上的定义:

定义 2.9 (DBDH 假设)

我们称 DBDH 假设相对于 GenBLGroup 成立当且仅当对于任意 PPT 敌手:

$$|\Pr[\mathcal{A}(g_1^a, g_1^b, g_2^c, e(g_1, g_2)^{abc}) = 1] - \Pr[\mathcal{A}(g_1^a, g_1^b, g_2^c, e(g_1, g_2)^z) = 1]| \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$、$\mathsf{GenBLGroup}(1^\kappa) \to (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, g_1, g_2, e)$, 和随机采样 $a, b, c, z \xleftarrow{R} \mathbb{Z}_q$ 的随机带上.

如果在上面公式的挑战实例中同时增加项 g_2^a, 则可得到更强的 co-DBDH 假设, 即要求分布 $(g_1^a, g_1^b, g_2^a, g_2^c, e(g_1, g_2)^{abc})$ 与分布 $(g_1^a, g_1^b, g_2^a, g_2^c, e(g_1, g_2)^z)$ 在计算意义下不可区分.

2.3.3 格类假设

1994 年, Shor [263] 给出了数论类问题 (包括整数分解类和离散对数类) 的有效量子算法. 在未来, 如果大规模量子计算机研制成功, 则数论类假设将不再成立. 迄今为止, 尚未有针对格基困难问题的有效量子算法, 通用的量子算法仅相对非量子算法有些许优势. 目前普遍的共识是格基困难问题具备抗量子安全能力, 这正是该类问题倍受关注的主要原因.

本小节中将介绍两个主要的平均意义下的格基困难问题, 短整数解问题和带误差学习问题. 需要提前说明的是, 格基困难问题的困难性与参数的选取密切相关, 因此格基问题的描述相比数论类问题要复杂得多.

Ajtai [264] 在 1996 年的开创性论文中正式提出了短整数解 (short integer solution, SIS) 问题. SIS 问题不仅可以作为所有 Minicrypt 世界中密码组件的安全基础, 包括单向函数、身份鉴别协议、数字签名, 还可以用来构造抗碰撞哈希函数. 非正式的, SIS 问题指在给定许多较大的有限加法群中随机选取的元素, 找到足够 "短" 的整系数组合使得其和是 0 是困难的. SIS 问题由以下参数刻画:

- 正整数 n 和 q, 用于刻画加法群 $\mathbb{Z}_q^n$;
- 正实数 β, 用于刻画解向量的长度;
- 正整数 m, 用于表征群元素的个数.

其中 n 是主要的参数 (如: $n \geq 100$), $q > \beta$ 通常设定为关于 n 的小多项式.

定义 2.10 (短整数解假设 ($\mathrm{SIS}_{n,q,\beta,m}$))

我们称 SIS 假设成立当且仅当对于任意 PPT 敌手:

$$\Pr[\mathcal{A}(\mathbf{a}_1, \ldots, \mathbf{a}_m) = \mathbf{z} \neq \mathbf{0} \in \mathbb{Z}^m \text{ s.t. } \sum_i^m \mathbf{a}_i z_i = \mathbf{0} \in \mathbb{Z}_q^n \wedge \|z\| \leq \beta] \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$ 和随机选取 $\mathbf{a}_i \in \mathbb{Z}_q^n$ 的随机带上.

以上定义中 m 个 $\mathbb{Z}_q^n$ 上的随机向量可以按列向量的方式组成矩阵 $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. 因此, SIS 假设实质上在要求找到函数 $f_{\mathbf{A}}(\mathbf{z}) := \mathbf{A}\mathbf{z}$ 的短整数非零向量原像是困难的.

下面简单讨论参数选取与问题困难性之间的关联:

- 如果不对 $\|z\|$ 进行限制, 那么可以轻易利用 Gaussian 消元法找到一个整数解. 同时, 我们必须要求 $\beta < q$, 否则 $\mathbf{z} = (q, 0, \ldots, 0) \in \mathbb{Z}^m$ 即构成一个合法的非平凡解.
- 注意到任何关于矩阵 $\mathbf{A}$ 的短整数解可通过补 0 平凡地延展为关于矩阵 $[\mathbf{A} \mid \mathbf{A}']$ 的解. 换言之, SIS 问题的困难性随着 m 的增大变得容易. 对应的, SIS 问题的困难性随着 n 增加变得困难.
- 向量范数界 β 和向量 $\mathbf{a}_i$ 的个数 m 必须足够大以保证解的存在性. 令 $\bar{m}$ 是大于 $n \log q$ 的最小正整数, 则我们必须有 $\beta > \sqrt{m}$ 和 $m \geq \bar{m}$. 不失一般性, 不妨假设 $m = \bar{m}$, 则存在超过 q^n 个向量 $\mathbf{x} \in \{0,1\}^m$, 根据鸽巢原理, 则必有 $\mathbf{x} \neq \mathbf{x}'$ 使得 $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}' \in \mathbb{Z}_q^n$, 从而它们的差值 $\mathbf{z} = \mathbf{x} - \mathbf{x}' \in \{0, \pm 1\}^m$ 是范数小于 β 的短整数解.
- 上述的鸽巢原理论证事实上蕴含更多深意: 函数族 $\{f_{\mathbf{A}} : \{0,1\}^m \to \mathbb{Z}_q^n\}$ 基于 SIS 假设是抗碰撞的. 若不然, 给定关于 $f_{\mathbf{A}}$ 的一对碰撞 $\mathbf{x}, \mathbf{x}' \in \{0,1\}^m$, 则立刻诱导出关于 $\mathbf{A}$ 的一个短整数解.

SIS 问题可以被理解为在以下特定 q 元 m 维整数格中的平均意义短向量问题 (short-vector problem, SVP), 该整数格的定义为:

$$\mathcal{L}^{\perp}(\mathbf{A}) \stackrel{\text{def}}{=} \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = 0 \in \mathbb{Z}_q^n\} \supseteq q\mathbb{Z}^m$$

从编码的角度理解, $\mathbf{A}$ 扮演着格/码字 $\mathcal{L}^{\perp}(\mathbf{A})$ 校验矩阵的角色. SIS 问题的困难性指对于随机选取的 $\mathbf{A}$, 找到一个短的码字是困难的.

Regev [265] 在 2005 年的开创性论文中提出了另一个平均意义下的重要格基困难问题–带误差学习问题 (learning with errors, LWE). LWE 问题与 SIS 问题互相对偶, 能够蕴含 Minicrypt 之外的密码体制.

在正式定义 LWE 问题之前, 首先引入 LWE 分布的概念. 称向量 $\mathbf{s} \in \mathbb{Z}_q^n$ 为秘密, LWE 分布 $A_{\mathbf{s},\chi}$ 定义在 $\mathbb{Z}_q^n \times \mathbb{Z}_q$ 上, 采样算法为随机选取 $\mathbf{a} \in \mathbb{Z}_q^n$, 选取 $e \leftarrow \chi$, 输出 $(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e \bmod q)$.

LWE 问题有两个版本, 其中搜索版本要求给定 LWE 采样求解秘密, 判定版本要求区分 LWE 采样和随机采样. LWE 问题由以下参数刻画:

- 正整数 n 和 q, 和 SIS 问题一样, 用于刻画加法群 $\mathbb{Z}_q^n$;
- 正整数 m 表征采样的个数, 通常选取的足够大以保证秘密的惟一性;
- $\mathbb{Z}$ 上的误差分布 χ, 通常的选取是宽度为 αq 的离散 Gaussian 分布, 其中 $\alpha < 1$ 称为相对错误率.

定义 2.11 (搜索 LWE 假设)

搜索 LWE 问题指给定 m 个 $A_{\mathbf{s},\chi}$ 的独立随机采样, 求解秘密向量 $\mathbf{s}$ 是困难的. 我们称搜索 LWE 假设成立当且仅当对于任意 PPT 敌手:

$$\Pr[\mathcal{A}(\{\mathbf{a}_i, b\}_{i=1}^m \leftarrow A_{\mathbf{s},\chi}) = \mathbf{s}] \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$、随机选取 $\mathbf{s} \in \mathbb{Z}_q^n$ 和采样 $A_{\mathbf{s},\chi}$ 的随机带上.

定义 2.12 (判定 LWE 假设)

判定 LWE 问题指区分 m 个独立采样是来自 $A_{\mathbf{s},\chi}$ 分布还是随机分布是困难的. 我们称判定 LWE 假设成立当且仅当对于任意 PPT 敌手:

$$|\Pr[\mathcal{A}(\{\mathbf{a}_i, b\}_{i=1}^m \leftarrow A_{\mathbf{s},\chi}) = 1] - \Pr[\mathcal{A}(\{\mathbf{a}_i, b\}_{i=1}^m \leftarrow U_{\mathbb{Z}_q^n \times \mathbb{Z}_q}) = 1] \leq \mathsf{negl}(\kappa)$$

上述概率建立在敌手 $\mathcal{A}$、随机选取 $\mathbf{s} \in \mathbb{Z}_q^n$ 和采样 $A_{\mathbf{s},\chi}$ 以及 $U_{\mathbb{Z}_q^n \times \mathbb{Z}_q}$ 的随机带上.

注记 2.8

LWE 问题是 LPN 问题 (learning parities with noise) 的一般化. 在 LPN 问题中, $q = 2$, χ 为 $\{0, 1\}$ 上的 Bernoulli 分布.

下面简单讨论参数选取与问题困难性之间的关联:

- 如果没有误差分布 χ, 则 LWE 问题的搜索版本和判定版本均可利用 Gaussian 消元法快速求解.
- 和 SIS 问题类似, 可以用矩阵的语言更简洁的描述 LWE 问题: (i) 将 m 个向量 $\mathbf{a}_i \in \mathbb{Z}_q^n$ 汇聚为矩阵 $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$; (ii) 将 m 个 $b_i \in \mathbb{Z}_q$ 汇聚为向量 $\mathbf{b} \in \mathbb{Z}_q^n$, 因此对于 LWE 采样我们有:

$$\mathbf{b}^t = \mathbf{s}^t \mathbf{A} + \mathbf{e}^t (\mathrm{mod} q),$$

其中 $\mathbf{e} \leftarrow \chi^m$.

LWE 问题可以被理解为在以下特定 q 元 m 维整数格中的平均意义有界距离解码问题 (bounded-distance decoding problem, BDD), 该整数格的定义为:

$$\mathcal{L}(\mathbf{A}) \stackrel{\text{def}}{=} \{\mathbf{A}^t \mathbf{s} : \mathbf{s} \in \mathbb{Z}_q^n\} + q\mathbb{Z}^m$$

从编码的角度理解, $\mathbf{A}$ 扮演着格/码字 $\mathcal{L}(\mathbf{A})$ 生成矩阵的角色. 对于 LWE 采样, $\mathbf{b}$ 与格中的惟一向量/码字相近, 搜索版本要求计算秘密向量 $\mathbf{s}$, 即根据带误差的码字进行解码. 对于随机采样, $\mathbf{b}$ 以大概率远离格 $\mathcal{L}(\mathbf{A})$ 中所有向量. SIS 问题的困难性指对于随机选取的 $\mathbf{A}$, 找到一个短的码字是困难的.

2.4 复杂性理论初步

在复杂性理论中, 困难问题 P 通常定义在 $L \subseteq X$ 上, X 是所有实例的集合, L 是 X 中满足特定性质的一个子集. 我们称 P 是可高效判定的, 如果存在确定性多项式时间的 Turing 机 M 满足:

$$x \in L \iff M(x) = 1$$

所有可高效判定问题的合集组成 $\mathcal{P}$ 复杂性类.

笔记 实例集合 X 的学术术语是词 (words), 子集 L 对应的学术术语是语言 (language). 术语来源自以下的类比: 不妨设世界上所有的词汇构成一个集合, 那么汉语、英语、法语、德语、C++ 语言、Rust 语言等多种多样的语言自然构成了这个集合的各个子集. 通常, 称语言内的元素为 Yes 实例, 语言外的元素为 No 实例.

密码学中的困难问题可以分为计算和判定两类:

- 计算类 (也称搜索类) 问题要求计算出问题的解: 如 RSA 问题、离散对数问题、计算 Diffie-Hellman 问题、短整数解问题等.
- 判定类问题要求判定是或否: 如二次剩余问题、判定 Diffie-Hellman 问题、判定 LWE 问题等.

从解空间的角度理解, 判定问题可以看做计算问题的特例, 即输出解为 1 比特. 通常, 同一个问题的计算版本难于判定版本, 对应的计算假设弱于判定假设.

困难的二元关系是对密码学中各种计算类困难问题的抽象.

定义 2.13 (二元关系 (binary relation))

令 $L \subseteq X$ 是一个 $\mathcal{NP}$ 语言. L 由二元关系 $\mathsf{R}_L : X \times W$ 定义, 其中 W 之证据集合:

$$x \in L \Leftrightarrow \exists w \in W \text{ s.t. } (x, w) \in \mathsf{R}_L$$

如果 R_L 满足如下两个性质, 则称其是困难的 (hard):

- 易采样 (easy to sample): $\exists$ PPT 算法 SampRel 对关系 R_L 进行随机采样, 其以公开参数 pp 为输入, 输出 "实例-证据" 元组 $(x, w) \in \mathsf{R}_L$.
- 难抽取 (hard to extract): $\forall$ PPT 敌手 $\mathcal{A}$:

$$\Pr[(x, \mathcal{A}(x) = w') \in \mathsf{R}_L : (x, w) \leftarrow \mathsf{SampRel}(r)] \leq \mathsf{negl}(\kappa)$$

♣

笔记 单向函数自然诱导了一个困难的二元关系. 易采样的性质由单向函数的定义域可高效采样和单向函数可高效求值两点保证, 难抽取的性质由单向函数的单向性保证.

问题任务: 计算/搜索　　解空间: $\{0,1\}^{\mathsf{poly}(\kappa)}$

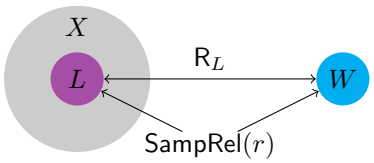


图 2.7: 计算类困难问题图示

子集成员判定问题 (subset membership problem, SMP) 则是密码学中各种判定类问题的抽象.

定义 2.14 (子集成员判定问题)

令 $L \subset X$ 是一个语言, 定义以 3 个 PPT 采样算法:

- $\mathsf{SampAll}(r)$: 输出 X 中的随机元素.
- $\mathsf{SampYes}(r)$: 输出 L 中的随机元素, 即随机 Yes 实例.
- $\mathsf{SampNo}(r)$: 输出 $X \backslash L$ 中的随机元素, 即随机 No 实例.

SMP 问题有两种类型:

- Type 1: $U_X \approx_c U_L$
- Type 2: $U_{X\setminus L} \approx_c U_L$

♣

注记 2.9

定义 $\rho = |L|/|X|$ 为语言 L 相对于 X 的密度. 容易证明:

- 当 $\rho = \mathsf{negl}(\kappa)$ 时: Type 1 $\iff$ Type 2
- 当 ρ 已知时: Type 2 $\Rightarrow$ Type 1
  - 归约的方法是对给定分布和 U_L 分布根据 ρ 进行加权重构: 如果给定分布是 $U_{X\setminus L}$, 则重构结果 U_X; 如果给定分布是 U_L, 则重构结果仍为 U_L. 因此, Type 2 的实例可以归约到 Type 1 的实例.

♠

问题任务: 区分/判定　　解空间: $\{0,1\}$

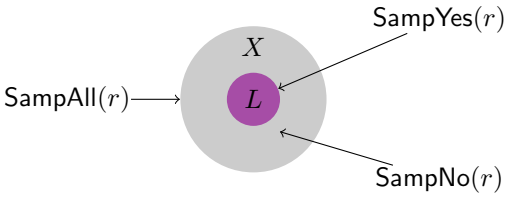


图 2.8: 判定类困难问题图示

2.5 信息论工具

2.5.1 熵的概念

Shannon 在 1948 年开创了信息论这一全新领域, 为编码与密码奠定了理论基础. 信息论关注的重点是 "消息" 和它们在 (有噪) 信道中的传播. 容易直观理解的是, 消息所包含的信息量取决于消息令人感到意外的程度 (如果消息平平无奇, 那么它包含的信息为 0). 相比而言, 稍显反直觉的是随机消息包含最多的信息.

信息论中最重要的概念是熵 (entropy). 熵量化了期望的意义下消息包含的信息量, 单位通常是比特. 从概率论的角度看, 熵是对随机变量不确定性的测度. 以下令 X 是定义在 Ω 上的随机变量.

定义 2.15 (熵)

X 的熵刻画了平均意义下 X 取值的 (不) 可预测性:

$$\mathsf{H}(X) = -\sum_{\omega\in\Omega} \Pr[X=\omega]\log\Pr[X=\omega]$$

♣

注记 2.10

一个消息的熵就是消息所包含信息的比特数, 用编码的语言刻画, 就是编码该消息所需的最短比特数.

♠

密码方案/协议的安全性分析均是针对恶意敌手展开的. 敌手单次正确预测某随机变量 (如私钥) 值的概率与密码方案/协议的安全性紧密相关. 显然, 敌手的最佳策略是猜测最大似然值. 在本章中, 我们用大写字母 X 表示随机变量, 用小写字母 x 表示 X 的取值, 用花体字母 $\mathcal{X}$ 表示 X 的支撑集.

一个随机变量 X 的最大可预测性是 $\max_{\omega\in\Omega}\Pr[X=\omega]$. 最大可预测性对应最小熵 (min-entropy), 严格定义如下:

定义 2.16 (最小熵)

X 的最小熵刻画了 X 的最大可预测性:

$$\mathsf{H}_\infty(X) = -\log\left(\max_{\omega\in\Omega}\Pr[X=\omega]\right)$$

♣

注记 2.11

最小熵可以看做 "最坏情形"(worst-case) 的熵.

♠

在很多场景中, 随机变量 X 与另一随机变量 Y 相关, 并且敌手知晓 Y 的取值. 因此, Dodis 等 [266] 引入了平均最小熵 (average min-entropy) 来刻画 $X|Y$ 的 (不) 可预测性:

$$\tilde{\mathsf{H}}_\infty(X|Y) = -\log\left(\mathbb{E}_{y\leftarrow Y}\left[2^{\mathsf{H}_\infty(X|Y=y)}\right]\right) = -\log\left(\mathbb{E}_{y\leftarrow Y}\left[\max_{\omega\in\Omega}\Pr[X=\omega|Y=y]\right]\right) \tag{2.1}$$

以下浅释平均最小熵的定义直觉. 考虑一对变量 X 和 Y(两者可能相关). 如果敌手知晓 Y 的取值 y, 则 X 在敌手视角中的可预测性是 $\max_x\Pr[X=x|Y=y]$. 在平均的意义下 (对 Y 做期望), 敌手成功预测 X 的概率为 $\mathbb{E}_{y\leftarrow Y}[\max_x\Pr[X=x|Y=y]]$.

平均最小熵的定义在对 Y 做加权平均的前提下 (Y 的取值不受敌手控制) 测度 X 最坏情形下的可预测性 (敌手知晓 y 后对 X 的预测是恶意行为). 一个微妙的细节是平均最小熵的定义 (2.1) 先对预测成功的概率做期望后再取对数, 那能否交换 log 和 $\mathbb{E}$ 的次序呢? 定义平均最小熵 $\mathbb{E}_{y\leftarrow Y}[\mathsf{H}_\infty(X|Y=y)]$ 是否合理呢? 交换次序后的定义失去了原本的意义. 考虑以下的例子, 令 X 和 Y 都是定义在 $\Omega=\{0,1\}^{1000}$ 上的随机变量, Y 是 Ω 上的随机分布, 当 Y 的取值 y 的首 bit 为 0 时, X 的取值与 y 相同, 否则随机分布. 因此对于 Y 的半数取值 y, $\mathsf{H}_\infty(X|Y=y)=0$, 对另外半数取值, $\mathsf{H}_\infty(X|Y=y)=1000$, 所以 $\mathbb{E}_{y\leftarrow Y}[\mathsf{H}_\infty(X|Y=y)]=500$. 然而, 声称 X 具有 500 比特的安全性显然不符合逻辑. 事实上, 知晓 Y 取值 y 的敌手直接输出 y, 既能够以大于 $1/2$ 的概率猜对 X 的取值. 平均最小

熵标准的定义准确刻画了至少 1/2 的可预测性, 因为 $\tilde{\mathsf{H}}_\infty(X|Y)$ 略小于 1. 我们也可以从数学的角度解释如下, $\mathbb{E}$ 是线性算子, 而 log 是非线性算子, 因此次序交换后意义不同.

平均最小熵和最小熵之间存在何种关系呢? Dodis 等 [266] 证明了如下的 Chaining Lemma, 建立了两者之间的关系, 给出了平均最小熵的一个下界.

引理 2.2 (Chaining Lemma)

令 X、Y 和 Z 是三个随机变量 (可任意相关), 其中 Y 的支撑集包含至多 2^r 个元素. 我们有 $\tilde{\mathsf{H}}_\infty(X|(Y,Z)) \geq \mathsf{H}_\infty(X|Z) - r$. 特别的, 当 Z 为空时, 上述不等式简化为: $\tilde{\mathsf{H}}_\infty(X|Y) \geq \mathsf{H}_\infty(X) - r$.

♡

2.5.2 随机性提取

随机性是密码学的主旋律, 几乎所有已知密码方案/协议都离不开均匀随机采样. 然而, 均匀无偏的完美信源并不易得, 很多场景下存在的是有偏的弱信源. 如何在信源有偏的情况下进行均匀随机采样呢? 这就是随机性提取器所要完成的工作.

定义 2.17 (强随机性提取器)

令 X 是最小熵 $\mathsf{H}_\infty(X) \geq n$ 的随机变量, $\mathsf{ext} : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ 是一个可高效计算的函数. 我们称 ext 是对信源 X 的 (n, ϵ)-强随机性提取器当且仅当以下成立:

$$\Delta((\mathsf{ext}(X, S), S), (Y, S)) \leq \epsilon,$$

其中 S 是定义在 $\mathcal{S}$ 上的均匀随机变量, Y 是定义在 $\mathcal{Y}$ 上的均匀随机变量.

♣

类比于平均最小熵和最小熵之间的关系, 当信源 X 与另一变量 Z 相关时, 我们需要引入平均强随机性提取器来对信源 X 进行萃取.

定义 2.18 (平均强随机性提取器)

令 (X, Z) 是满足约束 $\tilde{\mathsf{H}}_\infty(X|Z) \geq n$ 的任意变量对, $\mathsf{ext} : \mathcal{X} \times \mathcal{S} \to \mathcal{Y}$ 是一个可高效计算的函数. 我们称 ext 是对信源 X 的平均意义 (n, ϵ)-强随机性提取器当且仅当以下成立:

$$\Delta((\mathsf{ext}(X, S), S, Z), (Y, S, Z)) \leq \epsilon,$$

其中 S 是定义在 $\mathcal{S}$ 上的均匀随机变量, Y 是定义在 $\mathcal{Y}$ 上的均匀随机变量.

♣

Dodis 等 [266] 的 Conditional Leftover Hash Lemma(条件剩余哈希引理) 证明了任何强随机性提取性在适当的参数设定下都是平均强随机性提取器. 作为一个特例, Dodis 等证明了任何一族一致哈希函数 (universal hash functions) 都是平均强随机性提取器.

引理 2.3 (Leftover Hash Lemma)

令 X 和 Z 是满足约束 $\tilde{\mathsf{H}}_\infty(X|Z) \geq n$ 的任意变量对, $\mathcal{H} = \{h_s : \mathcal{X} \to \mathcal{Y}\}_{s \leftarrow S}$ 是一族一致哈希函数. 那么当 $n \geq \log|\mathcal{Y}| + 2\log(1/\epsilon)$ 时, $\mathsf{ext}(x, s) := h_s(x)$ 是 (n, ϵ)-平均强随机性提取器.

♡

2.6 密码组件

本章将简要介绍后续章节内容中所涉及的基本密码组件.

2.6.1 身份加密方案

身份加密方案 [267] 是一种能够以用户任意身份 (如 Email 地址、姓名、身份证号等) 作为加密公钥的新型公钥加密技术, 能够简化传统公钥加密技术中的密钥管理复杂性. 下面给出 IBE 方案的定义及安全模型.

定义 2.19 (身份加密方案)

身份加密方案 IBE *由以下* 5 *个* PPT *算法组成:*

- $\mathsf{Setup}(1^\kappa)$: *以安全参数* 1^κ *为输入, 输出系统公开参数* pp. *其中* pp *定义了系统的主公钥空间* MPK、*主私钥空间* MSK、*用户身份空间* I、*私钥空间* SK、*明文空间* M *和密文空间* C.
- $\mathsf{KeyGen}(pp)$: *以公开参数* pp *为输入, 输出主公/私钥对* (mpk, msk), *其中主公钥* mpk *公开, 主私钥* msk *由密钥生成中心秘密保存.*
- $\mathsf{Extract}(msk, id)$: *以主私钥* msk *和用户身份* $id \in ID$ *为输入, 输出用户私钥* sk_{id}.
- $\mathsf{Encrypt}(mpk, id, m)$: *以主公钥* mpk、*用户身份* id *和消息* $m \in M$ *为输入, 输出消息* m *在身份* id *下加密的一个密文* $c \in C$.
- $\mathsf{Decrypt}(sk_{id}, c)$: *以用户私钥* sk_{id} *和密文* c *为输入, 输出消息* m' *或* $\perp$ *表示解密失败.*

♣

正确性. 对于任意 $pp \leftarrow \mathsf{Setup}(1^\kappa)$, $(mpk, msk) \leftarrow \mathsf{KeyGen}(pp)$, 任意身份 $id \in I$ 和私钥 $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$, 任意明文 $m \in M$, 则有 $m = \mathsf{Decrypt}(sk_{id}, \mathsf{Encrypt}(mpk, id, m))$.

安全性. 令 $\mathcal{A}$ 是攻击身份加密方案安全性的敌手, 定义其优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}} = \left| \Pr \left[\beta = \beta' : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (mpk, msk) \leftarrow \mathsf{KeyGen}(pp); \\ (id^*, m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ext}}, \mathcal{O}_{\mathsf{decrypt}}}(pp, mpk); \\ \beta \xleftarrow{R} \{0,1\}, c^* \leftarrow \mathsf{Enc}(mpk, id^*, m_\beta); \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ext}}, \mathcal{O}_{\mathsf{decrypt}}}(c^*); \end{array} \right] - \frac{1}{2} \right|$$

$\mathcal{O}_{\mathsf{ext}}$ 是私钥询问谕言机, 其在接收到身份 id 的询问后输出 $sk_{id} \leftarrow \mathsf{Extract}(msk, id)$. $\mathcal{O}_{\mathsf{decrypt}}$ 是解密询问谕言机, 其在接收到身份 id 和密文 c 的询问后输出 $m \leftarrow \mathsf{Decrypt}(sk_{id}, c)$. 为了避免定义无意义, $\mathcal{A}$ 在第二阶段不得向 $\mathcal{O}_{\mathsf{ext}}$ 询问挑战身份 id^*, 也不能向 $\mathcal{O}_{\mathsf{decrypt}}$ 询问 (id^*, c^*). 如果任意 PPT 敌手 $\mathcal{A}$ 在上述安全试验中的优势函数均为可忽略的, 则称 IBE 方案是 IND-CCA 安全的. 如果不允许 $\mathcal{A}$ 访问解密谕言机, 则称 IBE 方案是 IND-CPA 安全的. 此外, 还可以类似地定义两种弱化的安全性, 包括 OW-CCA/OW-CPA 安全性和 sIND-CCA/sIND-CPA 安全性. 其中, OW-CCA/OW-CPA 的敌手目标是从一个随机密文中恢复出原始消息, 而 sIND-CCA/sIND-CPA 安全性要求敌手在看到 mpk 前要承诺攻击的身份 id^*.

2.6.2 非交互式密钥协商方案

在非交互式密钥协商 (non-interactive key exchange, NIKE) 方案中, 用户各自在公告板 (public bulletin board) 上发布一条消息, 所有用户均可阅读公告板上的消息, 且任意 n 个用户均可协商出一个共同的会话密钥, 且该会话密钥对于 n 个用户外的群体是隐藏的. 经典的 Diffie-Hellman NIKE [3] 基于 DDH 假设解决了 $n = 2$ 的情形, Joux [268] 使用了双线性映射解决了 $n = 3$ 的情形. 对于任意的正整数 n, Boneh 和 Silverberg [269] 基于多重线性映射给出了首个黑盒构造; Boneh 和 Zhandry [270] 使用不可区分程序混淆给出了一个非黑盒构造; 最近 Alamati 等 [271] 基于可复合输入同态弱伪随机函数 (composable input homomorphic weak PRF) 给出了另一个黑盒构造.

在 NIKE 的安全性研究中, Cash 等 [272] 提出了公钥场景下的安全性模型——CKS 模型. CKS 模型既允许敌手获得诚实生成的公钥, 也允许敌手注册非诚实生成的公钥 (用于刻画敌手不知晓公钥所对应私钥的情形).

这种非诚实密钥注册 (dishonest key registration, DKR) 设定刻画了实际的 PKI 运作流程, 即证书中心 (certificate authority, CA) 在签发证书时并不要求用户提交私钥的知识证明. Freire 等 [273] 提出了 CKS-light 模型, 并考察了诚实密钥注册 (honest key registration, HKR) 设定, 即不允许敌手注册非诚实生成的公钥.

以下我们给出多方 NIKE 的算法定义, 并将 CKS-light 安全模型 [273] 从两方推广到多方. 与已有定义不同的是, 我们的定义消除了算法中的身份, 同时允许多个用户拥有同一公钥, 使得定义本身更加简洁、易于使用.

定义 2.20 (多方非交互式密钥协商)

多方非交互式密钥协商方案包含以下 3 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa, n)$: 以安全参数 1^κ 和 n 为输入, 输出系统公开参数 pp.
- $\mathsf{KeyGen}(pp)$: 以系统公开参数 pp 为输入, 输出密钥对 (pk, sk).
- $\mathsf{ShareKey}(sk_i, S)$: 以私钥 sk_i 和 n 个公钥的集合 S 为输入, 若 $pk_i \in S$ 则可使用 sk_i 导出 S 对应的会话密钥 k_S.

♣

正确性. 我们要求 S 中任意用户均可导出相同的会话密钥, 即对于任意的 n 个用户群组 S 和 $pk_i \in S$, 均有:

$$\mathsf{ShareKey}(sk_i, S) = k_S$$

其中 sk_i 是 pk_i 对应的私钥.

一致性. 正确性仅刻画了算法 ShareKey 在 S 中所有公钥均来自公钥空间时的行为. 我们引入一致性, 刻画当 S 中存在一个公钥空间中的元素——如 pk_i 时, 算法 $\mathsf{ShareKey}(sk_j, S)$ 的输出对于所有 $j \neq i$ 仍然相同. 一致性是一个温和的性质, 若公钥空间可高效识别 (efficiently recognizable) 时, 该性质都可自然满足. 所有已知的 n 方 NIKE 方案 [268, 270, 271] 均满足一致性.

安全性. 令 $\mathcal{A}$ 是攻击 NIKE 安全性的敌手, 定义其优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}} = \left| \Pr \left[b = b' : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa, n); \\ S \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{regH}}, \mathcal{O}_{\mathsf{regC}}, \mathcal{O}_{\mathsf{reveal}}}(pp); \\ k_0^* \leftarrow k_S, k_1^* \xleftarrow{\mathrm{R}} K; \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{regC}}, \mathcal{O}_{\mathsf{reveal}}}(k_b^*); \end{array} \right] - \frac{1}{2} \right|.$$

$\mathcal{O}_{\mathsf{regH}}$ 是诚实用户注册谕言机, 刻画的是敌手可观察到诚实用户公钥的情形. $\mathcal{A}$ 可以询问 $\mathcal{O}_{\mathsf{regH}}$ 谕言机 n 次. 当 $\mathcal{A}$ 发起该类询问时, 挑战者运行算法 KeyGen 生成密钥对 (pk, sk), 将 (pk, sk) 记录到初始为空的列表 L_{honest} 中, 返回 pk 给 $\mathcal{A}$. S 记录了 $\mathcal{A}$ 询问 $\mathcal{O}_{\mathsf{regH}}$ 所得的公钥集合, $\mathcal{A}$ 将在其中选定攻击目标. $\mathcal{O}_{\mathsf{regC}}$ 是腐化用户注册谕言机, 刻画的是 CA 在签发证书时不检测公钥真实性的情形. $\mathcal{A}$ 可以询问 $\mathcal{O}_{\mathsf{regC}}$ 谕言机多项式次, 每次以不同的 pk 作为输入. 当 $\mathcal{A}$ 发起该类询问时, 挑战者将 $(pk, \perp)$ 记录到初始为空的列表 L_{corrupt} 中. $\mathcal{O}_{\mathsf{reveal}}$ 是腐化会话密钥谕言机, 刻画的是敌手可获得特定群组会话密钥的情形. $\mathcal{A}$ 可询问 $\mathcal{O}_{\mathsf{reveal}}$ 多项式次, 每次以 n 个公钥组成的集合为输入, 集合中至少有一个公钥是腐化的, 其余是诚实的. 挑战者返回对应的会话密钥. 为了避免定义平凡, $\mathcal{A}$ 不允许向 $\mathcal{O}_{\mathsf{reveal}}$ 询问关于 S 的会话密钥.

如果任意 PPT 敌手在上述安全试验中的优势函数均为可忽略的, 则称 NIKE 方案在 DKR 情形下是 CKS-light 安全的; 如果任意 PPT 敌手在禁止访问 $\mathcal{O}_{\mathsf{regC}}$ 和 $\mathcal{O}_{\mathsf{reveal}}$ 的安全试验中的优势函数均为可忽略的, 则称 NIKE 方案在 HKR 情形下时 CKS-light 安全的.

2.6.3 伪随机函数及其扩展

Goldreich 等 [274] 提出的伪随机函数 (pseudorandom functions, PRF) 是现代密码学中核心概念, 具有极为广泛的应用. 以下给出伪随机函数的定义和安全性.

定义 2.21 (伪随机函数)

伪随机函数包含以下 3 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出系统公开参数 pp, 刻画了一族带密钥函数 $F : K \times D \to R$, 其中 K 是密钥空间, D 是定义域, R 是值域.
- $\mathsf{KeyGen}(pp)$: 以系统公开参数 pp 为输入, 选取随机密钥 $k \xleftarrow{R} K$.
- $\mathsf{Eval}(k, x)$: 以密钥 $k \in K$ 和 $x \in D$ 为输入, 输出函数值 $y \leftarrow F(k, x)$. 为了叙述方便, $F(k, x)$ 和 $F_k(x)$ 常交替使用.

♣

伪随机性. 令 $\mathcal{A}$ 是攻击伪随机函数安全性的敌手, 定义其优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}} = \left| \Pr \left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ k \leftarrow \mathsf{KeyGen}(pp); \\ \beta \leftarrow \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ror}}(\beta,\cdot)}(\kappa); \end{array} \right] - \frac{1}{2} \right|.$$

$\mathcal{O}_{\mathsf{ror}}(\beta, \cdot)$ 是由 β 控制的真实或随机谕言机 (real-or-random oracle), $\mathcal{O}_{\mathsf{ror}}(0, x) := F_k(x)$, $\mathcal{O}_{\mathsf{ror}}(1, x) := \mathsf{H}(x)$ (这里 H 从 $D \to R$ 的函数空间中随机选择). $\mathcal{A}$ 可以自适应的访问 $\mathcal{O}_{\mathsf{ror}}(\beta, \cdot)$ 多项式次. 如果任意 PPT 敌手 $\mathcal{A}$ 在上述安全游戏中优势均是可忽略的, 则称 F 是伪随机的.

若在上述的安全试验中, 将 $\mathcal{O}_{\mathsf{ror}}(\beta, \cdot)$ 的输入由敌手 $\mathcal{A}$ 任意选取变为挑战者随机选取, 标准伪随机性将弱化为弱伪随机性, 此时称 F 是弱伪随机的. 在一些应用场景中, 弱伪随机函数就足够了.

注记 2.12 (真随机函数的模拟)

当 $D \to R$ 的函数空间很大 (如双重指数空间) 时, 无法对其进行高效随机采样, 因此在这种情形下真随机函数无法高效实例化. 幸好, 总可以通过懒惰模拟 (lazy simulation) 的方式有效模拟出真随机函数, 即对 $\mathsf{H} \xleftarrow{R} \{f : D \to R\}$ 的谕言机访问: 维护初始化为空的输入输出对的列表, 当敌手询问新鲜输入时, 随机在 R 中采样输出并将输入输出对插入列表, 否则返回列表中相应的输出保持回答的前后一致性.

♠

标准的伪随机函数不支持密钥代理, 函数求值是 "完全或无" 方式:

- 拥有 k, 则可对定义域内的所有输入计算函数值.
- 不拥有 k, 则伪随机性隐含了无法对定义域中的任意输入求值.

在 2013 年, 三组研究人员 [275, 276, 277] 几乎同时独立的提出了受限伪随机函数 (constrained PRF) 的概念. 在受限伪随机函数中, 密钥拥有者可以派生出主密钥 k 的受限密钥, 受限密钥仅能对定义域内的部分输入求值, 其他输入的输出仍伪随机.

定义 2.22 (受限伪随机函数)

受限伪随机函数包含以下 4 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出系统公开参数 pp, 刻画了一族带密钥函数 $F : K \times D \to R$, 其中 K 是密钥空间, D 是定义域, R 是值域. pp 还包含了一个集合系统 $\mathcal{S} \subset 2^D$, 即 D 幂集的一个子集.
- $\mathsf{KeyGen}(pp)$: 以系统公开参数 pp 为输入, 选取随机密钥 $k \xleftarrow{R} K$.
- $\mathsf{Constrain}(k, S)$: 以主密钥 k 和 $S \in \mathcal{S}$ 为输入, 输出受限密钥 k_S.
- $\mathsf{Eval}(k/k_S, x)$: 以密钥 k 或受限密钥 k_S 和 $x \in D$ 为输入, 当第一输入为 k 时输出 $F_k(x)$, 当第一输入为 k_{x^*} 时, 如果 $x \in S$ 则输出 $F_k(x)$, 否则输出 $\perp$.

♣

笔记 集合系统可以进一步泛化为电路族 $\mathcal{C} = \{c : D \to \{0,1\}\}$: 受限密钥 k_c 可以对所有满足 c 的输入求值, 即当 $c(x) = 1$ 时, $\mathsf{Eval}(k_c, x) = F_k(x)$.

注记 2.13

受限伪随机函数存在平凡的构造，即令受限密钥 $k_S = \{F_k(x)\}_{x\in S}$. 在平凡的构造中，$k_S$ 的尺寸与 S 的尺寸线性相关. 为了排除平凡的构造，我们要求 k_S 是紧致的，即对于任意 $S \in \mathcal{S}$，均有 $|k_S| = \kappa^{O(1)}$.

♠

受限伪随机函数的正确性保证了 k_S 可以对 S 中的输入正确求值. 伪随机性则保证了即使给定 k_S，对于 $x \notin S$ 之外的输入 $F_k(x)$ 仍然是伪随机的.

受限伪随机性. 令 $\mathcal{A}$ 是攻击受限伪随机函数安全性的敌手，定义其优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}} = \left| \Pr\left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ k \leftarrow \mathsf{KeyGen}(pp); \\ x^* \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{constrain}}, \mathcal{O}_{\mathsf{eval}}}(pp); \\ y_0^* \xleftarrow{R} R, y_1^* \leftarrow F_k(x^*); \\ \beta \leftarrow \{0,1\}; \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{constrain}}, \mathcal{O}_{\mathsf{eval}}}(state, y_\beta^*); \end{array} \right] - \frac{1}{2} \right|.$$

$\mathcal{O}_{\mathsf{constrain}}$ 是受限密钥询问谕言机，以 $S \in 2^D$ 为输入，输出 k_S. $\mathcal{O}_{\mathsf{eval}}$ 是求值谕言机，以 $x \in D$ 为输入，输出 $y \leftarrow F_k(x)$. $\mathcal{A}$ 在访问 $\mathcal{O}_{\mathsf{constrain}}$ 和 $\mathcal{O}_{\mathsf{eval}}$ 的限制是不可平凡的计算 $F_k(x^*)$. 如果任意 PPT 敌手 $\mathcal{A}$ 在上述安全游戏中优势均是可忽略的，则称 F 是相对于 $\mathcal{S} \in 2^D$ 是受限伪随机的.

Sahai 和 Waters [35] 引入了受限伪随机函数的特例——可穿孔伪随机函数 (puncturable PRF). 在可穿孔伪随机函数中，$\mathcal{S}$ 限定为单元素集合，从而仅支持 “全除一”(all-but-one, ABO) 方式的密钥派生: 主密钥持有方可从主密钥 k 中导出 k_{x^*}，可对除了 x^* 外的所有输入求值. 可穿孔伪随机函数的正式定义如下:

定义 2.23 (可穿孔伪随机函数 (puncturable PRF))

可穿孔伪随机函数包含以下 4 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入，输出公开参数 pp，其中包含了函数 $F : K \times X \to Y$ 的描述和电路族 $\mathcal{C} = \{f_{x^*} : X \to \{0,1\}\}_{x^*\in X}$ 的描述. $f_{x^*}(\cdot)$ 的具体定义是 $f_{x^*}(x) = \neg x^* \stackrel{?}{=} x$. 为了表述简洁，以下在不引起混淆的情况下使用 x^* 表征 f_{x^*}.
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入，随机采样密钥 $k \xleftarrow{R} K$.
- $\mathsf{Puncture}(k, x^*)$: 以密钥 k 和 $x^* \in X$ 为输入，输出受限密钥 k_{x^*}.
- $\mathsf{Eval}(k/k_{x^*}, x)$: 以密钥 k 或穿孔密钥 k_{x^*} 和 $x \in X$ 为输入，当第一输入为 k 时输出 $F_k(x)$，当第一输入为 k_{x^*} 时，如果 $x \neq x^*$ 时输出 $F_k(x)$，否则输出 $\perp$.

♣

可穿孔伪随机函数要求对于没有被受限密钥覆盖的输入，其输出仍然是伪随机的. 陈等 [73] 证明了可穿孔伪随机函数存在以下两种等价的安全性定义.

选择伪随机性. 定义敌手 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left| \Pr\left[\beta = \beta' : \begin{array}{l} (x^*, state) \leftarrow \mathcal{A}_1(\kappa); \\ pp \leftarrow \mathsf{Setup}(1^\kappa); \\ k \leftarrow \mathsf{KeyGen}(pp); \\ k_{x^*} \leftarrow \mathsf{Puncture}(k, x^*); \\ \beta \xleftarrow{R} \{0,1\}, y_0^* = F_k(x^*), y_1^* \xleftarrow{R} Y; \\ \beta' \leftarrow \mathcal{A}_2(state, k_x^*, y_\beta^*); \end{array} \right] - \frac{1}{2} \right|.$$

如果任意 PPT 敌手 $\mathcal{A}$ 在如图 2.9所示的安全游戏中优势函数均为可忽略函数，则称可穿孔伪随机函数是选择伪随机的.

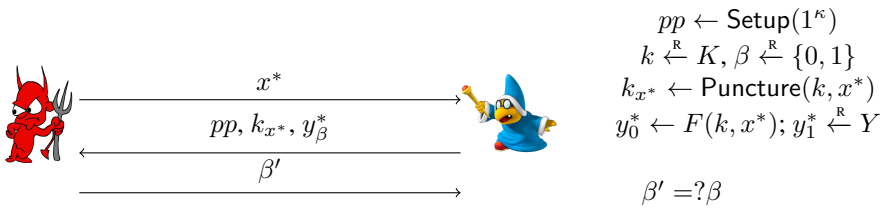


图 2.9: 可穿孔伪随机函数的选择伪随机性安全游戏

弱伪随机性. 定义敌手 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ 的优势函数如下:

$$\left| \Pr\left[\beta = \beta' : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ k \leftarrow \mathsf{KeyGen}(pp), x^* \xleftarrow{R} X; \\ k_{x^*} \leftarrow \mathsf{Puncture}(k, x^*); \\ \beta \xleftarrow{R} \{0,1\}, y^*_0 = F_k(x^*), y^*_1 \xleftarrow{R} Y; \\ \beta' \leftarrow \mathcal{A}(pp, x^*, k^*_x, y^*_\beta); \end{array} \right] - \frac{1}{2} \right|.$$

如果任意的 PPT 敌手 $\mathcal{A}$ 在如图 2.10所示的安全游戏中优势函数均为可忽略函数, 则称可穿孔伪随机函数是弱伪随机的.

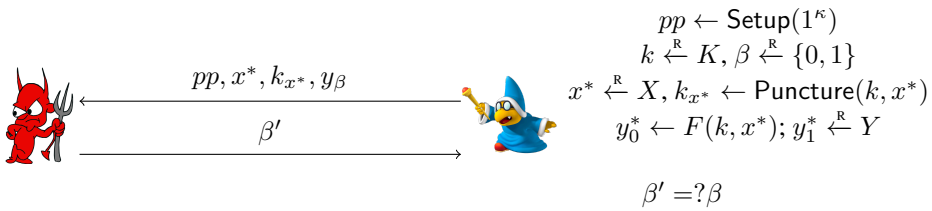


图 2.10: 可穿孔伪随机函数的弱伪随机性安全游戏

注记 2.14 (可穿孔伪随机函数的构造)

可穿孔伪随机函数可以通过 GGM 树形伪随机函数自然得出, k_{x^*} 由 x^* 到根节点路径上所有的兄弟节点组成, 如图 2.11所示. 因此可穿孔伪随机函数仍属于 Minicrypt 范畴.

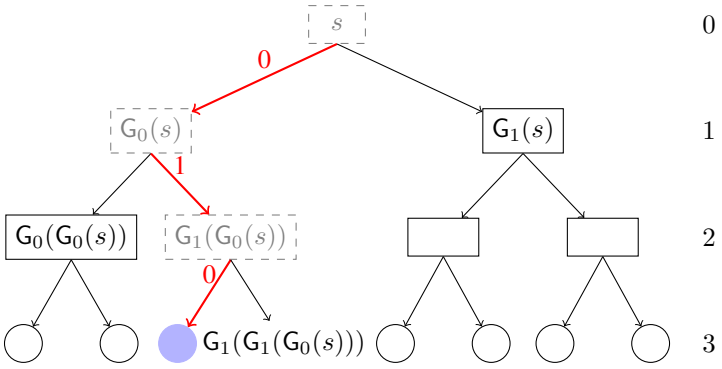


图 2.11: 基于 GGM 伪随机函数的可穿孔伪随机函数: $k_{010} = \{\mathsf{G}_1(\mathsf{G}_1(\mathsf{G}_0(s))), \mathsf{G}_0(\mathsf{G}_0(s)), \mathsf{G}_1(s)\}$

第三章 经典公钥加密方案回顾

章前概述

内容提要

- 公钥加密的定义与安全性
- 整数分解类经典方案
- 离散对数类经典方案
- 格类经典方案

本章开始介绍公钥密码学的第二部分内容—公钥加密. 3.1节定义了公钥加密的算法组成和安全性，3.2节介绍了基于整数分解类难题的经典公钥加密方案，3.3节介绍了基于离散对数类难题的经典公钥加密方案，3.4节介绍了基于格类难题的经典公钥加密方案.

3.1 公钥加密的定义与基本安全模型

雄兔脚扑朔, 雌兔眼迷离. 双兔傍地走, 安能辨我是雄雌?

— 南北朝《木兰诗》

3.1.1 公钥加密方案

公钥加密的概念由 Diffie 和 Hellman [3] 在 1976 年的划时代论文中正式提出, 其与对称加密的最大不同在于每个用户自主生成一对密钥, 公钥用于加密、私钥用于解密, 发送方仅需知晓接收方的公钥即可向接收方发送密文.

定义 3.1 (公钥加密方案)

公钥加密方案由以下 4 个 PPT 算法组成:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出系统公开参数 pp, 其中 pp 通常包含公钥空间 PK、私钥空间 SK、明文空间 M 和密文空间 C 的描述. 该算法由可信第三方生成并公开, 系统中的所有用户共享, 所有算法均将 pp 作为输入. 当上下文明确时, 常常为了行文简洁省去 pp.
- $\mathsf{KeyGen}(pp)$: 以系统公开参数 pp 为输入, 输出一对公/私钥对 (pk, sk), 其中公钥公开, 私钥秘密保存.
- $\mathsf{Encrypt}(pk, m; r)$: 以公钥 $pk \in PK$、明文 $m \in M$ 为输入, 输出密文 $c \in C$.
- $\mathsf{Decrypt}(sk, c)$: 以私钥 $sk \in SK$ 和密文 $c \in C$ 为输入, 输出明文 $m \in M$ 或者 $\perp$ 表示密文非法. 解密算法通常为确定性算法.

注记 3.1

系统公开参数的内容并没有严格的规定, 通常的设定是包含所有用户可共享的信息, 在一些例子中可能退化为 $\perp$. 如对于 RSA 公钥加密方案, 明文空间和密文空间均与公钥相关, 所以应由算法 KeyGen 输出. 读者需要根据具体情况, 对定义做灵活变通, 切勿固步自封.

正确性. 该性质保证公钥加密的功能性, 即使用私钥可以正确恢复出对应公钥加密的密文. 正式的, 对于任意明文 $m \in M$, 有:

$$\Pr[\mathsf{Decrypt}(sk, \mathsf{Encrypt}(pk, m)) = m] = 1 - \mathsf{negl}(\kappa). \tag{3.1}$$

公式 (3.1)的概率建立在 $\mathsf{Setup}(1^\kappa) \to pp$、$\mathsf{KeyGen}(pp) \to (pk, sk)$ 和 $\mathsf{Encrypt}(pk, m) \to c$ 的随机带上. 如果上述概率严格等于 1, 则称公钥加密方案满足完美正确性.

注记 3.2

通常基于数论假设的公钥加密方案满足完美正确性, 而格基方案由于底层困难问题的误差属性, 解密算法存在可忽略的误差.

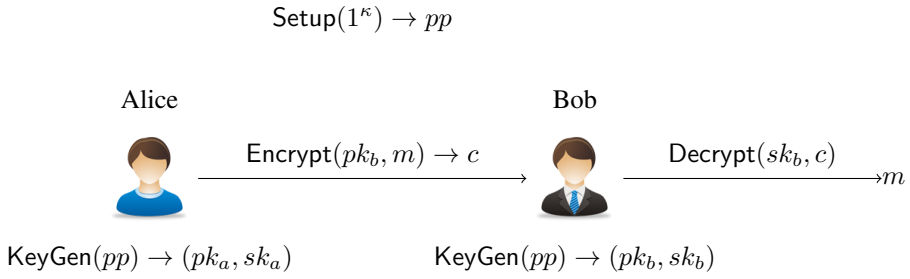


图 3.1: 公钥加密方案示意图

安全性. 定义公钥加密方案敌手 $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa)=\left|\Pr\left[\beta'=\beta:\begin{array}{l} pp\leftarrow\mathsf{Setup}(1^{\kappa});\\ (pk,sk)\leftarrow\mathsf{KeyGen}(pp);\\ (m_0,m_1,state)\leftarrow\mathcal{A}_1^{\mathcal{O}_{\mathsf{decrypt}}}(pp,pk);\\ \beta\xleftarrow{\mathrm{R}}\{0,1\};\\ c^*\leftarrow\mathsf{Encrypt}(pk,m_\beta);\\ \beta'\leftarrow\mathcal{A}_2^{\mathcal{O}_{\mathsf{decrypt}}}(pp,pk,state,c^*);\end{array}\right]-\frac{1}{2}\right|.$$

在上述定义中, $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ 表示敌手 $\mathcal{A}$ 可划分为两个阶段, 划分界线是接收到挑战密文 c^* 前后, $state$ 表示 $\mathcal{A}_1$ 向 $\mathcal{A}_2$ 传递的信息, 记录部分攻击进展. $\mathcal{O}_{\mathsf{decrypt}}$ 表示解密谕言机, 其在接收到密文 c 的询问后输出 $\mathsf{Decrypt}(sk,c)$. 如果任意的 PPT 敌手 $\mathcal{A}$ 在上述游戏中的优势函数均为可忽略函数, 则称公钥加密方案是 IND-CPA 安全的; 如果任意的 PPT 敌手在阶段 1 可自适应访问 $\mathcal{O}_{\mathsf{decrypt}}$ 的情形下仍仅具有可忽略优势, 则称公钥加密方案是 IND-CCA1 安全的; 如果任意的 PPT 敌手在阶段 1 和阶段 2 均可自适应访问 $\mathcal{O}_{\mathsf{decrypt}}$ 的情形下仍仅具有可忽略优势, 则称公钥加密方案是 IND-CCA2 或 IND-CCA 安全的.

以下阐述公钥加密安全性定义的一些细微之处:

- 自适应的含义是敌手的攻击行为可根据学习到的知识动态调整, 如我们称敌手能够自适应的访问解密谕言机指敌手可以根据历史询问结果发起新的询问. 简而言之, 自适应性极大的增强了敌手的攻击能力.
- IND-CCA 安全性远强于 IND-CCA1 和 IND-CPA 安全性, 这是因为敌手可以在观察到挑战密文 c^* 后有针对性的发起更加有威胁的解密询问.
- (m_0,m_1) 由敌手任意选择, 从而巧妙精准的刻画了密文不泄漏明文任何一比特信息的直觉.
- 为了避免定义无意义, 在 IND-CCA 的安全游戏中禁止敌手在第二阶段向 $\mathcal{O}_{\mathsf{decrypt}}(\cdot)$ 询问挑战密文 c^*.

笔记 对于密码方案, 给出恰当的安全性定义非常重要: 一方面安全性定义必须足够强以刻画现实中存在的攻击, 另一方面安全性定义不能过强使得无法构造满足其的密码方案. 公钥加密的安全性定义是逐渐演化的.

上世纪 70 年代, Diffie 和 Hellman 提出了公钥加密的概念, 随后 Riverst、Shamir 和 Adi 构造出了首个公钥加密方案——RSA 加密. 在这一阶段, 公钥加密的安全性仅具备符合直觉的单向性, 即在平均意义下从密文中恢复出明文是计算困难的. 到了上世纪 80 年代, 人们逐渐认识到单向性并不能满足应用需求, 这是因为对于单向安全的公钥加密方案, 敌手有可能从密文恢复出明文的部分信息, 而在应用中, 由于数据来源的多样性和不确定性, 明文的每一比特都可能包含关键的机密信息 (比如股票交易指令中的“买”或“卖”).

1982 年, Goldwasser 和 Micali [5] 指出单向安全的不足, 提出了语义安全性 (semantic security). 语义安全性的直观含义是密文对敌手求解明文没有帮助. 严格定义颇为精妙, 定义的形式是基于模拟的, 即敌手掌握密文的视角可以由一个 PPT 的模拟器在计算意义下模拟出来. 语义安全性可以看做 Shannon 完美安全性在计算意义的推广放松, 然而在论证的时候稍显笨重.

Goldwasser 和 Micali 给出了另一个等价的定义 (等价性的证明参见 Dodis 和 Ruhl 的短文 [278]), 即选择明文攻击下的不可区分性 (IND-CPA, indistinguishability against chosen-plaintext attack). IND-CPA 安全定义的直觉是密文在计算意义下不泄漏明文的任意一比特信息, 即对任意两个明文对应的密文分布是计算不可区分的, 其中选择明文攻击刻画了公钥公开特性使得任意敌手均可通过自行加密获得任意明文对应密文这一事实. 使用 IND-CPA 安全进行安全论证相比语义安全要便捷很多, 因此被广为采用.

注意到 IND-CPA 安全仅考虑被动敌手, 即敌手只窃听信道上的密文. 1990 年, Naor 和 Yung [9] 认为敌手有能力发起一系列主动攻击, 比如重放密文、修改密文等, 进而提出选择密文攻击 (CCA, chosen-ciphertext attack) 刻画这一系列主动攻击行为, 即敌手可以自适应的获取指定密文对应的明文. Naor 和 Yung 考虑了两种选择密文攻击, 一种是弱化版本, 称为午餐时间攻击 (lunch-time attack), 含义是敌手只能在极短的时间窗口 (收到挑战密文之前) 进行选择密文攻击; 另一种是标准版本, 敌手可以长时间窗口 (收到挑战密文前后) 进行选择密文攻击.

1998 年, Bleichenbacher [10] 展示了针对 PKCS#1 标准中公钥加密方案的有效选择密文攻击, 实证了关于选择密文安全的研究并非杞人忧天. Shoup [279] 进一步深入探讨了选择密文安全的重要性与必要性. 从此, IND-CCA 安全成为了公钥加密方案的事实标准.

公钥加密的有用性质

本小节介绍公钥加密两个常见的有用性质, 分别是同态和可重随机化.

同态. 公钥加密方案的正确性隐式保证了解密算法自然诱导出从密文空间 C 到明文空间 M 的一个映射 $\phi = \mathsf{Dec}(sk, \cdot)$, 如图 3.2所示. 如果 ϕ 具备同态性, 则第三方可对密文进行相应的公开计算, 得到的密文与对明文施加同样计算所得结果对应. 正式的, 令 $\mathcal{C} = \{f\}$ 是从 $M^n \to M$ 的某个电路族, 其中 n 是正整数; Eval 为密文求值算法, 以公钥 pk、$f \in \mathcal{C}$ 和密文向量 $\mathbf{c} = (c_1, \ldots, c_n)$ 为输入, 输出 $c' \in C$, 记作 $c' \leftarrow \mathsf{Eval}(pk, f, \mathbf{c})$. 如果对于任意 $f \in \mathcal{C}$ 和任意明文 $\mathbf{m} = (m_1, \ldots, m_n) \in M^n$ 以下公式成立:

$$\Pr\left[\mathsf{Dec}(sk, c') = f(\mathbf{m}) : \begin{array}{l}(pk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa);\\ \mathbf{c} \leftarrow \mathsf{Enc}(pk, \mathbf{m});\\ c' \leftarrow \mathsf{Eval}(pk, f, \mathbf{c});\end{array}\right] = 1$$

则称公钥加密方案是 $\mathcal{C}$-同态的, $\mathcal{C}$ 刻画了同态所支持的公开计算类型. 两种常见的同态类型如下:

- 部分同态 (partially homomorphic): 不失一般性, 若明文空间 M 为加法群, 密文空间 C 为乘法群, 若 $\mathcal{C}$ 仅包含 $M^2 \to M$ 的群运算, 则称加密方案是部分同态或者加法同态的. 此时同态性刻画如下:

$$\Pr\left[\mathsf{Dec}(sk, c_1 \cdot c_2) = f(m_1, m_2) : \begin{array}{l}(pk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa);\\ c_1 \leftarrow \mathsf{Enc}(pk, m_1), c_2 \leftarrow \mathsf{Enc}(pk, m_2);\end{array}\right] = 1$$

- 全同态 (fully homomorphic): 若 $\mathcal{C}$ 包含了 $M^n \to M$ 的所有多项式时间可计算函数, 则称方案是全同态的.

笔记 几乎所有公钥加密方案都构建在代数性质良好的结构上, 且大部分方案均天然满足部分同态, 如:

- RSA [4]: 支持无限次的模乘运算
- ElGamal [37]: 支持无限次的群加运算
- Goldwasser–Micali [5]: 支持无限次的 XOR 运算
- Benaloh [280]: 支持无限次的模加运算
- Paillier [281]: 支持无限次的模加运算
- Sander-Young-Yung [238]: 支持 NC^1 电路运算
- Boneh-Goh-Nissim [237]: 支持无限次的加法运算和一次乘法运算
- Ishai-Paskin [282]: 支持多项式规模的 branching program

在 RSA 公钥加密方案横空出世仅一年后, Rivest、Adleman 和 Dertouzos [283] 即提出了全同态公钥加密的概念. 直到 31 年后, 才由 Gentry [239] 通过引入理想格构给出首个全同态加密方案的构造. 自此突破之后, 全同态加密迅猛发展, 理论成果百花齐放, 效率不断提升, 成为了隐私保护技术中重要且实用的密码学工具. 感兴趣的读者请参阅 Halevi 的综述文章 [284].

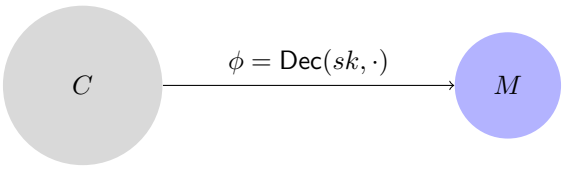


图 3.2: 密文空间至明文空间的同态映射

可重随机化. 若给定公钥加密方案的公钥 pk 和密文 c, 生成新的密文 c', 使得 c 和 c' 的解密结果相同, 且 c' 的分布与真实密文分布统计不可区分, 则称该公钥加密方案是可公开重随机化的 (re-randomizable), 简称可重随机化. 正式的, 若公钥加密方案存在 PPT 算法 $\mathsf{ReRand}(pk, c) \to c'$, 且满足以下的解密正确性和密文不可区分性, 则称其可重随机化.

- **解密正确性.** 对于任意 $pp \leftarrow \mathsf{Setup}(1^\kappa)$, 任意 $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$ 任意 $m \in M$, 任意 $c \leftarrow \mathsf{Encrypt}(pk, m)$ 以及任意 $c' \leftarrow \mathsf{ReRand}(pk, c)$ 均有: $\mathsf{Decrypt}(sk, c) = \mathsf{Decrypt}(sk, c')$.

- **密文不可区分性.** 对于任意 $pp \leftarrow \mathsf{Setup}(1^\kappa)$, 任意 $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$ 任意 $m \in M$, 分布 $c \leftarrow \mathsf{Encrypt}(pk, m)$ 与分布 $c' \leftarrow \mathsf{ReRand}(pk, c)$ 相同.

上述完美的解密正确性和密文不可区分性可根据应用场景适当放宽, 允许解密存在可忽略误差或密文统计接近.

注记 3.3 (公钥加密的安全性与功能性权衡)

对于密码方案和协议, 安全性、功能性和效率之间通常存在权衡关系 (trade-off). 对于公钥加密方案, IND-CPA 安全性与同态性可以共存, 而更强的 IND-CCA 安全性与同态性之间就存在冲突, 无法兼得. 在现实世界中应用公钥加密方案时, 需根据应用场景的具体需求在安全性和功能效率之间做出恰当的选择, 切不可教条.

3.1.2 密钥封装机制

主流的公钥加密方案基于数论或者格基困难问题构造. 基于数论问题的公钥加密方案因需要进行高精度算术运算导致加解密速率较低, 基于格基困难问题的公钥加密方案存在公钥和密文尺寸较大的问题. 而对称加密方案因其功能简单, 仅需异或等逻辑运算即可完成, 且硬件支持良好 (如定制的指令), 因此在较公钥加密具有较大的性能优势, 在加密长明文的场景下更为显著.

如何解决公钥加密在加密长消息时的性能短板呢? 解决思路是混合加密 (hybrid encryption), 朴素的实现方式是 PKE+SKE, 如图 3.3所示:

1. 发送方首先随机选择对称密钥 k, 调用公钥加密算法用接收方的公钥 pk 加密 k 得到 c, 再调用对称加密算法用 k 加密明文 m 得到 c', 最终的密文 (c, c').
2. 接收方在接收到密文 (c, c') 后, 首先使用私钥 sk 解密 c 恢复对称密钥 k, 再使用 k 解密 c'.

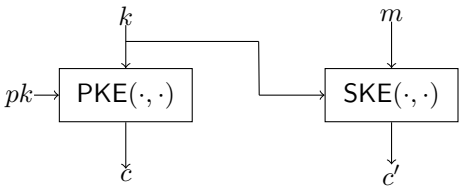


图 3.3: 混合加密: PKE+SKE

混合加密方法既保留了公钥加密的功能性, 同时性能几乎与对称加密相当, 因此是公钥加密加密长明文时的通用范式. Cramer 和 Shoup [285] 观察到公钥加密在混合加密范式中起到的关键作用是发送方向接收方传输对称密钥, 而传递的方式并非必须是加解密. 基于该观察, Cramer 和 Shoup 提出了 "密钥封装-数据封装" 范式, 简称为 KEM+DEM(key/data-encapsulation mechanism), 该范式可以看作是混合加密的另一种实现方式, 如图 3.4所示. 顾名思义, KEM+DEM 范式包含 KEM 和 DEM 两个组件, DEM 可以粗略的等同为对称加密, KEM 是该范式的核心. 简言之, KEM 与 PKE 的不同在于发送方不再先显式选择对称密钥再加密, 而是封装一个随机的对称密钥.

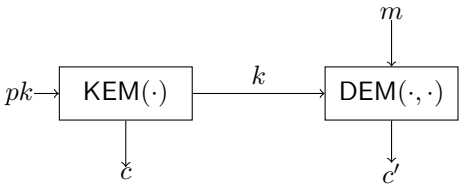


图 3.4: 混合加密: KEM+DEM

定义 3.2 (密钥封装机制)

KEM 由以下的 4 个 PPT 算法组成:

- $\mathsf{Setup}(1^\kappa)$: 系统生成算法以安全参数 1^κ 为输入, 输出系统公开参数 pp, 其中 pp 包含公钥空间 PK、私钥空间 SK、对称密钥空间 K 和密文空间 C 的描述. 该算法由可信第三方生成并公开, 系统中的所有用户共享, 所有算法均将 pp 作为输入. 当上下文明确时, 常常为了行文简洁省去 pp.
- $\mathsf{KeyGen}(pp)$: 密钥生成算法以系统公开参数 pp 为输入, 输出一对公/私钥对 (pk, sk), 其中公钥公开, 私钥秘密保存.
- $\mathsf{Encaps}(pk; r)$: 封装算法以公钥 $pk \in PK$ 为输入, 输出对称密钥 $k \in K$ 和封装密文 $c \in C$.
- $\mathsf{Decaps}(sk, c)$: 解封装算法以私钥 $sk \in SK$ 和密文 $c \in C$ 为输入, 输出对称密钥 $k \in K$ 或者 $\perp$ 表示封装密文非法. 解封装算法通常为确定性算法.

♣

注记 3.4

在 KEM 中, 对称密钥 k 起到的作用是在发送方和接收方之间建立安全的会话信道, 因此也常称为会话密钥.

♠

正确性. 该性质保证 KEM 的功能性, 即使用私钥可以正确恢复出封装密文所封装的会话密钥. 正式的, 对于任意会话密钥 $k \in K$, 有:

$$\Pr[\mathsf{Decaps}(sk, c) = k : (c, k) \leftarrow \mathsf{Encaps}(pk)] = 1 - \mathsf{negl}(\kappa). \tag{3.2}$$

公式 (3.2)的概率建立在 $\mathsf{Setup}(1^\kappa) \to pp$、$\mathsf{KeyGen}(pp) \to (pk, sk)$ 和 $\mathsf{Encaps}(pk) \to (c, k)$ 的随机带上. 如果上述概率严格等于 1, 则称 KEM 方案满足完美正确性.

安全性. 定义 KEM 敌手 $\mathcal{A}$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left| \Pr\left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ (c^*, k_0^*) \leftarrow \mathsf{Encaps}(pk), k_1^* \xleftarrow{\mathrm{R}} K; \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{decaps}}}(pp, pk, c^*, k_\beta^*); \end{array} \right] - \frac{1}{2} \right|.$$

在上述定义中, $\mathcal{O}_{\mathsf{decaps}}$ 表示解封装谕言机, 其在接收到密文 c 的询问后输出 $\mathsf{Decaps}(sk, c)$. 如果任意的 PPT 敌手 $\mathcal{A}$ 在上述游戏中的优势函数均为可忽略函数, 则称 KEM 方案是 IND-CPA 安全的; 如果任意的 PPT 敌手在可自适应访问 $\mathcal{O}_{\mathsf{decaps}}$ 的情形下仍仅具有可忽略优势, 则称 KEM 方案是 IND-CCA 安全的.

下面给出安全性定义的一些注记:

- KEM 的安全游戏中分阶段定义敌手不再必要, 因为挑战密文的生成不受敌手控制, 正是这点不同使得 KEM 的安全性定义要比 PKE 的安全性定义简单.
- 为了避免定义无意义, 在 IND-CCA 的安全游戏中禁止敌手向 $\mathcal{O}_{\mathsf{decaps}}$ 询问挑战密文 c^*.

KEM 组件产生了随机密钥 k 及其封装 c, DEM 组件则是利用随机密钥 k 对消息 m 进行高效的加密, 在实践中, 常利用具有恰当安全性的对称加密方案对其进行实例化.

定义 3.3 (数据封装机制)

DEM 由以下 4 个 PPT 算法组成:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出系统公开参数 pp, 其中 pp 包含对称密钥空间 K、明文空间 M 和密文空间 C 的描述. 该算法由可信第三方生成并公开, 系统中的所有用户共享, 所有算法均将 pp 作为输入. 当上下文明确时, 常常为了行文简洁省去 pp.
- $\mathsf{KeyGen}(pp)$: 以系统公开参数 pp 为输入, 输出对称密钥 k. 该密钥生成算法通常是从对称密钥空间 K 中均匀采样作为输出.

- $\mathsf{Encrypt}(k, m)$: 加密算法以对称密钥 $k \in K$ 和明文 $m \in M$ 为输入, 输出密文 $c \in C$.
- $\mathsf{Decrypt}(k, c)$: 以对称密钥 $k \in K$ 和密文 $c \in C$ 为输入, 输出 $\perp$ 表示密文非法. 解密算法通常为确定性算法.

♣

正确性. 该性质保证 DEM 的功能性, 即使用密钥可以正确恢复出密文所加密的明文. 正式的, 对于任意密钥 $k \in K$ 以及任意的 $m \in M$, 有:

$$\Pr[\mathsf{Decrypt}(k, c) = m : c \leftarrow \mathsf{Encrypt}(k, m)] = 1 - \mathsf{negl}(\kappa). \tag{3.3}$$

公式 (3.3)的概率建立在 $\mathsf{Setup}(1^\kappa) \to pp$、$\mathsf{KeyGen}(pp) \to k$ 和 $\mathsf{Encrypt}(k, m) \to c$ 的随机带上. 如果上述概率严格等于 1, 则称 DEM 方案满足完美正确性.

安全性. 定义 DEM 敌手 $\mathcal{A}$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left| \Pr \left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ k \leftarrow \mathsf{KeyGen}(pp); \\ (m_0, m_1, state) \leftarrow \mathcal{A}_1(pp); \\ \beta \xleftarrow{R} \{0,1\}; c^* \leftarrow \mathsf{Encrypt}(k, m_\beta); \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{decrypt}}}(pp, state, c^*); \end{array} \right] - \frac{1}{2} \right|.$$

在上述定义中, $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ 表示敌手 $\mathcal{A}$ 可划分为两个阶段, 划分界线是接收到挑战密文 c^* 前后, $state$ 表示 $\mathcal{A}_1$ 向 $\mathcal{A}_2$ 传递的信息, 记录部分攻击进展. $\mathcal{O}_{\mathsf{decrypt}}$ 表示解密谕言机, 其在接收到密文 c 的询问后输出 $\mathsf{Decrypt}(k, c)$. 为了避免定义平凡, 禁止 $\mathcal{A}_2$ 向 $\mathcal{O}_{\mathsf{decrypt}}$ 询问挑战密文 c^*. 如果任意的 PPT 敌手 $\mathcal{A}$ 在上述游戏中的优势函数均为可忽略函数, 则称 DEM 方案是 IND-CPA 安全的; 如果任意的 PPT 敌手 $\mathcal{A}$ 在第 2 阶段可自适应访问 $\mathcal{O}_{\mathsf{decrypt}}$ 的情形下仍仅具有可忽略优势, 则称 DEM 方案是 IND-CCA 安全的.

注记 3.5

DEM 的 IND-CCA 安全性定义与 PKE 定义的细微区别是禁止敌手在第 1 阶段询问 $\mathcal{O}_{\mathsf{decrypt}}$.

♠

下面正式给出图 3.4所示的 KEM+DEM 混合加密范式构造.

构造 3.1 (KEM+DEM 混合加密范式)

- $\mathsf{Setup}(1^\kappa)$: 系统生成算法以安全参数 1^κ 为输入, 输出系统公开参数 pp, 其中 pp 包含对公钥空间 PK、私钥空间 SK、对称密钥空间 K、明文空间 M 和密文空间 C_1, C_2 的描述. 该算法由可信第三方生成并公开, 系统中的所有用户共享, 所有算法均将 pp 作为输入. 当上下文明确时, 常常为了行文简洁省去 pp.
- $\mathsf{KeyGen}(pp)$: 调用 $\mathsf{KEM.KeyGen}(pp)$ 输出 (pk, sk) 作为密钥对.
- $\mathsf{Encrypt}(pk, m)$: 调用 $\mathsf{KEM.Encaps}(pk)$ 得到对称密钥 k 及其封装 c_1, 再调用 $\mathsf{DEM.Encrypt}(k, m)$ 对 m 加密得到 c_2, 输出 $c := (c_1, c_2)$.
- $\mathsf{Decrypt}(sk, c)$: 解析 $c = (c_1, c_2)$, 调用 $\mathsf{KEM.Decaps}(sk, c_1)$ 得到封装的对称密钥 k, 如果解封装出错, 则输出 $\perp$. 再调用 $\mathsf{DEM.Decrypt}(k, c_2)$ 得到明文 m, 如果解密错误, 则输出 $\perp$, 否则以 m 作为最终输出.

♣

构造 3.1所得 PKE 的安全性与 KEM 和 DEM 的安全性有关, 且不难通过混合论证技术进行证明 [285]:

- 如果 KEM 和 DEM 均具有为 IND-CPA 安全性, 则上述的混合加密方案为 IND-CPA 安全的.
- 如果 KEM 和 DEM 均具有为 IND-CCA 安全性, 则上述的混合加密方案为 IND-CCA 安全的.

注记 3.6

KEM-DEM 范式是常见的 IND-CCA 安全 PKE 方案的构造方法, 但是需要注意, 该构造仅在一般的意义下需要 KEM 和 DEM 均是 IND-CCA 安全的, 对于具体的 KEM 和 DEM 方案, 有可能通过更弱安全性的 KEM 和 DEM 即可以构造 IND-CCA 安全的 PKE 方案, 并且通常更弱安全性的 KEM/DEM 构造将会更为高效, 从而整体的 PKE 方案将会更为高效, 例如著名的 Kurosawa-Desmedt KEM 及其对应的 PKE 方案的设计框架 [286].

3.1.3 两类混合加密范式的比较

PKE+SKE 以及 KEM+DEM 两种混合方法的共性都是首先生成对称密钥, 再利用对称密钥加密明文, 因此效率方面的差异体现在第一阶段. PKE+SKE 范式的非对称部分是先选择一个随机的密钥 k, 再使用 PKE 对其加密得到 c, 而 KEM+DEM 范式的非对称部分是两步并做一步完成. 如果使用 PKE+SKE 范式, 密文 c 必然存在密文扩张, 这是由概率加密的本质决定的; 而如果使用 KEM+DEM 的方法, 密文 c 相比 k 可能不存在扩张, 原因是此时 c 是对 k 的封装, 而非加密. 综上, 使用 KEM 代替 PKE, 不仅能够缩减整体密文尺寸, 也能够提升效率.

注记 3.7

通常 KEM 要比 PKE 构造简单, 这是因为 KEM 可以看作功能受限的 PKE, 因为其只允许加密随机的明文.

相比效率提升, KEM+DEM 的理论价值更大. 首先, KEM+DEM 范式实现了对 PKE 的功能解耦, 将 PKE 中的非对称内核抽取出来凝练为 KEM, 意义如下:

- KEM+DEM 范式极大简化了 PKE 的可证明安全. 我们只需证明 KEM 和 DEM 满足一定性质即可. 对比安全模型即可发现, 对于 PKE 有 CPA/CCA1/CCA 三个依次增强的安全性, 而 KEM 只有 CPA/CCA 两个依次增强的安全性. 最关键的是: 在 PKE 中敌手对挑战密文 c^* 有一定的控制能力, 而 KEM 中 c^* 完全由挑战者控制, 这一区别使得 KEM 安全证明中的归约算法更容易设计.
- KEM+DEM 范式有助于简化 PKE 的设计. 该范式将 PKE 的设计任务简化为对应的 KEM, 在后面的章节中可以看到, 在设计高等级安全的 PKE 时, 仅需设计满足相应安全性的 KEM 即可.
- KEM+DEM 范式有助于洞悉 PKE 本质. 该范式揭示了构造 PKE 的核心机制在于构造 KEM. 后续的章节揭示了 KEM 的本质是公开可求值的伪随机函数, 是伪随机函数在 minicrypt 中的对应. 认识到这一点后, 不仅可以将几乎所有公钥加密的构造统一在同一框架下, 还可以将 SKE 和 PKE 的构造在伪随机函数的视角下实现高度统一.

注记 3.8

目前, 所有已知格基 KEM 的构造方式均为“先采样随机会话密钥、再使用 PKE 加密会话密钥”的方式, 显得迂回笨重, 如何设计精巧纯粹的格基 KEM 是很有挑战意义的研究课题.

3.2 基于数论问题的经典方案

3.2.1 Goldwasser-Micali PKE

Goldwasser 和 Micali [236] 在 1984 年基于 QR 假设构造出首个可证明安全的公钥加密方案. 该方案仅能加密一比特消息, 设计的思想可类比编码: 当明文为 0 时, 随机选取二次剩余元素作为密文; 当明文为 1 时, 随机选取 Jacobi 符号为 +1 的非二次剩余元素作为密文.

构造 3.2 (Goldwasser-Micali PKE)

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 生成全局公开参数 pp, 包含对明文空间 $M = \{0,1\}$ 的描述.
- $\mathsf{KeyGen}(pp)$: 从 pp 中解析出 1^κ, 运行 $\mathsf{GenModulus}(1^\kappa) \to (N,p,q)$, 随机选取 $z \in \mathcal{QNR}_N^{+1}$, 输出公钥 $pk = (N,z)$ 和私钥 $sk = (p,q)$.
- $\mathsf{Encrypt}(pk,m)$: 以公钥 $pk = (N,z)$ 和明文 $m \in \{0,1\}$ 为输入, 随机选择 $x \xleftarrow{R} \mathbb{Z}_N^*$, 输出密文 $c = z^m \cdot x^2 \bmod N$.
- $\mathsf{Decrypt}(sk,c)$: 以私钥 $sk = (p,q)$ 和密文 c 为输入, 利用私钥判定 c 是否是模 N 的二次剩余. 若是, 输出 0; 否则输出 1.

♣

Goldwasser-Micali PKE 的正确性显然, 安全性由以下定理保证.

定理 3.1

如果 QR 假设成立, 那么 Goldwasser-Micali PKE 是 IND-CPA 安全的.

♡

证明 令 S_i 表示敌手在 Game_i 中成功概率. 以游戏序列的方式组织证明如下:

Game_0: 该游戏是标准的 IND-CPA 游戏, 挑战者 $\mathcal{CH}$ 和敌手 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 运行 $\mathsf{Setup}(1^\kappa)$ 生成公开参数 pp, 同时运行 $\mathsf{KeyGen}(pp)$ 生成公私钥对 $pk = (N,z)$ 和 $sk = (p,q)$. $\mathcal{CH}$ 将 (pp,pk) 发送给 $\mathcal{A}$.
- 挑战: $\mathcal{A}$ 选择 $m_0, m_1 \in \mathbb{G}$ 并发送给 $\mathcal{CH}$. $\mathcal{CH}$ 选择随机比特 $\beta \in \{0,1\}$, 随机选择 $x \in \mathbb{Z}_N^*$, 计算 $c^* = z^{m_\beta} \cdot x^2 \bmod N$ 并发送给 $\mathcal{A}$.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. $\mathcal{A}$ 成功当且仅当 $\beta' = \beta$.

根据定义, 我们有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game_1: 与 Game_0 的唯一不同在于密钥对的生成方式, $\mathcal{CH}$ 将 pk 中元素 z 的选取由 Jacobi 符号为 +1 的随机非二次剩余元素切换为随机二次剩余元素. 在 Game_1 中, 无论 m_β 是 0 还是 1, 密文分布均是 $\mathcal{QR}_N$ 上的均匀分布, 完美掩盖了 β 的信息. 因此, 即使对于拥有无穷计算能力的敌手, 我们也有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_1] - 1/2| = 0$$

引理 3.1

如果 QR 假设成立, 那么对于任意 PPT 敌手我们均有 $|\Pr[S_0] - \Pr[S_1]| \le \mathsf{negl}(\kappa)$.

♡

证明 证明的思路是反证. 若存在 PPT 敌手 $\mathcal{A}$ 在 Game_0 和 Game_1 中成功的概率之差不可忽略, 则可构造出 PPT 算法 $\mathcal{B}$ 打破 QR 困难问题. 令 $\mathcal{B}$ 的 QR 挑战实例为 (N,z), $\mathcal{B}$ 的目标是区分挑战实例 z 选自 $\mathcal{QNR}_N^{+1}$ 还是 $\mathcal{QR}_N$ 上的均匀分布. 为此 $\mathcal{B}$ 扮演 IND-CPA 游戏中的挑战者与 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{B}$ 根据它的挑战实例生成 pp, 令 $pk = (N,z)$, 将 (pp,pk) 发送给 $\mathcal{A}$.
- 挑战: $\mathcal{A}$ 选择 $m_0, m_1 \in \mathbb{G}$ 并发送给 $\mathcal{B}$. $\mathcal{B}$ 随机选择 $\beta \xleftarrow{R} \{0,1\}$, 随机选取 $x \in \mathbb{Z}_N^*$ 设置 $c^* = z^{m_\beta} \cdot x^2$ 并发送给 $\mathcal{A}$.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. 如果 $\beta' = \beta$, $\mathcal{B}$ 输出 1.

对上述交互分析可知, 如果 $z \xleftarrow{R} \mathcal{QNR}_N^{+1}$, 那么 $\mathcal{B}$ 完美的模拟了 Game_0; 如果 $z \xleftarrow{R} \mathcal{QR}_N$, 那么 $\mathcal{B}$ 完美的模拟了 Game_1. 因此, $\mathcal{B}$ 解决 QR 挑战的优势 $\mathsf{Adv}_{\mathcal{B}}(\kappa) = |\Pr[S_0] - \Pr[S_1]|$. 如果 QR 假设成立, 我们有 $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\kappa)$.

综上, 定理得证. □

3.2.2 Rabin PKE

令 N 为 Blum 整数, 即 $N = p \cdot q$ 的素因子 p, q 均模 4 余 3. Rabin [287] 在 1979 年基于 SQR 假设构造出 $\mathcal{QR}_N$ 上的单向陷门置换 $f_N \stackrel{\text{def}}{=} [x^2 \bmod N]$, 称为 Rabin TDP. 可以证明, 最低有效位 (lsb, least significant bit) 函数是 Rabin TDP 的 hardcore 谓词. 基于 Rabin TDP, 可以构造公钥加密方案如下:

构造 3.3 (Rabin PKE)

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 生成全局公开参数 pp, 包含对明文空间 $M = \{0,1\}$ 的描述.
- $\mathsf{KeyGen}(pp)$: 从 pp 中解析出 1^κ, 运行 $\mathsf{GenModulus}(1^\kappa) \to (N, p, q)$, 其中 N 是 Blum 整数. 输出公钥 $pk = N$ 和私钥 $sk = (p, q)$.
- $\mathsf{Encrypt}(pk, m)$: 以公钥 $pk = N$ 和明文 $m \in \{0,1\}$ 为输入, 随机选择 $x \xleftarrow{R} \mathcal{QR}_N$, 计算 $c_0 = x^2 \bmod N$, 计算 $c_1 = m \oplus \mathsf{lsb}(x)$, 输出 $c = (c_0, c_1)$ 作为密文.
- $\mathsf{Decrypt}(sk, c)$: 以私钥 $sk = (p, q)$ 和密文 $c = (c_0, c_1)$ 为输入, 计算 x 满足 $x^2 = c_0 \bmod N$, 输出 $m' = c_1 \oplus \mathsf{lsb}(x)$.

♣

Rabin PKE 的正确性由 $f_N \stackrel{\text{def}}{=} [x^2 \bmod N]$ 是陷门置换这一事实保证, IND-CPA 安全性由陷门置换的单向性保证.

3.3 基于离散对数类问题的经典方案

3.3.1 ElGamal PKE

1985 年, ElGamal [37] 基于 Diffie-Hellman 非交互式密钥协商协议构造了 ElGamal PKE 方案. 该方案设计简洁精巧, 对后续的研究有深远的影响.

构造 3.4 (ElGamal PKE)

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 运行 $\mathsf{GenGroup}(1^\kappa) \to (\mathbb{G}, q, g)$, 输出 pp 包含循环群描述, 同时包含对公钥空间 $PK = \mathbb{G}$、私钥空间 $SK = \mathbb{Z}_q$、明文空间 $M = \mathbb{G}$ 和密文空间 $C = \mathbb{G}^2$.
- $\mathsf{KeyGen}(pp)$: 随机选取 $sk \in \mathbb{Z}_q$ 作为私钥, 计算公钥 $pk := g^{sk}$.
- $\mathsf{Encrypt}(pk, m)$: 以公钥 pk 和明文 $m \in \mathbb{G}$ 为输入, 随机选择 $r \xleftarrow{R} \mathbb{Z}_q$, 计算 $c_0 = g^r, c_1 = pk^r \cdot m$, 输出密文 $c = (c_1, c_2) \in C$.
- $\mathsf{Decrypt}(sk, c)$: 以私钥 sk 和密文 $c = (c_0, c_1)$ 为输入, 输出 $m' := c_1 / c_0^{sk}$.

♣

正确性. 以下公式 3.4说明方案具有完美正确性:

$$m' = c_1 / c_0^{sk} = pk^r \cdot m / (g^r)^{sk} = m \tag{3.4}$$

定理 3.2

如果 DDH 假设成立, 那么 ElGamal PKE 是 IND-CPA 安全的.

♡

证明 令 S_i 表示敌手在 Game_i 中成功的概率. 以游戏序列的方式组织证明如下:

Game_0: 该游戏是标准的 IND-CPA 游戏, 挑战者 $\mathcal{CH}$ 和敌手 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 运行 $\mathsf{Setup}(1^\kappa)$ 生成公开参数 pp, 同时运行 $\mathsf{KeyGen}(pp)$ 生成公私钥对 (pk, sk). $\mathcal{CH}$ 将 (pp, pk) 发送给 $\mathcal{A}$.
- 挑战: $\mathcal{A}$ 选择 $m_0, m_1 \in \mathbb{G}$ 并发送给 $\mathcal{CH}$. $\mathcal{CH}$ 选择随机比特 $\beta \in \{0, 1\}$, 随机选择 $r \in \mathbb{Z}_q$, 计算 $c^* = (g^r, pk^r \cdot m_\beta)$ 并发送给 $\mathcal{A}$.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. $\mathcal{A}$ 成功当且仅当 $\beta' = \beta$.

根据定义, 我们有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game_1: 与 Game_0 的唯一不同在于挑战密文的生成方式, $\mathcal{CH}$ 不再计算 pk^r 作为会话密钥掩蔽 m_β, 而是随机选取 $z \xleftarrow{R} \mathbb{Z}_q$, 用 g^z 作为会话密钥掩蔽 m_β, 得到挑战密文 $c^* = (g^r, g^z \cdot m_\beta)$. 在 Game_1 中, r 和 z 均为挑战者从 $\mathbb{Z}_q$ 中独立随机选取, 因此挑战密文 c^* 在 $\mathbb{G} \times \mathbb{G}$ 上均匀分布, 完美隐藏了 β 的信息. 因此, 即使对于拥有无穷计算能力的敌手, 我们也有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_1] - 1/2| = 0$$

引理 3.2

如果 DDH 假设成立, 那么对于任意 PPT 敌手我们均有 $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\kappa)$.

♡

证明 证明的思路是反证. 若存在 PPT 敌手 $\mathcal{A}$ 在 Game_0 和 Game_1 中成功的概率差不可忽略, 则可构造出 PPT 算法 $\mathcal{B}$ 打破 DDH 困难问题. 令 $\mathcal{B}$ 的 DDH 挑战实例为 (g, g^a, g^b, g^c), $\mathcal{B}$ 的目标是区分挑战实例是 DDH 四元组还是随机四元组. 为此, $\mathcal{B}$ 扮演 IND-CPA 游戏中的挑战者与 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{B}$ 根据它的挑战实例生成 pp, 令 $pk = g^a$, 将 (pp, pk) 发送给 $\mathcal{A}$. 注意, $\mathcal{B}$ 并不知晓 a (这是符合逻辑的, 不然归约无意义).

- 挑战: $\mathcal{A}$ 选择 $m_0, m_1 \in \mathbb{G}$ 并发送给 $\mathcal{B}$. $\mathcal{B}$ 随机选择 $\beta \xleftarrow{R} \{0,1\}$, 设置 $c^* = (g^b, g^c \cdot m_\beta)$ 并发送给 $\mathcal{A}$. 该设定隐式的设定 $r = b$.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. 如果 $\beta' = \beta$, $\mathcal{B}$ 输出 "1", 否则输出 "0".

对上述交互分析可知, 如果 $c = ab$, 那么 $\mathcal{B}$ 完美的模拟了 Game_0; 如果 c 在 $\mathbb{Z}_q$ 中随机选择, 那么 $\mathcal{B}$ 完美的模拟了 Game_1. 因此, $\mathcal{B}$ 解决 DDH 挑战的优势 $\mathsf{Adv}_{\mathcal{B}}(\kappa) = |\Pr[S_0] - \Pr[S_1]|$. 如果 DDH 假设成立, 我们有 $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\kappa)$.

综上, 定理得证. □

注记 3.9 (具有实际应用价值的同态)

ElGamal PKE 构建在 q 阶循环群 $\mathbb{G} = \langle g \rangle$ 上, 明文空间是 $\mathbb{G}$, 使用公钥 pk 对明文 m 的加密所得密文为 $(g^r, pk^r \cdot m)$. 容易验证, ElGamal PKE 相对于 $\mathbb{G}$ 中的群运算 "$\cdot$" 同态, 然而, 这种类型的同态并无实际意义, 现实应用中需要的是相对于 $\mathbb{Z}_q$ 上的模加运算 "+" 同态. 面向实际需求, ISO/IEC 标准化了 exponential ElGamal PKE 方案. 该方案同样构建在 q 阶循环群 $\mathbb{G} = \langle g \rangle$ 上, 所不同的是明文空间设定为 $\mathbb{G}$ 的自然同构 $\mathbb{Z}_q$, 使用公钥 pk 对明文 m 加密时, 首先计算 m 的自然同构映射结果 g^m, 再如常加密, 最终密文为 $(g^r, pk^r \cdot g^m)$. 容易验证, exponential ElGamal PKE 相对于 $\mathbb{Z}_q$ 中的 "+" 运算同态.

注记 3.10 (有实际意义的可重随机化性质)

目前几乎所有可重随机化的公钥加密方案都满足同态性, 这是因为同态性自然蕴含了可重随机化性质: 令待重随机化密文为 c, 随机生成对明文空间 M 中单位元的加密 c^*, 计算 $c + c^*$ 作为 c 的重随机化. 同态公钥加密方案均构建在群等代数结构上, 而真实的应用场景通常需要明文空间是 $\{0,1\}^n$, 且 n 为较大的正整数如 128, 这就隐式的要求存在 $\{0,1\}^n$ 到代数结构的高效编码方案. 然而, 若公钥加密方案的明文空间的二进制表示稀疏时 (如 ElGamal PKE 的明文空间为椭圆曲线群或 $\mathbb{Z}_q^*$ 的子群), 高效的编码方案是难以构造的, 这就形成实际应用中的痛点问题, 构成了密码学理论与应用之间的深坑裂隙: 理论上有完美的解决方案, 但找不到满足实际需求的高效实现.

注: 当代数结构的二进制表示稠密时或者明文空间较小时, 上述痛点问题有希望得到解决.

3.3.2 Twisted ElGamal PKE

近半个世纪, 随着网络技术的飞速发展, 计算模式逐渐由集中式迁移分布式. 新型计算模式对加密方案的需求也从单一的机密性保护扩展到对隐私计算的支持. 上一节注记中提到的 Exponential ElGamal PKE 支持 $\mathbb{Z}_q$ 上的模加运算 "+" 同态, 适用于密态计算场景. 在区块链和机器学习等涉及恶意敌手的计算场景中, 还常需要以隐私保护的方式证明密文加密的明文满足声称的约束关系, 特别的, 在指定的区间内, 我们称之为零知识密态区间范围证明.

零知识密态区间范围证明又可以根据证明者的角色分为两类:

1. 证明者为密文生成方: 证明者知晓加密随机数 r 和加密消息 m.
2. 证明者为密文接收方: 证明者知晓解密私钥 sk 和加密消息 m.

我们称上面两种情形下完成密态证明的组件为 Gadget-1 和 Gadget-2. 下面详细讨论 Gadget-1 的构造, Gadget-2 的设计可以通过重加密技术归结为 Gadget-1.

当前最高效的零知识区间范围证明系统是构建在离散对数群上的 Bulletproof [288], 其接受的断言类型为 Pedersen 承诺. 尽管 exponential ElGamal PKE 密文的第二项 $pk^r \cdot g^m$ 也是 Pedersen 承诺的形式, 但是若证明者为密文生成方, 则其知晓承诺密钥 (pk, g) 之间的离散对数关系, 因此无法调用 Bulletproof 完成证明 (合理性得不到保证), 如图 3.5所示.

解决该问题有两种技术手段:

1. 文献 [289] 中的方法: 证明者首先设计 NIZKPoK 协议证明其知晓密文的随机数和消息, 再引入新的 Pedersen 承诺作为桥接, 并设计 NIZK 协议证明新承诺的消息与明文的一致性 (注: NIZK 协议可与前面的 NIZKPoK

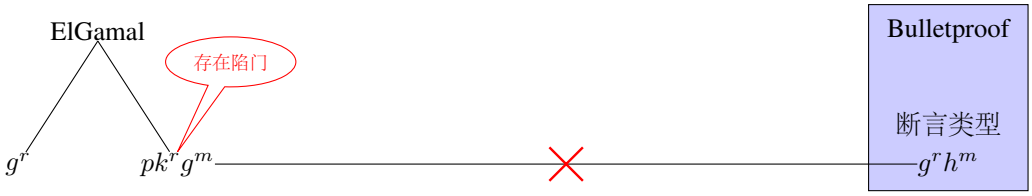


图 3.5: ElGamal 无法与 Bulletproof 直接对接

协议合并设计), 再调用 Bulletproof 对桥接承诺进行证明, 如图 3.6所示. 该方法的缺点是需要引入桥接承诺的额外的 Σ 协议, 增大了证明和验证的开销.

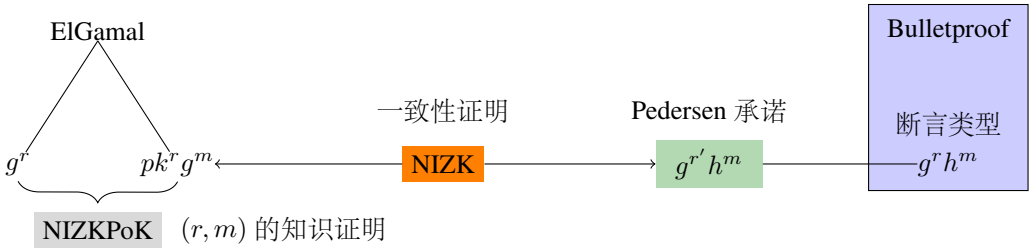


图 3.6: ElGamal PKE 的密态区间范围证明组件 Gadget-1 之设计方法一

2. 文献 [290] 中的方法: 结合待证明的 ElGamal PKE 密文对 Bulletproof 进行重新设计, 使用量身定制的 Σ-Bulletproof 完成证明, 如图 3.7所示. 该方法的缺点是需要对 Bulletproof 进行定制化的改动, 不具备模块化特性.

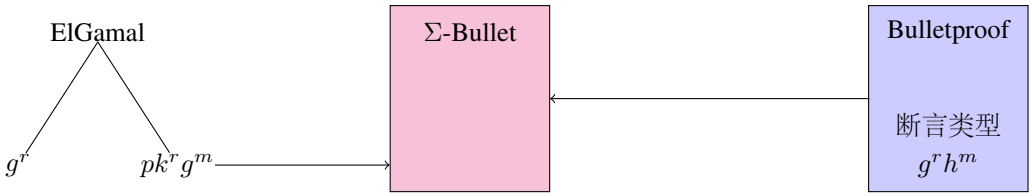


图 3.7: ElGamal PKE 的密态区间范围证明组件 Gadget-1 之设计方法二

上述两种技术手段均存在不足. 为了解决这一问题, 陈等 [291] 对 exponential ElGamal PKE 进行变形扭转, 将封装密文 g^r 与会话密钥 pk^r 的位置对调, 同时更改同构映射编码的基底, 得到 twisted ElGamal PKE.

构造 3.5 (Twisted ElGamal PKE)

- Setup(1^κ): 运行 GenGroup$(1^\kappa) \to (\mathbb{G}, q, g)$, 随机选取 $\mathbb{G}$ 的另一生成元 h, 输出 pp 包含循环群和 h 的描述, 同时包含对公钥空间 $PK = \mathbb{G}$、私钥空间 $SK = \mathbb{Z}_q$、明文空间 $M = \mathbb{Z}_q$ 和密文空间 $C = \mathbb{G}^2$.
- KeyGen(pp): 随机选取 $sk \in \mathbb{Z}_q$ 作为私钥, 计算公钥 $pk := g^{sk}$.
- Encrypt(pk, m): 以公钥 pk 和明文 $m \in \mathbb{Z}_q$ 为输入, 随机选择 $r \xleftarrow{R} \mathbb{Z}_q$, 计算 $c_0 = pk^r$, $c_1 = g^r \cdot h^m$, 输出密文 $c = (c_1, c_2) \in C$.
- Decrypt(sk, c): 以私钥 sk 和密文 $c = (c_0, c_1)$ 为输入, 输出 $m' := \log_h c_1 / c_0^{sk^{-1}}$.

♣

正确性. 以下公式 3.5说明方案具有完美正确性:

$$c_1 / c_0^{sk^{-1}} = g^r \cdot h^m / (pk^r)^{sk^{-1}} = h^m \tag{3.5}$$

定理 3.3

如果 DDH 假设成立, 那么 twisted ElGamal PKE 是 IND-CPA 安全的.

♡

证明与标准的 ElGamal PKE 证明类似, 我们留给读者作为练习.

笔记 为获得 $\mathbb{Z}_q$ 上的加法同态, exponential ElGamal PKE 和 twisted ElGamal PKE 均将明文空间设定为 $\mathbb{Z}_q$, 加密时必须先进行同构编码, 解密时则在最后需要进行解码. 解码的过程等同于求解离散对数, 因此为了确保解密高效, 必须将有效的明文空间限制在较小的范围内, 如 $[0, 2^{40}]$.

零知识证明友好特性. 新的加密方案与 exponential ElGamal PKE 的性能和安全性相当, 同样满足 $\mathbb{Z}_q$ 上的模加同态. 特别的, 密文的第二部分恰好是标准的 Pedersen 承诺形态 (承诺密钥陷门未知), 可无缝对接 Bulletproof 等一系列断言类型为 Pedersen commitment 的区间范围证明, 如图 3.8所示. 我们称公钥加密方案的这种性质为零知识证明友好.

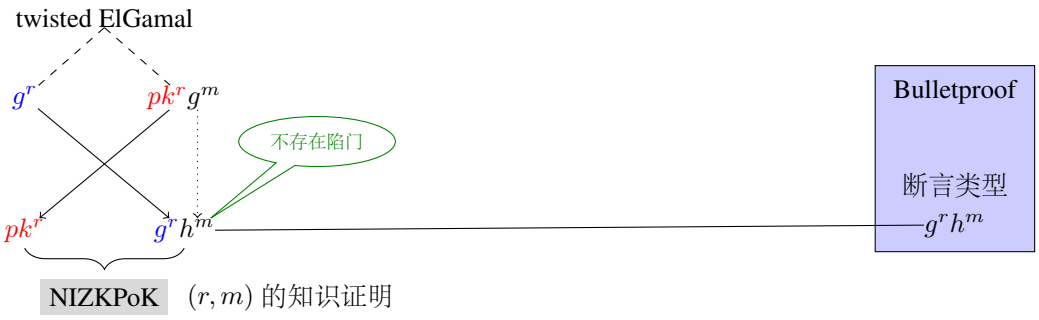


图 3.8: Twisted ElGamal PKE 的密态区间范围证明组件 Gadget-1

Twisted ElGamal PKE 的密态证明组件 Gadget-2 的设计可以通过如下步骤完成:

1. 证明者使用 sk 对密文 $(pk^r, g^r h^m)$ 进行部分解密得到 h^m;
2. 证明者选取新的随机数对 m 进行重加密得到新密文 $(pk^{r^*}, g^{r^*} h^m)$;
3. 证明者设计 NIZK 协议证明新旧密文的一致性, 即均是对同一个消息的加密 (具体可通过证明 DDH 元组的 Sigma 协议实现);
4. 证明者调用 Gadget-1 对新密文完成密态证明.

相比标准的 ElGamal PKE, twisted ElGamal PKE 的显著优势就在于零知识证明友好, 下表对比了两者的密态证明组件的效率.

表 3.1: 标准 ElGamal 和 twisted ElGamal 与 Bulletproof 对接开销的对比

对接开销	PKE	证明大小	证明者开销	验证者开销
Gadgets-1/2	标准 ElGamal	$n(2\lvert\mathbb{G}\rvert + \lvert\mathbb{Z}_q\rvert)$	n(4Exp+2Add)	n(3Exp+2Add)
	twisted ElGamal	0	0	0

在统计证明者和验证者开销时, 略去了数域上的操作, 因其开销相比椭圆曲线群上的操作相比可忽略. n 是需要证明的密文数. 在很多实际应用中, 单个证明者需要对多个密文通过 Gadget-1/2 进行区间范围证明. 当密文数量为百万量级时, 使用 twisted ElGamal 带来的性能提升是相当可观的.

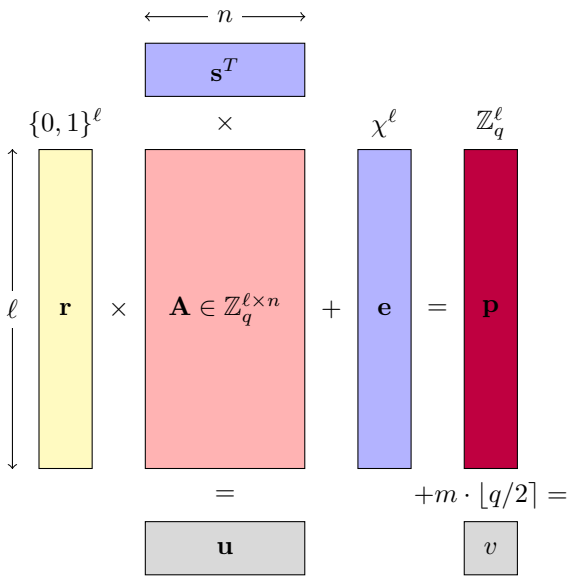


图 3.9: Regev PKE 加密方案示意图

3.4 基于格问题的经典方案

3.4.1 Regev PKE

Regev [265] 中提出了 LWE 困难问题, 并基于该问题构造了一个公钥加密方案, 称为 Regev PKE.

构造 3.6 (Regev PKE)

- Setup(1^κ): 以安全参数 1^κ 为输入, 生成随机矩阵 $\mathbf{A} \in \mathbb{Z}_q^{\ell \times n}$ 作为公开参数.
- KeyGen(pp): 以公开参数 pp 为输入, 随机选取向量 $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$ 作为私钥, 随机选取噪声向量 $\mathbf{e} \xleftarrow{R} \chi^\ell$ (其中 $\chi^\ell = D_{\mathbb{Z}^\ell, r}$), 计算 $\mathbf{p} \leftarrow \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^\ell$ 作为公钥.
- Encrypt(pk, m): 以公钥 $pk = \mathbf{p}$ 和明文 $m \in \{0,1\}$ 为输入, 随机选取向量 $\mathbf{r} \xleftarrow{R} \{0,1\}^\ell$ 计算 $\mathbf{u}^T = \mathbf{r}^T\mathbf{A}$ 和 $v = \mathbf{r}^T\mathbf{p} + m \cdot \lfloor q/2 \rceil$, 输出密文 $(\mathbf{u}, v)$.
- Decrypt($\mathbf{s}, c$): 以私钥 $sk = \mathbf{s}$ 和密文 $c = (\mathbf{u}, v)$, 计算 $y = v - \mathbf{u}^T\mathbf{s} \in \mathbb{Z}_q$, 若 y 接近 0 则输出 0, 若 y 接近 $\lfloor q/2 \rceil$ 则输出 1.

正确性. 观察以下等式:

$$\begin{aligned} y &= v - \mathbf{u}^T\mathbf{s} \\ &= \mathbf{r}^T\mathbf{p} + m \cdot \lfloor q/2 \rceil - \mathbf{r}^T\mathbf{A}\mathbf{s} \\ &= \mathbf{r}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) + m \cdot \lfloor q/2 \rceil - \mathbf{r}^T\mathbf{A}\mathbf{s} \\ &= \mathbf{r}^T\mathbf{e} + m \cdot \lfloor q/2 \rceil \end{aligned}$$

由上述推导可知, 当累计误差 $|\langle \mathbf{r}, \mathbf{e} \rangle| \leq q/4$ 时解密正确. 因此, 在参数选取时应令 q 的取值相对于误差分布 χ 和 ℓ 相对较大. 比如, 当 $\chi = D_{\mathbb{Z},r}$ 是离散 Gaussian 分布时, $\langle \mathbf{r}, \mathbf{e} \rangle$ 是参数至多为 $r\sqrt{\ell}$ 的亚 Gaussian 分布, 其尺寸小于 $r\sqrt{\ell \ln(1/\varepsilon)/\pi}$ 的概率至少为 $1 - 2\varepsilon$. 为了确保解密错误的概率可忽略, 可设定 $r = \Theta(\sqrt{n})$, $q = \tilde{O}(n)$, 对应 LWE 错误率 $\alpha = r/q = 1/\tilde{O}(n)$.

定理 3.4

如果判定性 LWE 假设成立, 则 Regev PKE 是 IND-CPA 安全的.

证明 令 S_i 表示敌手在 Game_i 中成功概率. 以游戏序列的方式组织证明如下:

Game_0: 该游戏是标准的 IND-CPA 游戏. 挑战者 $\mathcal{CH}$ 和敌手 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 运行 $\mathsf{Setup}(1^\kappa)$ 生成公开参数 $\mathbf{A} \in \mathbb{Z}_q^{\ell \times n}$, 同时生成公私钥对, 其中私钥 sk 为随机向量 $\mathbf{s} \in \mathbb{Z}_q^n$, 公钥 pk 为 $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^\ell$, 其中 $\mathbf{e} \leftarrow \chi^\ell$.
- 挑战: $\mathcal{A}$ 选取 (m_0, m_1) 发送给 $\mathcal{CH}$. $\mathcal{CH}$ 随机选取 $\mathbf{r} \xleftarrow{R} \{0,1\}^\ell$, $\beta \xleftarrow{R} \{0,1\}$, 计算 $\mathbf{u} = \mathbf{r}^T\mathbf{A}$, $v = \mathbf{r}^T\mathbf{p} + m_\beta \cdot \lfloor q/2 \rceil$, 发送 $(\mathbf{u}, v)$ 给 $\mathcal{A}$ 作为挑战密文.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. $\mathcal{A}$ 成功当且仅当 $\beta = \beta'$.

根据定义, 我们有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game_1: 与 Game_0 惟一不同的是 $\mathcal{CH}$ 生成公钥的方式由计算 $\mathbf{A}\mathbf{s} + \mathbf{e}$ 变为随机选取 $\mathbb{Z}_q^\ell$ 上的向量. 在 Game_1 中, $\vec{\mathbf{A}} = \mathbf{A} \mid \mathbf{p}$ 是 $\mathbb{Z}_q^{\ell \times n}$ 上的随机矩阵, 容易验证 $f_{\vec{\mathbf{A}}}(\mathbf{r}) = \mathbf{r}^T\vec{\mathbf{A}}$ 是从 $\{0,1\}^\ell$ 到 $\mathbb{Z}_q^{n+1}$ 的 universal hash, 由参数选取 $\ell > n \log q$ 和剩余哈希引理 (leftover hash lemma) 可知, 函数的输出统计不可区分于 $\mathbb{Z}_q^{n+1}$ 上的均匀分布. 因此, 挑战密文几乎完美掩盖了 β 的信息. 因此, 即使对于拥有无穷计算能力的敌手, 我们也有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_1] - 1/2| = \mathsf{negl}(\kappa)$$

断言 3.1

如果判定性 LWE 假设成立, 那么对于任意 PPT 敌手均有 $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\kappa)$.

证明 证明的思路是反证. 若存在 PPT 敌手 $\mathcal{A}$ 在 Game_0 和 Game_1 中成功的概率差不可忽略, 则可构造出 PPT 算法 $\mathcal{B}$ 打破 LWE 困难问题. 令 $\mathcal{B}$ 的 LWE 挑战实例为 $(\mathbf{A}, \mathbf{p})$, $\mathcal{B}$ 的目标是区分挑战实例是随机采样还是 LWE 采样. 为此 $\mathcal{B}$ 扮演 IND-CPA 游戏中的挑战者与 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{B}$ 发送 $(\mathbf{A}, \mathbf{p})$ 给 $\mathcal{A}$. 该操作将 pk 隐式地设定为 $\mathbf{p}$.
- 挑战: $\mathcal{A}$ 选取 (m_0, m_1) 发送给 $\mathcal{CH}$. $\mathcal{CH}$ 随机选取 $\mathbf{r} \xleftarrow{R} \{0,1\}^\ell$, $\beta \xleftarrow{R} \{0,1\}$, 计算 $\mathbf{u} = \mathbf{r}^T\mathbf{A}$, $v = \mathbf{r}^T\mathbf{p} + m_\beta \cdot \lfloor q/2 \rceil$, 发送 $(\mathbf{u}, v)$ 给 $\mathcal{A}$ 作为挑战密文.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. 如果 $\beta = \beta'$, $\mathcal{B}$ 输出 1.

对上述交互分析可知, 如果 $\mathbf{p}$ 是 LWE 采样, 那么 $\mathcal{B}$ 完美模拟了 Game_0; 如果 $\mathbf{p}$ 是随机采样, 那么 $\mathcal{B}$ 完美模拟了 Game_1. 因此, $\mathcal{B}$ 解决 LWE 挑战的优势 $\mathsf{Adv}_{\mathcal{B}}(\kappa) = |\Pr[S_0] - \Pr[S_1]|$. 如果 LWE 假设成立, 我们有 $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\kappa)$. □

综上, 定理得证. □

注记 3.11

Regev PKE 和 Goldwasser-Micali PKE 在设计上有异曲同工之处, 均采用的是有损加密思想, 即公钥存在正常和有损这两种计算不可区分的类型, 正常公钥生成的密文可以正确解密, 而有损公钥生成的密文丢失了明文的全部信息. 在安全性证明时, 利用两种类型公钥的计算不可区分性以及有损加密的性质, 即可完成 IND-CPA 安全的论证.

3.4.2 GPV PKE

Gentry, Peikert 和 Vaikuntanathan [127] 基于 LWE 假设构造出另一个 PKE 方案, 称为 GPV PKE.

构造 3.7 (GPV PKE)

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 生成随机矩阵 $\mathbf{A} \in \mathbb{Z}_q^{\ell \times n}$ 作为公开参数.
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入, 随机选取噪声向量 $\mathbf{r} \xleftarrow{R} \{0,1\}^\ell$ 作为私钥, 计算 $\mathbf{u}^T \leftarrow \mathbf{r}^T\mathbf{A}$ 作为公钥. 从编码的角度, $\mathbf{u}$ 可以理解为 $\mathbf{r}$ 相对于 $\mathbf{A}$ 的 syndrome.
- $\mathsf{Encrypt}(pk, m)$: 以公钥 $pk = \mathbf{u}$ 和明文 $m \in \{0,1\}$ 为输入, 随机选取向量 $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$ 和 $\mathbf{e} \xleftarrow{R} \chi^\ell$, 随机选

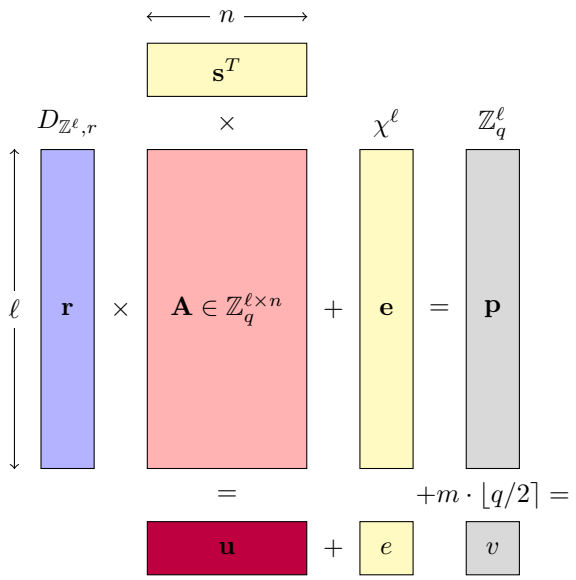


图 3.10: GPV PKE 加密方案示意图

取 $e \xleftarrow{R} \chi$, 计算 $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^\ell$ 和 $v = \mathbf{u}^T\mathbf{s} + e + m \cdot \lfloor q/2 \rceil$, 输出密文 $(\mathbf{p}, v)$.

- Decrypt$(\mathbf{r}, c)$: 以私钥 $sk = \mathbf{r}$ 和密文 $c = (\mathbf{p}, v)$, 计算 $y = v - \mathbf{r}^T\mathbf{p} \in \mathbb{Z}_q$, 若 y 接近 0 则输出 0, 若 y 接近 $\lfloor q/2 \rceil$ 则输出 1.

♣

正确性. 观察以下等式:

$$\begin{aligned} y &= v - \mathbf{r}^T\mathbf{p} \\ &= \mathbf{u}^T\mathbf{s} + e + m \cdot \lfloor q/2 \rceil - \mathbf{r}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) \\ &= \mathbf{u}^T\mathbf{s} + e + m \cdot \lfloor q/2 \rceil - \mathbf{u}^T\mathbf{s} - \mathbf{r}^T\mathbf{e} \\ &= m \cdot \lfloor q/2 \rceil + e - \mathbf{r}^T\mathbf{e} \end{aligned}$$

由上述推导可知, 当累计误差 $|\langle e - \mathbf{r}^T\mathbf{e} \rangle| \leq q/4$ 时解密正确. 通过恰当的参数选择, 可确保累计误差以接近 1 的绝对优势概率小于等于 $q/4$, 更多细节请参考 [127].

定理 3.5

如果判定性 LWE 假设成立, 则 GPV PKE 是 IND-CPA 安全的.

♡

证明 令 S_i 表示敌手在 Game$_i$ 中成功概率. 以游戏序列的方式组织证明如下:

Game$_0$: 该游戏是标准的 IND-CPA 游戏. 挑战者 $\mathcal{CH}$ 和敌手 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 运行 Setup(1^κ) 生成公开参数 $\mathbf{A} \in \mathbb{Z}_q^{\ell \times n}$, 同时生成公私钥对, 其中私钥 sk 为随机向量 $\mathbf{r} \in D_{\mathbb{Z}^\ell,r}$, 公钥 pk 为 $\mathbf{u} = \mathbf{r}^T\mathbf{A}$.
- 挑战: $\mathcal{A}$ 选取 (m_0, m_1) 发送给 $\mathcal{CH}$. $\mathcal{CH}$ 随机选取 $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$, 随机选取 $\mathbf{e} \xleftarrow{R} \chi^\ell$ 和 $e \xleftarrow{R} \chi$, $\beta \xleftarrow{R} \{0,1\}$, 计算 $\mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^\ell$, $v = \mathbf{u}^T\mathbf{s} + e + m_\beta \cdot \lfloor q/2 \rceil$ 作为密文. 发送 $(\mathbf{u}, v)$ 给 $\mathcal{A}$ 作为挑战密文.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. $\mathcal{A}$ 成功当且仅当 $\beta = \beta'$.

根据定义, 我们有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game$_1$: 与 Game$_0$ 惟一不同的是 $\mathcal{CH}$ 生成公钥的方式由计算 $\mathbf{u}^T = \mathbf{r}^T\mathbf{A}$ 变为随机选取 $\mathbf{u} \xleftarrow{R} \mathbb{Z}_q^n$ 上的向量. 在 Game$_1$ 中, $(\mathbf{A}, \mathbf{p} = \mathbf{A}\mathbf{s} + \mathbf{e}, \mathbf{u}, \mathbf{u}^T\mathbf{s} + e)$ 恰好构成 $\ell + 1$ 个 LWE 采样结果. 有 LWE 假设立刻可知, 敌手在 Game$_1$

中的视角计算意义下隐藏了 β 的信息, 因此基于 LWE 假设有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_1] - 1/2| \leq \mathsf{negl}(\kappa)$$

断言 3.2

对于任意的敌手 $\mathcal{A}$(即使拥有无穷计算能力), 均有 $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\kappa)$ ♡

证明 根据 $\ell \geq 2n \log q$ 的参数选择可知, 公钥 $\mathbf{u}$ 的分布与 $\mathbb{Z}_q^n$ 上的均匀分布统计不可区分, 因此敌手在 Game_0 和 Game_1 中的视图统计不可区分, 从而 $|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{negl}(\kappa)$. □

综上, 定理得证. □

注记 3.12

Regev PKE 和 GPV PKE 在形式上相似, 构造使用了相同的元素 $\mathbf{A}, \mathbf{s}, \mathbf{r}, \mathbf{e}, \mathbf{p}, \mathbf{u}, v$, 但用途含义不完全相同, 构造互为对偶. Regev PKE 中, $\mathbf{p}$ 为公钥, $(\mathbf{s}, \mathbf{e})$ 为私钥, $\mathbf{u}$ 为密文; GPV PKE 中 $\mathbf{p}$ 为密文, $(\mathbf{s}, \mathbf{e})$ 为加密随机数, $\mathbf{u}$ 为公钥. 感兴趣的读者可以参阅 Micciancio [292] 了解更多格密码学中的对偶性. Regev PKE 中, 公钥空间是稀疏的; 而在 GPV PKE 中, 公钥空间是稠密的, 这一特性使得我们可以借助随机谕言机将 GPV PKE 编译为身份加密方案—GPV IBE. ♠

第四章 公钥加密的通用构造方法

本章开始介绍公钥加密的通用构造方法. 4.1节介绍了基于各类单向陷门函数的构造，4.2节介绍了基于哈希证明系统的构造，4.3节介绍了基于可提取哈希证明系统的构造，4.4节介绍了基于程序混淆的构造，4.5节介绍了基于可公开求值伪随机函数的构造, 统一阐释上述通用构造.

4.1 单向陷门函数类

4.1.1 基于单向陷门函数的构造

单向陷门函数 (TDF) 是单向函数 (OWF) 在 Cryptomania 中的对应, 简言之, 其正向计算容易, 逆向计算困难但在有陷门信息辅助时容易.

定义 4.1 (单向陷门函数 (TDF))

TDF 由以下 4 个 PPT 算法组成:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 pp, 其中 pp 包含对定义域 D, 值域 R, 求值公钥空间 EK、求逆陷门空间 TD 和单向陷门函数族 $f : EK \times D \to R$ 的描述. 换言之, f 是由求值公钥索引的函数族. 不失一般性, D 支持高效的随机采样, 即存在 PPT 算法 SampDom 可以从 D 中随机选取一个元素. 在多数情况下, D 和 R 是与求值公钥无关的 (该性质也被称为 index-independent), 但在有些情形下, D 和 R 是由求值公钥索引的空间簇. 为了叙述简洁, 以下均假设 D 和 R 是单一空间. 空间簇的情形由单一集合的情形自然推广得到.
- $\mathsf{KeyGen}(pp)$: 以公共参数 pp 为输入, 输出密钥对 (ek, td), 其中 ek 为求值公钥, td 为求逆陷门.
- $\mathsf{Eval}(ek, x)$: 以求值公钥 ek 和定义域元素 $x \in D$ 为输入, 输出 $y \leftarrow f_{ek}(x)$.
- $\mathsf{TdInv}(td, y)$: 以求逆陷门 td 和值域元素 $y \in R$ 为输入, 输出 $x \in D$ 或特殊符号 $\perp$ 指示 y 不存在原像.

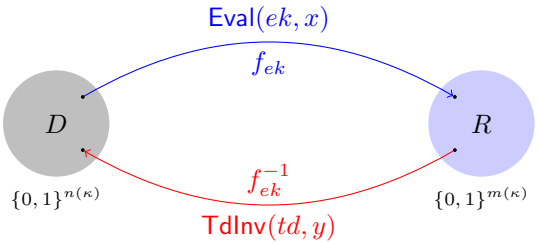
♣

定义以下两条性质:

- 单射: $\forall ek$, 称 f_{ek} 是单射的当且仅当 $x \neq x' \Rightarrow f_{ek}(x) \neq f_{ek}(x')$.
- 置换: $\forall ek, \mathrm{Img}(f_{ek}) = D = R$.

$$\mathsf{Setup}(1^\kappa) \to pp$$

$$\mathsf{KeyGen}(pp) \to (ek, td)$$



正确性. 对于 $\forall \kappa \in \mathbb{N}, pp \leftarrow \mathsf{Setup}(1^\kappa), (ek, td) \leftarrow \mathsf{KeyGen}(pp)$ 和 $x \in D$ 以及 $y = \mathsf{Eval}(ek, x)$, 有:

$$\Pr[\mathsf{TdInv}(td, y) \in f_{ek}^{-1}(y)] = 1$$

单向性. 定义单向陷门函数敌手 $\mathcal{A}$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[x \in f_{ek}^{-1}(y^*) : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (ek, td) \leftarrow \mathsf{KeyGen}(pp); \\ x^* \xleftarrow{R} D, y^* \leftarrow \mathsf{Eval}(ek, x^*); \\ x \leftarrow \mathcal{A}(pp, ek, y^*) \end{array}\right].$$

如果对于任意的 PPT 敌手 $\mathcal{A}$, 其优势函数均是可忽略的, 则称该陷门函数是单向的.

注记 4.1

1. 不失一般性, 假定 D 和 R 均存在经典表示, 分别是 $\{0,1\}^{n(\kappa)}$ 和 $\{0,1\}^{m(\kappa)}$, 其中 $n(\cdot)$ 和 $m(\cdot)$ 是关于 κ 的多项式函数. 容易验证, 长度函数不能过大, 如果 $n(\cdot)$ 或 $m(\cdot)$ 是超多项式函数, 则函数无法高效

计算; 长度函数也不能过小, 如果 $n(\cdot)$ 或 $m(\cdot)$ 是亚线性函数, 则函数不可能满足单向性.

2. 在抽象定义中, 只限定了 $\mathsf{TDFInv}(td,\cdot)$ 在输入为像集元素时返回原像, 而未限定其输入为非像集元素时的行为. 在具体构造时, $\mathsf{TDFInv}(td,\cdot)$ 在输入为非像集元素时的行为往往需要精心设定, 以方便安全性证明.
3. 在单向性的定义中, 敌手 $\mathcal{A}$ 仅观察到 ek 和 y^* 的信息. $x^* \xleftarrow{R} D$ 可以放宽至 x^* 选自 D 上具有高最小熵的分布, 即 $\mathsf{H}_\infty(x^*) \geq \omega(\log\kappa)$.

在介绍基于单向陷门函数的 PKE 构造前, 先展示一个基于单向陷门置换的朴素构造. 该构造并不安全, 但对得到正确的构造很有启发意义.

构造 4.1 (基于 TDP 的朴素 PKE 构造 (不安全))

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{TDP.Setup}(1^\kappa)$ 生成公开参数 pp, 其中明文空间和密文空间均为单向陷门置换的定义域 D.
- $\mathsf{KeyGen}(pp)$: 运行 $\mathsf{TDP.KeyGen}(pp) \to (ek, td)$, 其中 ek 作为加密公钥, td 作为解密私钥.
- $\mathsf{Encrypt}(ek, m)$: 以公钥 ek 和明文 $m \in D$ 为输入, 运行 $\mathsf{TDP.Eval}(ek, m)$ 计算 $c \leftarrow f_{ek}(m)$ 作为密文.
- $\mathsf{Decrypt}(td, c)$: 以私钥 td 和密文 $c \in D$ 为输入, 运行 $\mathsf{TDP.TdInv}(td, c)$ 计算 $m \leftarrow f_{ek}^{-1}(c)$ 恢复明文.

上述构造来自 Diffie 和 Hellman 的经典论文 [3], 原始的 RSA 公钥加密方案就是该构造的具体实例化. 该构造的想法直观, 利用单向陷门置换将明文转化为密文, 同时利用陷门可以求逆从密文中恢复明文. 但其仅仅满足较弱的 OW-CPA 安全, 并不满足现在公认的最低要求 IND-CPA 安全, 因此其也被称为公钥加密的 textbook 构造[1]. 朴素构造不满足 IND-CPA 安全的根本原因是加密算法是确定型的而非概率型的, 因此敌手可以通过 "加密-比较" 即可打破 IND-CPA 安全. 因此, 强化朴素构造的第一步是选择定义域中的随机元素 x, 计算其函数值 $f_{ek}(x)$ 作为封装密文, 再用 x 作为会话密钥掩蔽明文. 强化构造仍然不满足 IND-CPA 安全性, 原因是 $f_{ek}(\cdot)$ 是公开可计算函数, 其函数值泄漏了原像信息, 使得原像在敌手的视角中不再伪随机. 针对性的强化方法是计算 x 的硬核函数 (hardcore function) 值作为会话密钥.

以下首先介绍硬核函数的概念.

定义 4.2 (硬核函数)

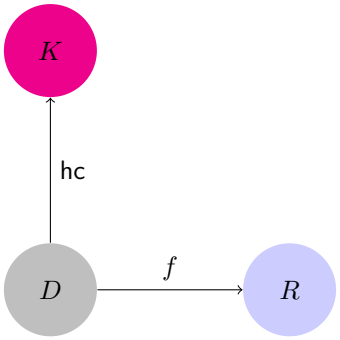
称多项式时间可计算的确性型函数 $\mathsf{hc}: D \to K$ 是函数 $f: D \to R$ 的硬核函数当且仅当:

$$(f(x^*), \mathsf{hc}(x^*)) \approx_c (f(x^*), U_K)$$

其中概率空间建立在 $x^* \xleftarrow{R} D$ 的随机带上. 以安全实验的方式可如下定义, 即对于任意 PPT 敌手 $\mathcal{A}$, 其安全优势可忽略:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left| \Pr\left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ x^* \xleftarrow{R} D, y^* \leftarrow f(x^*); \\ k_0^* \leftarrow \mathsf{hc}(x^*), k_1^* \xleftarrow{R} K, \beta \xleftarrow{R} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}(pp, ek, y^*, k_\beta^*); \end{array} \right] - \frac{1}{2} \right|.$$

[1] textbook 指其仅适合作为以科普为目的教学.



定理 4.1 (Goldreich-Levin 定理)

如果 $f:\{0,1\}^n \to \{0,1\}^m$ 是单向函数, 那么 $\mathsf{GL}(x) = \bigoplus_{i=1}^n x_i r_i$ 是 $\{0,1\}^n \to \{0,1\}$ 的单比特输出硬核函数 (硬核谓词).

♡

注记 4.2

Goldreich-Levin 定理是现代密码学中极为重要的结论, 它的意义在于通过显式构造硬核函数, 建立起单向性与伪随机性之间的关联. 从另一个角度理解, GL 硬核谓词可以看做一个计算意义下的随机性提取器, 对 $x|f(x)$ 的计算熵进行随机性提取, 萃取出伪随机的一比特. 还需要特别说明的是, 到目前为止尚不知晓如何针对任意单向函数 f 设计一个确定型的硬核谓词. GL 是相对于 $g(x,r) := f(x)||r$ 的硬核谓词, 或者可以将 $r \xleftarrow{R} \{0,1\}^n$ 理解为硬核谓词的描述, 将 GL 理解为 f 的随机性硬核谓词. 本书中采用第二种观点.

另一方面, GL 硬核谓词是通用的 (universal), 即构造对于任意单向函数均成立. 强通用性的代价是效率较低, 输出仅是单比特. 当单向函数具有特殊的结构 (如函数是置换) 或者依赖额外困难假设 (如判定性假设、差异输入程序混淆假设) 时, 存在更高效的构造.

♠

以下我们展示如何基于单射的单向陷门函数构造构造 KEM 方案.

构造 4.2 (基于单射 TDF 的 CPA 安全的 KEM 构造)

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{TDF.Setup}(1^\kappa)$ 生成公开参数 pp. pp 中不仅包含单向陷门函数 $f_{ek}: D \to R$ 的描述, 还包括相应硬核函数 $\mathsf{hc}: D \to K$ 的描述. KEM 方案的密文空间是 TDF 的定义域 D, 密钥空间是硬核函数的值域 K.
- $\mathsf{KeyGen}(pp)$: 运行 $\mathsf{TDF.KeyGen}(pp) \to (ek, td)$, 其中 ek 作为封装公钥 pk, td 作为解封装私钥 sk.
- $\mathsf{Encaps}(pk)$: 以公钥 $pk = ek$ 为输入, 随机选取 $x \xleftarrow{R} D$, 运行 $\mathsf{TDF.Eval}(ek, x)$ 计算 $c \leftarrow f_{ek}(x)$ 作为封装密文, 计算 $k \leftarrow \mathsf{hc}(x)$ 作为会话密钥.
- $\mathsf{Decaps}(sk, c)$: 以私钥 $sk = td$ 和密文 c 为输入, 运行 $\mathsf{TDF.TdInv}(td, c)$ 计算 $x \leftarrow f_{ek}^{-1}(c)$, 输出 $k \leftarrow \mathsf{hc}(x)$.

♣

正确性. 由单向陷门函数的单射性质和求逆算法的正确性可知, 上述 KEM 构造满足正确性.

定理 4.2

如果 f_{ek} 是一族单射单向陷门函数, 那么上述 KEM 构造是 IND-CPA 安全的.

♡

证明 证明可通过单一归约完成, 证明若存在敌手 $\mathcal{A}$ 打破 KEM 方案的 IND-CPA 安全性, 则存在敌手 $\mathcal{B}$ 打破 hc 的伪随机性, 进而与 f_{ek} 的单向性矛盾. 令 $\mathcal{B}$ 的挑战实例为 (pp, ek, y^*, k_β^*), 其中 pp 为单射单向陷门函数的公开参数, ek 为随机生成的求值密钥, $y^* \leftarrow f_{ek}(x^*)$ 是随机选取原像 x^* 的像, $k_0^* \leftarrow \mathsf{hc}(x^*)$, $k_1^* \xleftarrow{R} K$ $\mathcal{B}$ 的目标是判定 $\beta = 0$ 抑或 $\beta = 1$. $\mathcal{B}$ 与 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{B}$ 根据 pp 生成 KEM 方案的公开参数, 并设定公钥 $pk := ek$, 将 (pp, ek) 发送给 $\mathcal{A}$.
- 挑战: $\mathcal{B}$ 设定 $c^* := y^*$, 将 (c^*, k_β^*) 发送给 $\mathcal{A}$.

- 猜测: $\mathcal{A}$ 输出 β', $\mathcal{B}$ 将 β' 转发给它自身的挑战者.

容易验证, $\mathcal{B}$ 完美地模拟了 KEM 方案中的挑战者, $\mathcal{B}$ 成功当且仅当 $\mathcal{A}$ 成功. 因此我们有:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{KEM}}(\kappa) = \mathsf{Adv}_{\mathcal{B}}^{\mathrm{hc}}(\kappa)$$

由 f_{ek} 的单向性可知, hc 伪随机, 从 KEM 构造满足 IND-CPA 安全性. □

以上的结果展示了单射单向陷门函数蕴含 IND-CPA 的公钥加密. 一个自然的问题是, 单向陷门函数需要满足何种性质才能蕴含 IND-CCA 的公钥加密. 以下, 我们按照时间先后顺序依次介绍单向陷门函数的三个增强版本, 并展示如何基于这些增强版本的单向陷门函数构造 IND-CCA 的公钥加密.

4.1.2 基于有损陷门函数的构造

天之道, 损有余而补不足, 是故虚胜实, 不足胜有余.

— 宋 · 黄裳《九阴真经》

理想世界中的镜中月和水中花体现的是信息完美复刻, 而现实世界中更多的现象体现的却是信息有损, 如拍照、录音, 无论设备和手段多么先进, 都无法做到完美复刻信源信息, 只能做到尽可能的高保真. 单射函数可以形象的理解为理想世界中信息无损的编码过程, 那么什么形式的函数刻画了现实世界中信息有损的编码过程呢? Peikert 和 Waters [31] 正是基于上述的思考, 在 2008 年开创性提出了有损陷门函数 (lossy trapdoor function, LTDF) 的概念. 简言之, 有损陷门函数有两种模式, 即单射和有损模式. 在单射模式下, 函数是单射的, 像完全保留了原像的全部信息; 在有损模式下, 函数是有损的, 像在信息论意义下丢失了原像的部分信息. 两种模式之间的关联是计算不可区分.

定义 4.3 (有损陷门函数 (LTDF))

有损陷门函数 LTDF 由 n 和 τ 两个参数刻画, 包含以下 5 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 pp, 其中 pp 包含对定义域 X 和值域 Y 的描述. 其中 $|X| = 2^{n(\kappa)}$.
- $\mathsf{GenInjective}(pp)$: 以公共参数 pp 为输入, 输出密钥对 (ek, td), 其中 ek 为求值公钥, td 为求逆陷门. 该算法输出的 ek 定义了从 X 到 Y 的单射函数 f_{ek}, 拥有对应 td 可以对 f_{ek} 进行高效求逆.
- $\mathsf{GenLossy}(pp)$: 以公共参数 pp 为输入, 输出密钥对 $(ek, \perp)$, 其中 ek 为求值公钥, $\perp$ 表示陷门不存在无法求逆. 该算法输出的 ek 定义了从 D 到 R 的有损函数 f_{ek}, 像集的大小至多为 $2^{\tau(\kappa)}$.
- $\mathsf{Eval}(ek, x)$: 以求值公钥 ek 和定义域元素 $x \in X$ 为输入, 输出 $y \leftarrow f_{ek}(x)$.
- $\mathsf{TdInv}(td, y)$: 以求逆陷门 td 和值域元素 $y \in Y$ 为输入, 输出 $x \in X$ 或特殊符号 $\perp$ 指示 y 不存在原像.

♣

有损陷门函数需满足以下性质:

模式不可区分性. $\mathsf{GenInjective}(pp)$ 和 $\mathsf{GenLossy}(pp)$ 的第一个输出构成的分布在计算意义下不可区分, 即任意 PPT 敌手无法判定求值公钥 ek 属于单射模式还是有损模式.

相比常规的单向陷门函数, 有损陷门函数额外具备一个计算不可区分的有损模式, 这正是其威力的来源. 在利用有损陷门函数设计密码方案/协议时, 通常按照如下的步骤:

1. 在单射模式下完成密码方案/协议的功能性构造 (功能性通常需要函数单射可逆);
2. 在有损模式下完成密码方案/协议的安全性论证 (论证通常在信息论意义下进行);
3. 利用单射模式和有损模式的计算不可区分性证明密码方案/协议在正常模式下计算安全性.

细心的读者可能已经发现了有损陷门函数的定义中并没有显式的要求函数在单射模式下具备单向性, 这是因为单射和有损模式的计算不可区分性已经隐式的保证了这一点. 以下进行严格证明, 具体展示应用有损陷门函数设计密码方案/协议的过程.

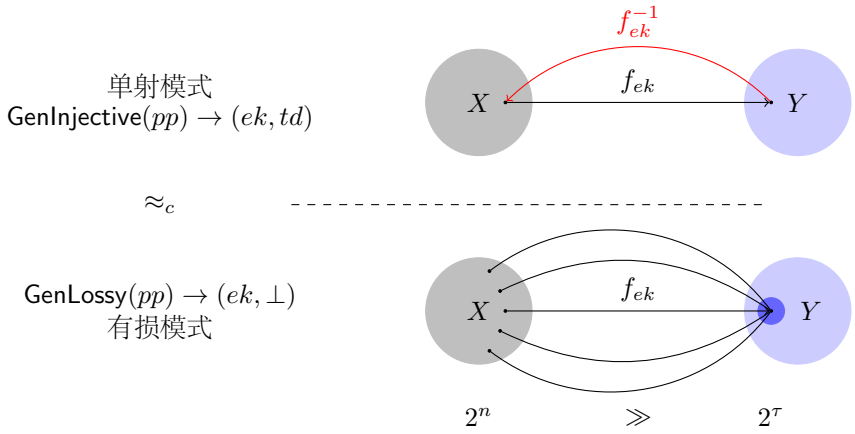


图 4.1: 有损陷门函数 (LTDF) 示意图

定理 4.3

令 $\mathcal{F}$ 是一族 (n,τ)-LTDF, 当 $n-\tau \geq \omega(\log\kappa)$ 时, $\mathcal{F}$ 的单射模式构成一族单射单向陷门函数.

证明 证明通过游戏序列组织.

Game_0: 该游戏是标准的单射单向陷门函数安全游戏. 挑战者 $\mathcal{CH}$ 和敌手 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 运行 $pp \leftarrow \mathsf{Setup}(\kappa)$, $(ek,td) \leftarrow \mathsf{GenInjective}(pp)$, 发送 (pp,ek) 给 $\mathcal{A}$.
- 挑战阶段: $\mathcal{CH}$ 随机选择 $x^* \xleftarrow{R} X$, 发送 $y^* \leftarrow f_{ek}(x^*)$ 给 $\mathcal{A}$.
- 猜测阶段: $\mathcal{A}$ 输出 x', $\mathcal{A}$ 赢得游戏当且仅当 $x'=x^*$.

根据定义, 我们有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr[S_0]$$

Game_1: 该游戏与上一个游戏完全相同, 唯一不同的是将单射模式切换到有损模式

- 初始化: $\mathcal{CH}$ 运行$(ek,\perp) \leftarrow \mathsf{GenLossy}(pp)$生成求值公钥 ek.

根据定义, 我们有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr[S_1]$$

断言 4.1

单射和有损两种模式的计算不可区分性保证了 $|\Pr[S_0]-\Pr[S_1]| \leq \mathsf{negl}(\kappa)$.

证明 我们利用反证法完成归约论证: 若 $|\Pr[S_0]-\Pr[S_1]|$ 不可忽略, 则可构造出 PPT 的敌手 $\mathcal{B}$ 打破模式的不可区分性. $\mathcal{B}$ 在收到模式不可区分性的挑战 (pp,ek) 后, 将 (pp,ek) 发送给 $\mathcal{A}$, 随后随机选取 $x^* \xleftarrow{R} X$, 计算并发送 $y^* \leftarrow f_{ek}(x^*)$ 给 $\mathcal{A}$. 当收到 $\mathcal{A}$ 的输出 x' 后, 若 $x'=x^*$, $\mathcal{B}$ 输出 '1', 否则输出 '0'. 分析可知, 当 ek 来自单射模式时, $\mathcal{B}$ 完美的模拟了 Game_0; 当 ek 来自有损模式时, $\mathcal{B}$ 完美的模拟了 Game_1. 因此, 我们有:

$$|\Pr[\mathcal{B}(ek)=1 : ek \leftarrow \mathsf{GenInjective}(pp)] - \Pr[\mathcal{B}(ek)=1 : ek \leftarrow \mathsf{GenLossy}(pp)]| = |\Pr[S_0]-\Pr[S_1]|$$

其中 $pp \leftarrow \mathsf{Setup}(1^\kappa)$. 单射和有损两种模式的计算不可区分性保证了 $|\Pr[S_0]-\Pr[S_1]| \leq \mathsf{negl}(\kappa)$. □

断言 4.2

对于任意的敌手 $\mathcal{A}$(即使拥有无穷计算能力), 其在 Game_1 中的优势也是可忽略的.

证明 Game_1 处于有损模式, 因此由 Chaining Lemma 2.2可知, x^* 的平均条件最小熵 $\tilde{\mathsf{H}}_\infty(x^*|y^*) \geq n-\tau \geq \omega(\log\kappa)$, 从而即使拥有无穷计算能力的敌手在 Game_1 中的优势也是可忽略的. □

综合以上, 定理得证! □

注记 4.3

有损陷门函数相比标准单向陷门函数多了有损模式, 也正因为如此, 其具有标准单向陷门函数很多不具备的优势.

在安全方面, 根据上述论证容易验证只要参数设置满足一定约束, 则有损 (陷门) 函数在泄漏模型下仍然安全. 具体的, 在敌手获得关于原像任意长度为 ℓ 有界泄漏的情形下, 只要 $n-\tau-\ell \geq \omega(\log\kappa)$, 则单向性依然成立. 因此, 有损 (陷门) 函数是构造抗泄漏单向函数的重要工具 [293, 73].

在效率方面, 令 $\mathcal{H}$ 是一族从 X 到 $\{0,1\}^{m(\kappa)}$ 的对独立哈希函数族 (pairwise-independent hash family), 只要 $n-\tau-m \geq \omega(\log\kappa)$, 那么从 $\mathcal{H}$ 中随机选择的 h 即构成单向函数的多比特输出硬核函数. 论证的方式是应用 Conditional Leftover Hash Lemma 2.3和对独立哈希函数族构成强随机性提取器的事实, 得到硬核函数输出和均匀随机输出不可区分的结论.

有损陷门函数还有一个非平凡的扩展, 称为全除一 (ABO, All-But-One) 有损陷门函数. 简言之, ABO-LTDF 存在一个分支集合 (branch set), 记为 B. 求值密钥 ek 和分支值 $b \in B$ 共同定义了从 X 到 Y 的函数 $f_{ek,b}$, 该函数当且仅当 b 等于某特定分支值时有损, 在其它分支均单射可逆. ABO-LTDF 的功能与安全性如图 4.2所示.

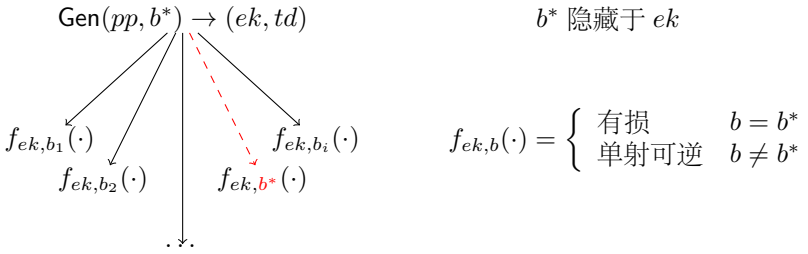


图 4.2: 全除一有损陷门函数 (ABO-LTDF)

定义 4.4 (全除一有损陷门函数 (ABO-LTDF))

ABO-LTDF 由 n 和 τ 两个参数刻画, 包含以下 5 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 pp, 其中 pp 包含对定义域 X、值域 Y 和分支集合 B 的描述. 其中 $|X| = 2^{n(\kappa)}$.
- $\mathsf{Gen}(pp, b^*)$: 以公共参数 pp 和给定分支值 $b^* \in B$ 为输入, 输出密钥对 (ek, td), 其中 ek 为求值公钥, td 为求逆陷门. 该算法输出的 ek 和分支值 $b \in B$ 定义了从 X 到 Y 的函数 $f_{ek,b}$. 当 $b \neq b^*$ 时, $f_{ek,b}$ 单射且拥有对应 td 可高效求逆; 当 $b = b^*$ 时, f_{ek,b^*} 有损, 像集的大小至多为 $2^{\tau(\kappa)}$, b^* 因此称为有损分支.
- $\mathsf{Eval}(ek, b, x)$: 以求值公钥 ek、分支值 $b \in B$ 和定义域元素 $x \in X$ 为输入, 输出 $y \leftarrow f_{ek,b}(x)$.
- $\mathsf{TdInv}(td, b, y)$: 以求逆陷门 td、分支值 $b \in B$ 和值域元素 $y \in Y$ 为输入, 输出 $x \in X$ 或特殊符号 $\perp$ 指示 y 不存在原像.

全除一有损陷门函数须满足以下性质:

有损分支隐藏性. 该性质刻画的安全性质是求值公钥不泄漏有损分支的信息. 严格定义类似公钥加密的不可区分安全或是承诺的隐藏性, 即 $\forall b_0, b_1 \in B$, 我们有:

$$\mathsf{Gen}(pp, b_0) \approx_c \mathsf{Gen}(pp, b_1)$$

其中 $pp \leftarrow \mathsf{Setup}(1^\kappa)$.

注记 4.4

ABO-LTDF 可以理解为 LTDF 的扩展, 分支集合由 $\{0,1\}$ 延拓至 $\{0,1\}^b$. LTDF 已经有较为丰富的应用, 如 IND-CPA 的公钥加密方案、不经意传输、抗碰撞哈希函数等; LTDF 与 ABO-LTDF 结合有着更强的应用,

如 IND-CCA 的公钥加密方案. IND-CCA 的公钥加密方案构造原理蕴含在如何基于 LTDF 和 ABO-LTDF 构造更高级的单向陷门函数中 (将在章节中阐述), 为了避免重复, 此处不再详述.

以下展示如何给出 LTDF 和 ABO-LTDF 的具体构造. 构造的难点是需要巧妙设计密钥对生成算法, 使其可以工作在单射可逆和有损两个模式, 且两种模式在计算意义下不可区分. 设计的思路是令定义域 X 是向量空间, 输入 x 是向量空间中的元素, 求值公钥 ek 是刻画线性变换的矩阵, 函数求值 $f(ek, x)$ 的过程就是对输入进行线性变换, 当 ek 满秩时, 函数单射可逆; 当 ek 非满秩时, 函数有损. 隐藏 ek 工作模式的思路则是对其 "加密". 我们称上述技术路线为矩阵式方法.

下面展示矩阵式构造的一个具体例子, 以剥丝抽茧的方式阐明设计思想和关键技术.

隐藏矩阵生成. 最简单的满秩矩阵是单位阵, 最简单的非满秩矩阵是全零阵, 两者之间差异显著, 为了保证计算不可区分性, 思路是生成一个伪随机的隐藏矩阵 (concealer matrix) $\mathbf{M}$ 对其加密. 我们期望 $\mathbf{M}$ 满足如下结构: $\mathbf{M}$ 的所有行向量均处于同一个一维子空间, 后面可以看到子空间的描述将作为陷门信息使用. 具体的, 隐藏矩阵生成算法 $\mathsf{GenConcealMatrix}(n)$ 细节如下:

1. 随机选择 $\mathbf{r} = (r_1, \ldots, r_n) \xleftarrow{R} \mathbb{Z}_q^n$ 和 $\mathbf{s} = (s_1, \ldots, s_n, 1) \xleftarrow{R} \mathbb{Z}_q^n \times \{1\}$
2. 计算张量积 $\mathbf{V} = \mathbf{r} \otimes \mathbf{s} = \mathbf{r}^t\mathbf{s} \in \mathbb{Z}_q^{n\times(n+1)}$

$$\mathbf{V} = \left(\begin{array}{cccc|c} r_1s_1 & r_1s_2 & \ldots & r_1s_n & r_1 \\ r_2s_1 & r_2s_2 & \ldots & r_2s_n & r_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ r_ns_1 & r_ns_2 & \ldots & r_ns_n & r_n \end{array}\right)$$

3. 输出 $\mathbf{M} = g^{\mathbf{V}} \in \mathbb{G}^{n\times(n+1)}$ 作为隐藏矩阵, $\mathbf{s}$ 作为陷门信息.

$$\mathbf{M} = \left(\begin{array}{cccc|c} g^{r_1s_1} & g^{r_1s_2} & \ldots & g^{r_1s_n} & g^{r_1} \\ g^{r_2s_1} & g^{r_2s_2} & \ldots & g^{r_2s_n} & g^{r_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g^{r_ns_1} & g^{r_ns_2} & \ldots & g^{r_ns_n} & g^{r_n} \end{array}\right)$$

算法前两步的作用是生成特定结构: 通过张量积确保 $\mathbf{V}$ 中所有行向量均处于向量 $(s_1, \ldots, s_n, 1)$ 张成的一维子空间中. 当前向量定义在有限域 $\mathbb{F}_p$ 上, 而 ek 矩阵不可以定义在有限域 $\mathbb{F}_p$ 上, 否则存在高效的算法判定 ek 对应的矩阵是否满秩. 令 $\mathbb{G}$ 是 p 阶循环群, 其中 DDH 假设成立. 可以证明, 如果 ek 矩阵定义在 $\mathbb{G}$ 上, 那么满秩和非满秩无法有效判定. 因此, 算法的第三步利用从 $\mathbb{F}_p$ 到 $\mathbb{G}$ 的同构映射 $\phi : t \to g^t$ 将 $\mathbb{V}$ 中的所有元素从 $\mathbb{F}_p$ 提升到 $\mathbb{G}$ 中.

注记 4.5

如果将 $\mathbf{s}$ 截断为 $\mathbf{s}' = (s_1, \ldots, s_n)$, 那么 $g^{\mathbf{r}\otimes\mathbf{s}'} = (g^{r_i\cdot s_j}) \in \mathbb{G}^{n\times n}$ 恰好是 Naor-Reingold 基于 DDH 假设的伪随机合成器构造 (pseudorandom synthesizer)

- 伪随机合成器 $f(r, s)$ 是满足如下性质的函数: 令 $r_1, \ldots, r_n$ 和 $s_1, \ldots, s_m$ 独立随机分布, 当输入 (r, s) 取遍 (r_i, s_j) 组合时, 输出伪随机.
- Naor 和 Reingold 证明了从 $\mathbb{Z}_q \times \mathbb{Z}_q$ 映射到 $\mathbb{G}$ 的函数 $f(r, s) = g^{rs}$ 是基于 DDH 假设的伪随机合成器.

引理 4.1

如果 DDH 假设成立, 那么由 $\mathsf{GenConcealMatrix}(n)$ 生成的矩阵 $\mathbf{M} = g^{\mathbf{V}}$ 在 $\mathbb{G}^{n\times(n+1)}$ 上伪随机.

证明 证明的过程分为两个步骤, 我们首先在一行上从左至右逐个列元素进行混合论证, 证明其与 $\mathbf{G}^{n+1}$ 上的随机向量计算不可区分, 再利用该结论从上到下逐行进行混合论证, 从而证明隐藏矩阵 $\mathbf{M}$ 在 $\mathbf{G}^{n\times(n+1)}$ 上伪随机分布.

- 逐列论证: 令 $r \xleftarrow{R} \mathbb{Z}_q, \mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n, \mathbf{t} \xleftarrow{R} \mathbb{Z}_q^n$, 证明如下两个分布计算不可区分:

$$(g^{\mathbf{s}}, g^r, \mathbf{y} = g^{r\cdot\mathbf{s}}) \approx_c (g^{\mathbf{s}}, g^r, \mathbf{y} = g^{\mathbf{t}})$$

  证明的方法是设计如下的游戏序列进行混合论证:

$$\begin{aligned}
\mathrm{Hyb}_0 &: g^{\mathbf{s}} \quad g^{rs_1} \quad \dots \quad g^{rs_n} \quad g^r \\
\mathrm{Hyb}_1 &: g^{\mathbf{s}} \quad g^{t_1} \quad \dots \quad g^{rs_n} \quad g^r \\
\mathrm{Hyb}_j &: g^{\mathbf{s}} \quad g^{t_1} \quad g^{t_j} \quad g^{rs_n} \quad g^r \\
\mathrm{Hyb}_n &: g^{\mathbf{s}} \quad g^{t_1} \quad \dots \quad g^{t_n} \quad g^r
\end{aligned}$$

  基于 DDH 假设, 可以证明任意两个相邻的游戏中定义的分布簇均计算不可区分, 利用 hybrid lemma 立刻可得: $\mathrm{Hyb}_0 \approx_c \mathrm{Hyb}_1$.
- 逐行论证: 基于上述结果, 我们再逐行变换, 每次将一行替换成 $\mathbf{G}^{n+1}$ 上的随机向量, 再次利用 hybrid lemma 即可证明

$$(g^{\mathbf{s}}, \mathbf{M}) \approx_c (g^{\mathbf{s}}, U_{\mathbb{G}^{n\times(n+1)}}) \tag{4.1}$$

综上, $\mathbf{M}$ 在 $\mathbb{G}^{n\times(n+1)}$ 上伪随机分布. □

注记 4.6

公式 (4.1)事实上证明了比引理更强的结果, 即在敌手观察到 $g^{\mathbf{s}}$ 的情形下, $\mathbf{M}$ 仍与 $\mathbb{G}^{n\times(n+1)}$ 上随机矩阵计算不可区分. 在以上两个步骤的证明过程中, 横向的归约损失是 n, 纵向的归约损失为 n, 因此证明的总归约损失是 n^2. 可以利用 DDH 类假设的随机自归约性质 (random self-reducibility) 将归约损失降为 n.

以下首先展示基于 DDH 假设的 LTDF 构造.

构造 4.3 (基于 DDH 假设的 LTDF 构造)

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{GenGroup}(1^\kappa) \to (\mathbb{G}, q, g)$, 其中 $\mathbb{G}$ 是一个阶为素数 q 的循环群, 生成元为 g. 输出 $pp = (\mathbb{G}, q, g)$. pp 还包括了定义域 $X = \{0,1\}^n$ 和值域 $Y = \mathbb{G}$ 的描述.
- $\mathsf{GenInjective}(n)$: 运行 $\mathsf{GenConcealMatrix}(n) \to (g^{\mathbf{V}}, \mathbf{s})$, 输出 $g^{\mathbf{Z}} = g^{\mathbf{V}+\mathbf{I}'}$ 作为公钥 ek, 其中 $\mathbf{I}' \in \mathbb{Z}_q^{n\times(n+1)}$ 由 n 阶单位阵在最右侧补上全零列扩展得来 (即 $(\mathbf{e}_1, \dots, \mathbf{e}_n, \mathbf{0})$), 输出 $\mathbf{s}$ 作为函数的陷门 td.

$$g^{\mathbf{Z}} = \left(\begin{array}{cccc|c}
g^{r_1s_1+1} & g^{r_1s_2} & \dots & g^{r_1s_n} & g^{r_1} \\
g^{r_2s_1} & g^{r_2s_2+1} & \dots & g^{r_2s_n} & g^{r_2} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
g^{r_ns_1} & g^{r_ns_2} & \dots & g^{r_ns_n+1} & g^{r_n}
\end{array}\right)$$

- $\mathsf{GenLossy}(n)$: $\mathsf{GenConcealMatrix}(n) \to g^{\mathbf{V}}$, 输出 $g^{\mathbf{Z}} = g^{\mathbf{V}}$ 作为公钥 ek, 陷门 td 为 $\perp$.

$$g^{\mathbf{Z}} = \left(\begin{array}{cccc|c}
g^{r_1s_1} & g^{r_1s_2} & \dots & g^{r_1s_n} & g^{r_1} \\
g^{r_2s_1} & g^{r_2s_2} & \dots & g^{r_2s_n} & g^{r_2} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
g^{r_ns_1} & g^{r_ns_2} & \dots & g^{r_ns_n} & g^{r_n}
\end{array}\right)$$

- $\mathsf{Eval}(ek, \mathbf{x})$: 以 $ek = g^{\mathbf{Z}}$ 和 $\mathbf{x} \in \{0,1\}^n$ 为输入, 计算 $\mathbf{y} \leftarrow g^{\mathbf{xZ}} \in \mathbb{G}^{n+1}$.
- $\mathsf{TdInv}(td, \mathbf{y})$: 解析 $td = \mathbf{s} = (s_1, \dots, s_n)$, 对每个 $i \in [n]$, 计算 $a_i = y_i / y_{n+1}^{s_i}$ 并输出 $x_i \in \{0,1\}$ s.t. $a_i = g^{x_i}$.

定理 4.4

基于 DDH 假设，上述构造是一族 $(n, \log p)$-LTDF.

证明 单射可逆模式的正确性由算法 TdInv 的正确性保证. 在有损模式下，所有输出 $\mathbf{y}$ 都具有 g^{cs} 的形式，其中 $c = \langle \mathbf{x}, \mathbf{r} \rangle \in \mathbb{Z}_q$. 向量 $\mathbf{s}$ 被 ek 固定，因此 $\mathrm{Img}(f_{ek}) \leq q$.

单射可逆模式和有损模式的计算不可区分性由 GenConcealMatrix 输出的伪随机性 (引理 4.1) 保证. □

下面展示如何基于 DDH 假设构造 ABO-LTDF.

构造 4.4 (基于 DDH 假设的 ABO-LTDF 构造)

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{GenGroup}(1^\kappa) \to (\mathbb{G}, q, g)$，其中 $\mathbb{G}$ 是一个阶为素数 q 的循环群，生成元为 g. 输出 $pp = (\mathbb{G}, q, g)$. pp 还包括定义域 $X = \{0,1\}^n$、值域 $Y = \mathbb{G}$ 和分支集合 $B = \mathbb{Z}_q$ 的描述.
- $\mathsf{Gen}(pp, b^*)$: 运行 $\mathsf{GenConcealMatrix}(n) \to (g^{\mathbf{V}}, \mathbf{s})$，输出 $g^{\mathbf{Z}} = g^{\mathbf{V} - b^* \mathbf{I}'}$ 作为公钥 ek，其中 $\mathbf{I}' = (\mathbf{e}_1, \ldots, \mathbf{e}_n, \mathbf{0}) \in^{n \times (n+1)}$，输出 $(b^*, \mathbf{s})$ 作为陷门 td.
- $\mathsf{Eval}(ek, b, \mathbf{x})$: 以 $ek = g^{\mathbf{Z}}$ 和 $\mathbf{x} \in \{0,1\}^n$ 为输入，计算 $\mathbf{y} \leftarrow g^{\mathbf{x}(\mathbf{Z} + b(\mathbf{e}_1, \ldots, \mathbf{e}_n, \mathbf{0}))} \in \mathbb{G}^{n+1}$，记为 $y \leftarrow f(ek, b, x)$ 或 $y \leftarrow f_{ek,b}(x)$.
- $\mathsf{TdInv}(td, b, \mathbf{y})$: 解析 td 为 $\mathbf{s} = (s_1, \ldots, s_n)$，对每个 $i \in [n]$，计算 $a_i = y_i / y_{n+1}^{s_i}$ 并输出 $x_i \in \{0,1\}$ s.t. $a_i = g^{(b-b^*)x_i}$.

$$\mathsf{Gen}(pp, b^*) \to (ek, \mathbf{s})$$

$$\mathsf{GenConcealMatrix}(n) = g^{\mathbf{V}}$$

$$\mathbf{x} \in \mathbb{Z}_2^n \times \left(\begin{array}{cccc|c} g^{r_1 s_1} & g^{r_1 s_2} & \cdots & g^{r_1 s_n} & g^{r_1} \\ g^{r_2 s_1} & g^{r_2 s_2} & \cdots & g^{r_2 s_n} & g^{r_2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g^{r_n s_1} & g^{r_n s_2} & \cdots & g^{r_n s_n} & g^{r_n} \end{array}\right) \begin{array}{l} -b^*(\mathbf{e}_1, \ldots, \mathbf{e}_n, \mathbf{0}) \\ +b(\mathbf{e}_1, \ldots, \mathbf{e}_n, \mathbf{0}) \\ \to \mathbf{y} \in \mathbb{G}^{n+1} \end{array}$$

$$\mathrm{DDH} \Rightarrow \approx_c U_{\mathbb{G}^{n \times (n+1)}}$$

定理 4.5

基于 DDH 假设，上述构造是一族分支集合为 $B = \mathbb{Z}_q$ 的 $(n, \log p)$-ABO-LTDF.

证明 容易验证，当 $b \neq b^*$ 时，$\mathbf{V} + (b - b^*)\mathbf{I}'$ 矩阵满秩，$f_{ek,b}$ 单射且可高效求逆；当 $b = b^*$ 时，矩阵 $\mathbf{V} + (b - b^*)\mathbf{I}'$ 的秩为 1，$\mathrm{Img}(f_{ek,b}) \leq p$. 有损分支隐藏性由 GenConcealMatrix 输出的伪随机性 (引理 4.1) 保证. □

注记 4.7

为了确保求逆算法的高效性，以上构造有两个重要的设定：(1) 首先在 ConcealMatrix 设置了辅助列 $(g^{r_1}, \ldots, g^{r_n})^T$，便于计算出 $a_i = g^{x_i}$；(2) 从 a_i 中计算 x_i 需要求解离散对数，因此定义域 X 设定为 $\mathbb{Z}_2^n$，其中 2 可以进一步放宽至 $\kappa^{O(1)}$(关于 κ 的多项式规模)，以保证可以在多项式时间完成离散对数求解.

扩展与深化

注意到在公钥加密的选择密文安全定义中敌手对解密谕言机的访问权限是全除一的，由此可以看出全除一有损陷门函数的应用局限于“全除一”类安全的密码方案设计. Hofheinz [41] 引入了全除多有损陷门函数，将有损分支的数量从 1 扩展到 $\mathsf{poly}(\kappa)$，并展示了其在选择打开选择密文安全 (selective opening chosen-ciphertext security) 中的应用. 在有损陷门函数的应用中，我们通常期望有损模式下函数丢失的信息尽可能的多，即像集尽可能的小.

这是因为单射和有损模式的反差越大, 所蕴含的结果越强, 如更高的泄漏容忍能力、更紧的安全归约等. 但凡事有度, 物极必反, 在常规的一致归约 (universal reduction) 模型下, 有损模式的像集尺寸 2^{τ} 不能过小, 至少是关于计算安全参数 κ 的超多项式规模, 否则 PPT 的敌手可以通过生日攻击有效的区分单射和有损模式. Zhandry [294] 创造性的提出了极度有损函数 (extremely lossy functions, ELF). 在 ELF 中, 有损模式下函数的像集可以缩小至关于计算安全参数 κ 的多项式规模, 只要在指定 PPT 敌手的生日攻击能力之外即可. ELF 的有损模式之所以能够打破像集多项式界的关键在更为精细的个体归约 (individual reduction) 模型 [260] 下进行安全性证明. Zhandry 基于不可区分程序混淆给出了 ELF 的构造, 并展示了其强大的应用. 在无须求逆的应用场景中, 不仅不需要陷门, 甚至是单射的性质也可以弱化. 陈等 [72] 根据这一观察, 提出了规则有损函数 (regular lossy functions, RLF). 相比标准的 LTDF, RLF 将单射可逆模式放宽至规则有损, 即每个像的原像集合大小相同. 正是这一弱化, 使得 RLF 不仅有更加高效的具体构造, 也可由哈希证明系统通用构造得出, 并在抗泄漏密码学领域有着重要的应用.

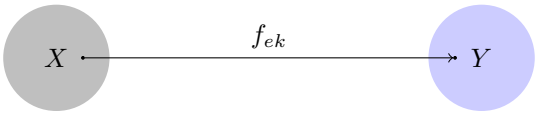
4.1.3 基于相关积单向陷门函数的构造

山重水复疑无路, 柳暗花明又一村.

— 宋 · 陆游《游山西村》

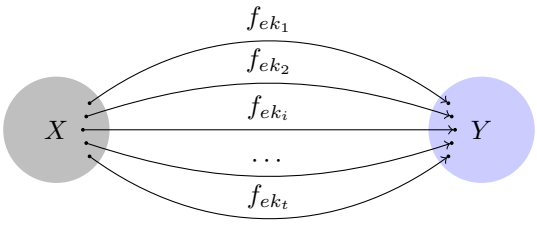
令 $\mathcal{F}=\{f_{ek}: X \to Y\}$ 是一族单向陷门函数, 可以自然对 $\mathcal{F}$ 进行 t 重延拓, 得到 $\mathcal{F}^t=\{g_{ek_1,\ldots,ek_t}: X^t \to Y^t\}$, 其中 $f^t_{ek_1,\ldots,ek_t}(x_1,\ldots,x_t) := (f_{ek_1}(x_1),\ldots,f_{ek_t}(x_t))$. 我们称 $\mathcal{F}^t$ 为 $\mathcal{F}$ 的 t 重积 (t-wise product).

$$\mathcal{F}=\{f_{ek}: X \to Y\}$$



t-wise product

$$\mathcal{F}^t=\{f^t_{ek_1,\ldots,ek_t}: X^t \to Y^t\}$$



定理 4.6

如果 $\mathcal{F}=\{f_{ek}\}$ 是一族单向函数, 那么它的 t 重积 $\mathcal{F}^t=\{f^t_{ek_1,\ldots,ek_t}\}$ 也是一族单向函数. ♡

证明 证明的思路简述如下: 如果存在 PPT 的敌手 $\mathcal{A}$ 打破 $\mathcal{F}^t$ 的单向性, 那么其必然以不可忽略的优势对 t 个单向函数的实例 $f_{ek_i}(\cdot)$ 求逆, 这显然与 $\mathcal{F}$ 的单向性冲突, 因此得证.

注记 4.8

上述定理在 $ek_1=\cdots=ek_t$(即所有 f_{ek_i} 相同) 时仍然成立, 该情形恰好对应单向函数的单向性放大 (one-wayness amplification). ♠

需要注意的是, $f^t_{ek_1,\ldots,ek_t}$ 单向性成立的前提是各分量输入 x_i 独立随机采样, 而当各分量输入相关时, 单向性则未必成立, 这是因为多个像的分量交叉组合可能会泄漏原像的信息.

构造 4.5 (相关积单向性与标准单向性的分离反例构造)

令 $\hat{f}_{ek} : X = \{0,1\}^n \to Y$ 是一个单向函数, 构造一个新的函数 $f_{ek} : \{0,1\}^{2n} \to Y||\{0,1\}^n$ 如下:

$$f_{ek}(x_l||x_r) := \hat{f}_{ek}(x_l)||x_r$$

♣

在上述构造中, f_{ek} 以 $\hat{f}_{ek}$ 为核, 因此如果 $\hat{f}_{ek}$ 是单向的, 那么 f_{ek} 也是单向的. 考察 2 重积 $f^2_{ek_1,ek_2}$ 在相关输入 $(x_1 = x_l||x_r, x_2 = x_r||x_l)$ 下的行为:

$$f^2_{ek_1,ek_2}(x_1,x_2) := (f_{ek_1}(x_1), f_{ek_2}(x_2)) = \hat{f}_{ek_1}(x_l)||x_r||\hat{f}_{ek_2}(x_r)||x_l$$

根据 f_{ek} 的设计, $f^2_{ek_1,ek_2}$ 的原像信息 (x_1,x_2) 可以从像中的 (x_r,x_l) 完全恢复出来, 因此在输入呈如上相关时并不满足单向性. 上述反例构造的精髓是设计具有特殊结构的单向函数.

反例 4.5说明单向函数的 t 重积在输入相关时并不一定仍然单向. Alon 和 Rosen [32] 引入了相关积 (correlated products) 单向陷门函数, 定义如下: 要求函数的 t 重积在输入分量相关时仍然保持单向性

定义 4.5 (相关积单向性)

令 $\mathcal{F} : X \to Y$ 是一族单向函数, $\mathcal{C}_t$ 是定义在 X^t 上的分布 (分量相关). 如果 $\mathcal{F}$ 的 t 重积 $\mathcal{F}^t : X^t \to Y^t$ 在 $\mathcal{C}_t$ 相关积下仍然是单向的, (即对于任意 PPT 敌手 $\mathcal{A}$, 其在如下的安全实验中优势是可忽略的)

$$\Pr\left[f^t_{ek_1,\ldots,ek_t}(x') = y^* : \begin{array}{l} ek_i \leftarrow \mathsf{Gen}(\kappa); \\ (x_1^*,\ldots,x_t^*) \xleftarrow{R} \mathcal{C}_t; \\ y^* \leftarrow (f_{ek_1}(x_1^*),\ldots,f_{ek_t}(x_t^*)); \\ x' \leftarrow \mathcal{A}(ek_1,\ldots,ek_t,y^*); \end{array}\right]$$

则称 $\mathcal{F}$ 是 $\mathcal{C}_t$ 相关积安全的 (correlated-product secure). 该定义可以自然延拓到陷门函数场景.

♣

在给出 CP-TDF 的定义后, 接下来需要研究的问题是分析什么样的 $\mathcal{F}$ 在何种相关积下仍然单向. 本书中聚焦最为典型的一种 $\mathcal{C}_t$ 相关积——均匀重复相关积 $\mathcal{U}_t$, 即 $x_1 \xleftarrow{R} X$ 且 $x_1 = \cdots = x_t$. Rosen 和 Segev [32] 基于 LTDF 给出了 CP-TDF 的一个通用构造, 揭示了两者之间的联系.

定理 4.7

令 $\mathcal{F}$ 是一族 (n,τ)-LTDF, 那么 $\mathcal{F}$ 在相关积 $\mathcal{U}_t$ 下仍然单向, 其中 $t \le (n - \omega(\log\kappa))/\tau$.

♡

证明 证明通过以下的游戏序列完成, 敌手在 Game_i 中成功的事件为 S_i.

Game_0: 对应真实的相关积单向性实验, 函数以单射模式运作

- $\mathcal{CH}$ 独立运行 $\mathcal{F}.\mathsf{GenInjective}(\kappa)$ 算法 t 次, 生成 $ek = (ek_1,\ldots,ek_t)$ 并将其发送给 $\mathcal{A}$.
- $\mathcal{CH}$ 随机采样 $x^* \xleftarrow{R} X$, 计算 $y^* \leftarrow (f_{ek_1}(x^*),\ldots,f_{ek_t}(x^*))$ 并将 y^* 发送给 $\mathcal{A}$.
- $\mathcal{A}$ 输出 x', 当且仅当 $x' = x^*$ 时成功.

根据定义, 我们有:

$$\mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\kappa) = \Pr[S_0]$$

Game_1: 与上一游戏相同, 区别在于函数切换到有损模式运作

- $\mathcal{CH}$ 独立运行$\mathcal{F}.\mathsf{GenLossy}(\kappa)$算法 t 次, 生成 $ek = (ek_1,\ldots,ek_t)$ 并将其发送给 $\mathcal{A}$.

$$\mathsf{Adv}^{\mathsf{Game}_1}_{\mathcal{A}}(\kappa) = \Pr[S_1]$$

断言 4.3

基于 LTDF 的单射/有损模式不可区分性, 任意 PPT 敌手 $\mathcal{A}$ 在 Game_0 和 Game_1 中的成功概率差可忽略.

♡

证明 Game_0 和 Game_1 的差别在于 $(ek_1,\ldots,ek_t)$ 的生成模式. 基于 LTDF 的单射/有损模式不可区分性和 hybrid argument, 可以推出 $\mathsf{Game}_0 \approx_c \mathsf{Game}_1$, 进而保证 $|\Pr[S_0] - \Pr[S_1]| \le \mathsf{negl}(\kappa)$.

断言 4.4

对于任意敌手 $\mathcal{A}$(即使拥有无穷计算能力), $\Pr[S_1] = \mathsf{negl}(\kappa)$.

证明 在 Game_1 中, 函数工作在有损模式, 因此像集的大小至多为 $2^{t\cdot\tau}$, 由 Chainning Lemma 2.2可知 x^* 的平均最小熵 $\tilde{\mathsf{H}}_\infty(x^*|y^*) \geq n - t\tau$. 根据定理前提条件中的参数选取, 有 $\tilde{\mathsf{H}}_\infty(x^*|y^*) \geq \omega(\log\kappa)$, 因此断言得证. □

综上, 我们有 $\Pr[S_0] \leq \mathsf{negl}(\kappa)$. 定理得证! □

笔记 追求简洁、消除冗余在科学和文学领域似乎都是真理. 然而, 正如知乎上一篇文章 [295] 所说:"尽管我们偏爱简洁, 但冗余让一切皆有可能". 相关积单向函数的定义和构造就充分诠释了冗余的力量.

4.1.4 基于自适应单向陷门函数的构造

他强由他强, 清风拂山冈; 他横由他横, 明月照大江; 他自狠来他自恶, 我自一口真气足.

— 达摩《九阳真经》

构造 4.2仅具备 IND-CPA 安全性, 并不一定能够满足 IND-CCA 安全性. 这是因为底层的 TDF 可能具备诸如同态等优良的代数性质, 使得上层 PKE/KEM 方案具有可延展性. 从安全归约的角度分析, 归约算法无法对解密/解封装询问做出正确的应答. 基于以上分析, 一个自然的问题是: TDF 满足何种增强的性质才能够使得构造 4.2满足 IND-CCA 安全性.

Kiltz, Mohassel 和 O'Neill [33] 提出了自适应单向性 (adaptive one-wayness), 该性质要求 TDF 的单向性在敌手能够访问求逆谕言机的情况下仍然成立.

定义 4.6 (自适应单向性)

令 $\mathcal{F}$ 是一族陷门函数, 定义敌手 $\mathcal{A}$ 的优势如下:

$$\Pr\left[x' \in f_{ek}^{-1}(y^*) : \begin{array}{l} pp \leftarrow \mathsf{Setup}(\kappa); \\ (ek, td) \leftarrow \mathsf{KeyGen}(pp); \\ x^* \xleftarrow{R} X, y^* \leftarrow f_{ek}(x^*); \\ x' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{inv}}}(ek, y^*); \end{array}\right]$$

其中 $\mathcal{O}_{\mathsf{inv}}$ 是求逆谕言机, $\forall y \neq y^*, \mathcal{O}_{\mathsf{inv}}(y) = \mathsf{TdInv}(td, y)$. 如果任意 PPT 敌手 $\mathcal{A}$ 在上述安全试验中的优势均为 $\mathsf{negl}(\kappa)$, 那么则称 $\mathcal{F}$ 是自适应单向的.

为了方便在公钥加密场景中的应用, 引入自适应伪随机性如下.

定义 4.7 (自适应伪随机性)

令 $\mathcal{F}$ 是一族单向函数, hc 是其硬核函数. 定义敌手 $\mathcal{A}$ 的优势如下:

$$\Pr\left[\beta' = \beta : \begin{array}{l} (ek, td) \leftarrow \mathcal{F}.\mathsf{Gen}(); \\ x^* \xleftarrow{R} X, y^* \leftarrow f_{ek}(x^*); \\ k_0^* \leftarrow \mathsf{hc}(x^*), k_1^* \xleftarrow{R} K, \beta \xleftarrow{R} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{inv}}}(ek, y^*, k_\beta^*); \end{array}\right] - \frac{1}{2}$$

其中 $\mathcal{O}_{\mathsf{inv}}$ 是求逆谕言机, hc 是 $\mathcal{F}$ 的硬核函数. 如果任意 PPT 敌手 $\mathcal{A}$ 在上述安全试验中的成功概率均为 $\mathsf{negl}(\kappa)$, 那么则称 hc 的是自适应伪随机的.

推论 4.1

$\mathcal{F}$ 的自适应单向性蕴含硬核函数的自适应伪随机性.

证明 Goldreich-Levin 定理的证明可以平行推广到求逆谕言机 $\mathcal{O}_{\mathsf{inv}}$ 存在的情形下, $\mathsf{hc}(x^*)$ 自适应伪随机性由 x^* 的自适应单向性保证. □

自适应单向陷门函数 (ATDF, adaptive TDF) 定义简洁, 威力强大, 将 ATDF 代入构造 4.2中, 得到的 KEM 满足 IND-CCA 安全. 从安全归约的角度观察, ATDF 的自适应单向性是为 KEM 的 CCA 安全性量身定制的, 都是 "全除一" 类型的安全定义. 那么, 如何构造 ATDF 呢? 文献 [33] 一方面基于实例独立 (instance-independent) 假设给出 ATDF 的具体构造, 一方面分别基于 LTDF 和 CP-TDF 给出了 ATDF 的两个通用构造.

以下我们聚焦 ATDF 的通用构造, 首先展示如何基于 LTDF 构造 ATDF. 构造的技术困难点在于 ATDF 的安全试验中挑战者 $\mathcal{CH}$ 向敌手 $\mathcal{A}$ 提供了 "全除一" 式解密谕言机 $\mathcal{O}_{\mathsf{inv}}$, 而 LTDF 的安全试验中并没有提供类似的谕言机访问接口. 因此, 构造的思路是通过引入精巧的结构完成解密谕言机 $\mathcal{O}_{\mathsf{inv}}$ 的模拟. 总体的思路如下:

- 令 ATDF 的像 y 形如 (y_0, y_1), 确保 y_1 由 y_0 惟一确定, 可行的设计是计算原像 x 的 LTDF 值作为 y_0, 再以 y_0 为分支编号计算 x 的 ABO-LTDF 值作为 y_1.

$$y_0 \leftarrow f(ek_{\text{ltdf}}, x), y_1 \leftarrow g(ek_{\text{abo}}, y_0, x)$$

- 上述设计利用 ABO-LTDF 的相对分支标签的 "全除一" 求逆陷门嵌入了相对于像的 "全除一" 可逆结构.

构造 4.6 (基于 LTDF 和 ABO-LTDF 的 ATDF 构造)

构造所需的组件是:

- (n, τ_1)-LTDF – $\mathcal{F} : X \to Y_1$;
- (n, τ_2)-ABO-LTDF – $\mathcal{G} : X \to Y_2$ w.r.t. Y_1 作为分支集合;

其中 $\log_2 |X| = n, \log_2 |Y_1| = m_1, \log_2 |Y_2| = m_2$.

构造 ATDF : $X \to Y_1 \times Y_2$ 如下:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 计算 $pp_{\text{ltdf}} \leftarrow \mathcal{F}.\mathsf{Setup}(1^\kappa)$, $pp_{\text{abo}} \leftarrow \mathcal{G}.\mathsf{Setup}(1^\kappa)$, 输出 $pp = (pp_{\text{ltdf}}, pp_{\text{abo}})$.
- $\mathsf{Gen}(pp)$: 以公开参数 $pp = (pp_{\text{ltdf}}, pp_{\text{abo}})$ 为输入, 计算 $(ek_{\text{ltdf}}, td_{\text{ltdf}}) \leftarrow \mathcal{F}.\mathsf{GenInjective}(pp_{\text{ltdf}})$, $(ek_{\text{abo}}, td_{\text{abo}}) \leftarrow \mathcal{G}.\mathsf{Gen}(pp_{\text{abo}}, 0^{m_1})$, 输出求值公钥 $ek = (ek_{\text{ltdf}}, ek_{\text{abo}})$ 和陷门 $td = (td_{\text{ltdf}}, td_{\text{abo}})$.
- $\mathsf{Eval}(ek, x)$: 以求值公钥 $ek = (ek_{\text{ltdf}}, ek_{\text{abo}})$ 和 $x \in \{0,1\}^n$ 为输入, 计算 $y_1 \leftarrow f_{ek_{\text{ltdf}}}(x)$, $y_2 \leftarrow g_{ek_{\text{abo}}}(y_1, x)$, 输出 $y = (y_1, y_2)$.
- $\mathsf{TdInv}(td, y)$: 以陷门 $td = (td_{\text{ltdf}}, td_{\text{abo}})$ 和 $y = (y_1, y_2)$ 为输入, 计算 $x \leftarrow \mathcal{F}.\mathsf{TdInv}(td_{\text{ltdf}}, y_1)$, 验证 $y_2 \stackrel{?}{=} g_{ek_{\text{abo}}}(y_1, x)$: 如果是输出 x, 否则输出 $\perp$.

♣

上述构造的正确性显然成立. 安全性由如下定理保证:

定理 4.8

基于 LTDF 和 ABO-LTDF 的安全性, 上述构造在 $n - \tau_1 - \tau_2 \geq \omega(\log \kappa)$ 构成一族 ATDF. ♡

证明 令 $(x^*, y^* = (y_1^*, y_2^*))$ 为单向挑战实例, 其中 x^* 是原像, y^* 是像. 证明的思路将像 $y^* = (y_1^*, y_2^*)$ 的计算方式从单射无损模式逐步切换到有损模式, 最终在信息论意义下论证单向性.

Game_0: 真实的 ATDF 单向性试验. $\mathcal{CH}$ 与 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 进行如下操作
  1. 运行 $pp_{\text{ltdf}} \leftarrow \mathcal{F}.\mathsf{Setup}(1^\kappa)$, $pp_{\text{abo}} \leftarrow \mathcal{G}.\mathsf{Setup}(1^\kappa)$;
  2. 计算 $(ek_{\text{ltdf}}, td_{\text{ltdf}}) \leftarrow \mathcal{F}.\mathsf{GenInjective}(pp_{\text{ltdf}})$, $(ek_{\text{abo}}, td_{\text{abo}}) \leftarrow \mathcal{G}.\mathsf{Gen}(pp_{\text{abo}}, 0^{m_1})$;
  3. 发送 $pp = (pp_{\text{ltdf}}, pp_{\text{abo}})$ 和 $ek = (ek_{\text{ltdf}}, ek_{\text{abo}})$ 给 $\mathcal{A}$.
- 挑战: $\mathcal{CH}$ 随机选取 $x^* \xleftarrow{R} X$, 计算 $y_1^* \leftarrow f_{ek_{\text{ltdf}}}(x^*)$, $y_2^* \leftarrow g_{ek_{\text{abo}}}(y_1^*, x^*)$, 发送 $y^* = (y_1^*, y_2^*)$ 给 $\mathcal{A}$.
- 求逆询问: 当 $\mathcal{A}$ 向 $\mathcal{O}_{\mathsf{inv}}$ 询问 $y = (y_1, y_2)$ 的原像时, $\mathcal{CH}$ 分情况应答如下:
  - $y_1 = y_1^*$: 直接返回 $\perp$.
  - $y_1 \neq y_1^*$: 首先计算 $x \leftarrow \mathcal{F}.\mathsf{TdInv}(td_{\text{ltdf}}, y_1)$, 如果 $y_2 = g_{ek_{\text{abo}}}(y_1, x)$ 则返回 x, 否则返回 $\perp$.

  根据 ATDF 像的生成方式可知, 第一部分完全确定了第二部分, 当 $y_1 = y_1^*$ 时, 如 $y_2 = y_2^*$ 则 $\mathcal{A}$ 的询问为禁询点, 如 $y_2 \neq y_2^*$ 则像的格式不正确. 基于以上分析, $\mathcal{CH}$ 在应答形如 (y_1^*, y_2) 的求逆询问时, 无须进一步检

查第二部分 y_2, 直接返回 $\perp$ 即可保证应答的正确性.

Game_1: 在 Game_0 中 $\mathcal{CH}$ 使用 $\mathcal{F}$ 的陷门进行求逆, 因此 $\mathcal{F}$ 必须工作在单射可逆模式. 为了将 y_1^* 的计算模式切换到有损模式, 需要利用 $\mathcal{G}$ 的陷门进行求逆. 注意到在 Game_0 中 $\mathcal{G}$ 的 "全除一" 陷门根据预先设定的有损分支 0^{m_1} 生成, 因此必须先激活再使用, 因此 Game_1 的设计目的是为激活做准备:

- $\mathcal{CH}$ 在初始化阶段即随机采样 $x^* \xleftarrow{R} X$, 并计算 $y_1^* \leftarrow f_{ek_{\text{ltdf}}}(x^*)$.

与 Game_0 相比, Game_1 仅将上述操作从挑战阶段提前至初始化阶段, 敌手的视图没有发生任何变化, 因此有:

$$\text{Game}_0 \equiv \text{Game}_1$$

Game_2: 上一游戏已经做好激活 $\mathcal{G}$ 陷门的准备, 因此在 Game_2 中将预设的有损分支值由 0^{m_1} 替换为 y_1^* 完成激活:

- $(ek_{\text{abo}}, td_{\text{abo}}) \leftarrow G.\mathsf{Gen}(pp_{\text{abo}}, y_1^*)$

由 ABO-LTDF 的有损分支隐藏性质, 可以得到:

$$\text{Game}_1 \approx_c \text{Game}_2$$

Game_3: 使用 $\mathcal{G}$ 的陷门 td_{abo} 应答求逆询问, 当 $\mathcal{A}$ 发起询问 $y = (y_1, y_2)$ 时, $\mathcal{CH}$ 分情形应答如下:

- $y_1 = y_1^*$: 直接返回 $\perp$.
- $y_1 \neq y_1^*$: 计算 $x \leftarrow \mathcal{G}.\mathsf{TdInv}(td_{\text{abo}}, y_1, y_2)$, 如果 $y_1 = f_{ek_{\text{ltdf}}}(x)$ 则返回 x, 否则返回 $\perp$.

像的生成方式和 $\mathcal{G}$ 求逆算法的正确性和保证了 $\mathcal{O}_{\mathsf{inv}}$ 应答的正确性, 因此有:

$$\text{Game}_2 \equiv \text{Game}_3$$

Game_4: 将 y_1^* 的生成方式切换到有损模式

- $\mathcal{CH}$ 在初始化阶段计算$(ek_{\text{ltdf}}, \perp) \leftarrow \mathcal{F}.\mathsf{GenLossy}(pp_{\text{ltdf}})$

LTDF 的单射/有损模式的计算不可区分性保证了

$$\text{Game}_3 \approx_c \text{Game}_4$$

断言 4.5

任意 PPT 敌手 $\mathcal{A}$(即使拥有无穷计算能力) 在 Game_4 中的优势函数是忽略的. ♡

证明 在 Game_4 中, 函数 $f_{ek_{\text{ltdf}}}(\cdot)$ 有损且像集大小至多为 2^{τ_1}, 函数 $g_{ek_{\text{abo}}}(y_1^*, \cdot)$ 有损且像集大小至多为 2^{τ_2}. 因此 y_1^* 和 y_2^* 均在信息论意义下损失了原像 x^* 的信息, 在敌手 $\mathcal{A}$ 的视图中, x^* 的平均最小熵为 $\tilde{\mathsf{H}}_\infty(x^*|(y_1^*, y_2^*)) \geq \mathsf{H}_\infty(x^*) - \tau_1 - \tau_2 = n - \tau_1 - \tau_2 \geq \omega(\log \kappa)$. 从而对于任意敌手 $\mathcal{A}$ 均有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \mathsf{negl}(\kappa)$$

断言得证! □

综上, 定理得证! □

注记 4.9

上述构造的设计思想值得读者反复拆解, 体会其精妙之处. 上述 ATDF 构造在形式上与 Naor-Yung 的双钥加密有异曲同工之处: 分别使用 $f_{ek_{\text{ltdf}}}(\cdot)$ 和 $g_{ek_{\text{abo}}}(\cdot,\cdot)$ 两个函数计算原像的函数值作为像. 一个自然的想法是上述构造显得冗余, 是否仅用 ABO-LTDF 即可呢? 答案是否定的, 如果仅依赖 ABO-LTDF 构造 ATDF, 需要满足以下四点:

- 求值分支可由输入公开确定计算得出, 以确保 ATDF 是公开可计算函数.
- 像所对应的求值分支可由像中计算得出, 以确保 ATDF 的求逆算法可以基于 ABO-LTDF 的求逆算法设计.
- 在安全归约中势必需要将 ATDF 的单向性建立在 ABO-LTDF 的信息有损性上, 也即 y^* 是 x^* 在有损分支的求值.

上述三点潜在要求 ATDF 的像包含两个部分, 一部分是原像对应的分支值, 一部分是 ABO-LTDF 在该分支值下的像, 这使得在安全证明时存在如下两个障碍:

1. 分支值泄漏原像的多少信息难以确定

2. 敌手可以从挑战的像中计算出有损分支值, 从而可以发起关于有损分支的求逆询问, 而归约算法无法应答

通过上述的拆解分析, 便可看出 ATDF 设计的必然性. 引入 LTDF 并将分支值设定为原像的 LTDF 值有三重作用:

- LTDF 的陷门确保了 ATDF 构造存在功能完备的陷门.
- 可将分支值泄漏的关于原像信息量控制在指定范围.
- 分支值完全确定了像, 从而使得 ABO-LTDF 的陷门在归约证明中可用于模拟求逆谕言机 $\mathcal{O}_{\mathsf{inv}}$.

LTDF+ABO-LTDF $\Rightarrow$ ATDF 的设计思路有如二级运载火箭, 第一级运载火箭 (LTDF) 在完成推动后从单射切换到有损模式, 同时激活第二级运载火箭 (ABO-LTDF).

我们再展示如何基于 CP-TDF 构造 ATDF. 构造的难点是在归约证明中归约算法如何在不掌握全部 CP-TDF 实例陷门的情况下正确模拟 $\mathcal{O}_{\mathsf{inv}}$. 大体的设计思路和以上基于有损陷门函数构造 LTDF 相似, 通过多重求值引入冗余结构, 从而使得归约算法在掌握部分 CP-TDF 实例陷门时能够正确应答求逆询问.

- 设计像 y 形如 $(y_0, y_1, \ldots, y_n)$, 确保 y_0 能够惟一确定 $(y_1, \ldots, y_n)$
  - 令 y_0 是原像的 CP-TDF 函数值, 目的是确保 y_0 不会破坏最终 ATDF 函数的单向性
  - 令 $(y_1, \ldots, y_n)$ 是关于原像 x 的 $|y_0| = n$ 重冗余函数求值
- 嵌入 "全除一" 求逆结构
  - 对 y_0^* 进行比特分解: 归约算法使用 Dolev-Dwork-Naor(DDN) 类技术逐比特嵌入对应的陷门, 使得对于点 $y = (y_0, y_1, \ldots, y_n)$ 处的求逆询问:
    1. $y_0 = y_0^*$: 归约算法可根据 $\mathcal{O}_{\mathsf{inv}}$ 的定义直接拒绝, 返回 $\perp$
    2. $y_0 \neq y_0^*$: $\mathcal{R}$ 可至少寻找到一个可用陷门用于应答 $\mathcal{O}_{\mathsf{inv}}$.

构造 4.7 (基于 CP-TDF 的 ATDF 构造)

构造所需组件: 单射 CP-TDF $\mathcal{F} : X \to \{0,1\}^n$

构造 ATDF: $X \to \{0,1\}^{n(n+1)}$ 如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp \leftarrow \mathcal{F}.\mathsf{Setup}(1^\kappa)$, 输出 pp 作为公开参数.
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入
  1. 计算 $(\hat{ek}, \hat{td}) \leftarrow \mathcal{F}.\mathsf{KeyGen}(pp)$;
  2. 对于 $b \in \{0,1\}$ 和 $i \in [n]$, 计算 $(ek_{i,b}, td_{i,b}) \leftarrow \mathcal{F}.\mathsf{KeyGen}(pp)$;
  3. 输出 $(\hat{ek}, (ek_{i,0}, ek_{i,1}), \ldots, (ek_{n,0}, ek_{n,1}))$ 作为求值公钥, 输出 $(\hat{td}, (td_{i,0}, td_{i,1}), \ldots, (td_{n,0}, td_{n,1}))$ 作为求逆陷门.
- $\mathsf{Eval}(ek, x)$: 以求值公钥 $ek = \hat{ek}||(ek_{1,0}, ek_{1,1}) \ldots (ek_{n,0}, ek_{n,1})$ 和原像 x 为输入, 计算
  1. 计算 $y_0 \leftarrow f_{\hat{ek}}(x)$;
  2. 令 $b_i \leftarrow y_0[i]$, 对 $i \in [n]$ 计算 $y_i \leftarrow f_{ek_{i,b_i}}(x)$;
  3. 输出 $y = y_0||y_1|| \ldots ||y_n$.
- $\mathsf{TdInv}(td, y)$: 以陷门 $td = (\hat{td}, \{(td_{i,0}, td_{i,1})\}_{i\in[n]})$ 和像 $y = y_0||y_1|| \ldots ||y_n$ 为输入:
  1. 计算 $x_0 \leftarrow \mathcal{F}.\mathsf{TdInv}(\hat{td}, y_0)$;
  2. 令 $b_i \leftarrow y_0[i]$, 对所有 $i \in [n]$ 计算 $x_i \leftarrow \mathcal{F}.\mathsf{TdInv}(td_{i,b_i}, y_i)$;
  3. 检查 $x_i = x_0$ 是否对于 $i \in [n]$ 均成立, 若是则输出 x_0, 否则输出 $\perp$.

$\hat{ek}$	$ek_{1,0}$	$ek_{2,0}$	$ek_{3,0}$
$\hat{td}$	$td_{1,0}$	$td_{2,0}$	$td_{3,0}$
	$ek_{1,1}$	$ek_{2,1}$	$ek_{3,1}$
	$td_{1,1}$	$td_{2,1}$	$td_{3,1}$

图 4.3: $n = 3$ 时的求值公钥和求逆陷门图示

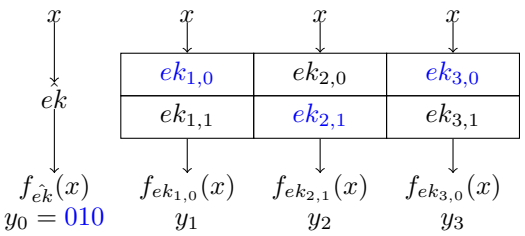


图 4.4: $n = 3, y = 010$ 时的求值图示

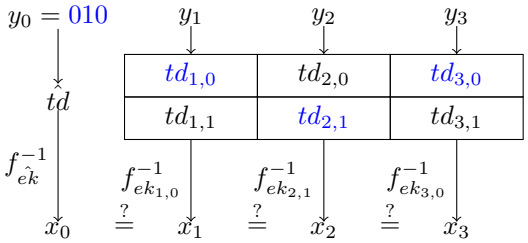


图 4.5: $n = 3, y = 010$ 时的求逆图示

上述 ATDF 构造的正确性显然. 构造中, 函数的像 $y = y_0||y_1||\ldots||y_n$ 是对原像的 $n+1$ 重求值, 其中 y_0 确定了使用哪些求值公钥 $ek_{i,b}$ 计算 y_i, 因此当底层的 CP-TDF 是单射函数时, y_0 可惟一确定 $y_1, \ldots, y_n$. 下面的定理就是利用上述结构特性模拟求逆谕言机 $\mathcal{O}_{\mathsf{inv}}$.

定理 4.9

如果 $\mathcal{F}$ 是一族相对于 $\mathcal{U}_{n+1}$ 安全的 CP-TDF, 那么上述构造是一族自适应单向陷门函数. ♡

证明 使用反证法通过单一游戏完成归约证明. 假设存在 PPT 的敌手 $\mathcal{A}$ 能以不可忽略的优势打破 ATDF 的自适应单向性, 那么可以黑盒调用 $\mathcal{A}$ 的能力构造 PPT 的 $\mathcal{B}$ 打破 CP-TDF 相对于 $\mathcal{U}_{n+1}$ 的单向性. $\mathcal{B}$ 的 CP-TDF 挑战是公开参数 pp、求值公钥 $(ek_0, ek_1, \ldots, ek_n)$ 和像 $y^* = (y_0^*, y_1^*, \ldots, y_n^*)$, 其中 $y_i^* \leftarrow f_{ek_i}(x^*)$, $x^* \xleftarrow{R} X$. $\mathcal{B}$ 并不知晓 x^*, 其攻击目标是求解 x^*.

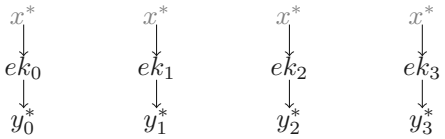


图 4.6: $n = 3$ 时 $\mathcal{B}$ 的 CP-TDF 挑战实例

令 b_i^* 是 y_0^* 的第 i 比特, $\mathcal{B}$(扮演挑战者) 与 $\mathcal{A}$ 在 ATDF 的自适应单向性游戏中交互如下:

- 初始化: $\mathcal{B}$ 将 CP-TDF 的 pp 设为 ATDF 的公开参数, 设定 $\hat{ek} := ek_0$, 对 $i \in [n]$ 设定 $ek_{i,b_i^*} := ek_i$, 计算 $(ek_{i,1-b_i^*}, td_{i,1-b_i^*}) \leftarrow \mathcal{F}.\mathsf{KeyGen}(\kappa)$.
- 挑战阶段: $\mathcal{B}$ 发送 $(y_0^*, y_1^*, \ldots, y_n^*)$ 给 $\mathcal{A}$ 作为挑战.
- 求逆询问: $\mathcal{A}$ 向 $\mathcal{B}$ 发起求逆询问 $y = (y_0, y_1, \ldots, y_n)$, $\mathcal{B}$ 分情况应答如下:
  1. $y_0 = y_0^*$: 直接返回 $\perp$, 应答的正确性由以下两种细分情况保证:
     - 对于所有的 $i \in [n]$ 均有 $y_i = y_i^*$: 询问为禁询点, 因此根据 $\mathcal{O}_{\mathsf{inv}}$ 的定义需返回 $\perp$.
     - 对于某个 $i \in [n]$ 使得 $y_i \neq y_i^*$: $\mathcal{F}$ 的单射性质和像的生成方式保证了像的首项 y_0 确定了其余 n 项 $y_1, \ldots, y_n$.
  2. $y_0 \neq y_0^*$: 必然存在 $\exists j \in [n]$ s.t. $b_j \neq b_j^*$ 且 $y_j = f_{ek_{j,b_j}}(x)$, 其中 x 是未知原像. 此时, $\mathcal{B}$ 拥有关于 ek_{j,b_j} 的求逆陷门 td_{j,b_j}, $\mathcal{B}$ 可计算 $x \leftarrow f^{-1}_{ek_{j,b_j}}(y_j)$
     - 如果 $y_0 = f_{ek_0}$ 且 $y_i = f_{ek_{i,b_i}}(x)$ 对其余所有 $i \neq j$ 也均成立, 那么返回 x, 否则返回 $\perp$.
- 求解: $\mathcal{A}$ 输出 x 作为 ATDF 的挑战应答, $\mathcal{B}$ 将 x 转发给 CP-TDF 的挑战者.

容易验证, $\mathcal{B}$ 的优势与 $\mathcal{A}$ 的优势相同. 定理得证! □

注记 4.10 (基于 CP-TDF 的 ATDF 构造优化)

以上 ATDF 构造的像 $(y_0, y_1, \ldots, y_n)$ 包含了对原像的 $(n+1)$ 重 CP-TDF 求值:

- y_0 构造中起到的作用求值的公钥选择向量, 在归约证明中起到的作用是“全除一”求逆陷门的激活扳机 (trigger), 当 $y_0 \neq y_0^*$ 时即可激活求逆陷门.

y_0 的编码长度决定了像的冗余重数. 能否缩减 $|y_0|$ 以提高效率呢? 答案是肯定的, 可以使用密码组件进行定义域扩张 (domain extension) 的通用技术, 使用 y_0 的抗碰撞哈希值代替 y_0. 在上述构造中, 我们贴合 ATDF 的安全定义进行更为精细的处理, 使用定向抗碰撞哈希函数 (target collision resistant hash function,

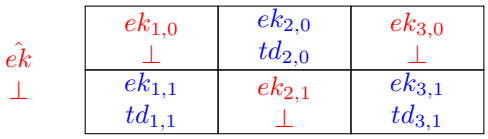


图 4.7: $y_0^* = 010$ 时生成求值公钥和求逆陷门的过程图示

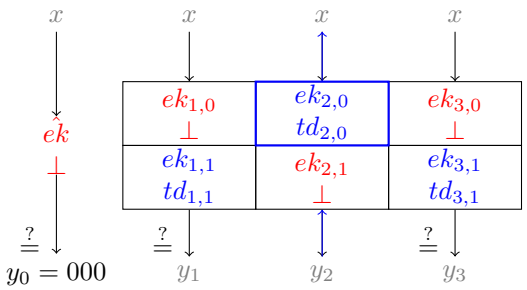


图 4.8: $y_0 = 000$ 时的求逆过程图示

TCRHF) 代替 CRHF. 具体的, 令 $\mathsf{TCR} : \{0,1\}^n \to \{0,1\}^m$, 使用 $\mathsf{TCR}(y_0)$ 代替 y_0 作为作为公钥选择向量和陷门激活扳机. 从而利用 TCR 压缩的性质将像的重数从 $1+n$ 缩减到 $1+m$. 安全论证仍然成立, 这是因为 TCR 的抗碰撞性质保证了在计算意义下:

$$y_0 \neq y_0^* \iff \mathsf{TCR}(y_0) \neq \mathsf{TCR}(y_0^*)$$

类似的优化技术同样可以用于 LTDF+ABO-LTDF $\Rightarrow$ ATDF 的构造中: 可以使用 y_0 的 TCR 哈希值代替 y_0 作为分支值. 这样处理的好处是增加分支集合选择的灵活性.

笔记 LTDF+ABO-LTDF $\Rightarrow$ ATDF 与 CP-TDF $\Rightarrow$ ATDF 的构造分别与 Naor-Yung 范式 [9] 和 Dolev-Dwork-Naor 范式 [23] 在思想上极为相似, 总体思路都是通过冗余的结构来保证求逆谕言机的完美模拟.

自适应单向陷门关系

将 ATDF 中的确定性函数泛化为可公开高效验证的二元关系可得到自适应单向陷门关系 (ATDR, adaptive trapdoor relation).

- 确定性函数 $\rightsquigarrow$ 概率关系
- 可高效计算 $\rightsquigarrow$ 可高效采样

定义 4.8 (单向陷门关系 (TDR))

一族单向陷门关系包含以下算法:

- $\mathsf{Setup}(\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 $pp = (X, Y, EK, TD, \mathsf{R})$, 其中 $\mathsf{R} = \{\mathsf{R}_{ek} : X \times Y\}_{ek \in EK}$ 是定义在 $X \times Y$ 上由 ek 索引的一族二元单向关系.
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入, 输出公钥 ek 和陷门 td.
- $\mathsf{Sample}(ek)$: 输出二元关系的一个随机采样 $(x, y) \xleftarrow{\mathsf{R}} \mathsf{R}_{ek}$.
- $\mathsf{TdInv}(td, y)$: 以 td 和 $y \in Y$ 为输入, 输出 $x \in X \cup \perp$.

正确性: $\forall (ek, td) \leftarrow \mathsf{KeyGen}(pp), \forall (x, y) \leftarrow \mathsf{Sample}(ek)$, 总有 $(\mathsf{TdInv}(td, y), y) \in \mathsf{R}_{ek}$.

我们可以将函数的单射性质平行推广至二元关系的场景下: 如果 $\forall (x_1, y_1), (x_2, y_2) \in \mathsf{R}_{ek}$ 均有 $x_1 \neq x_2 \Rightarrow y_1 \neq y_2$, 即 y 惟一确定了 x, 那么则称二元关系满足单射性.

笔记 Sample 是概率算法, 因此当 $y_1 \neq y_2$ 时, 存在 $x_1 = x_2$ 的可能.

定义 4.9 (自适应单向性)

令 R 是一族二元关系, 定义敌手 $\mathcal{A}$ 的优势如下:

$$\Pr\left[(x', y^*) \in \mathsf{R}_{ek} : \begin{array}{l} pp \leftarrow \mathsf{Setup}(\kappa); \\ (ek, td) \leftarrow \mathsf{KeyGen}(pp); \\ (x^*, y^*) \leftarrow \mathsf{Sample}(ek); \\ x' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{inv}}}(ek, y^*); \end{array}\right]$$

其中 $\mathcal{O}_{\mathsf{inv}}$ 是求逆谕言机, $\forall x \neq x^*, \mathcal{O}_{\mathsf{inv}}(y) = \mathsf{TdInv}(td, y)$. 如果任意 PPT 敌手 $\mathcal{A}$ 在上述安全试验中的优势均为 $\mathsf{negl}(\kappa)$, 那么则称 R 是自适应单向的.

♣

ATDR 是 ATDF 的弱化, 弱化允许我们可以给出更加高效灵活的设计, 同时不严重降低可用性. 在给出 ATDR 的构造之前, 我们首先回顾基于 CP-TDF 的 ATDF 构造. 构造的关键之处是将像 y 设计为 y_0 和 $(y_1, \ldots, y_n)$ 两部分, 其中 y_0 设定为 $f_{ek}(x)$, 通过单射性完美绑定了 $(y_1, \ldots, y_n)$, 同时在归约证明中起到了 "全除一" 陷门触发器的作用: 当目标不再是构造确定性单向函数而是概率二元关系时, 我们有着更加灵活的选择: 使用一次性签名 (OTS, one-time signature) 的验证公钥作为 $(y_1, \ldots, y_n)$ 的求值选择器和求逆陷门触发器.

构造 4.8 (基于 CP-TDF 和 OTS 的 ATDR 构造)

构造组件: 单射 CP-TDF $\mathcal{F} : X \to Y$ 和 strong OTS (令 $|vk| = \{0,1\}^n$, 签名空间为 Σ);

构造目标: ATDR $X \to VK \times Y^n \times \Sigma$

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp_{\text{cptdf}} \leftarrow \mathcal{F}.\mathsf{Setup}(1^\kappa)$, $pp_{\text{ots}} \leftarrow \mathsf{OTS.Setup}(1^\kappa)$, 输出 $pp = (pp_{\text{cptdf}}, pp_{\text{ots}})$.
- $\mathsf{KeyGen}(pp)$: 以 $pp = (pp_{\text{cptdf}}, pp_{\text{ots}})$ 为输入, 对 $b \in \{0,1\}$ 和 $i \in [n]$ 运行 $(ek_{i,b}, td_{i,b}) \leftarrow \mathcal{F}.\mathsf{KeyGen}(pp_{\text{cptdf}})$ 输出 $ek = ((ek_{i,0}, ek_{i,1}), \ldots, (ek_{n,0}, ek_{n,1}))$, $td = ((td_{i,0}, td_{i,1}), \ldots, (td_{n,0}, td_{n,1}))$.
- $\mathsf{Sample}(ek)$: 以 $ek = (ek_{1,0}, ek_{1,1}) \ldots (ek_{n,0}, ek_{n,1})$ 为输入, 采样如下:
  1. 生成 $(vk, sk) \leftarrow \mathsf{OTS.KeyGen}(pp_{\text{ots}})$;
  2. 随机选择 $x \in X$, 对 $i \in [n]$ 计算 $y_i \leftarrow f_{ek_{i,b_i}}(x)$, 其中 $b_i \leftarrow vk[i]$;
  3. 计算 $\sigma \leftarrow \mathsf{OTS.Sign}(sk, y_1 || \ldots || y_n)$;

  输出 $y = (vk, y_1 || \ldots || y_n, \sigma)$.
- $\mathsf{TdInv}(td, y)$: 以 $td = (\{(td_{i,0}, td_{i,1})\}_{i\in[n]})$ 和 $y = (vk, y_1 || \ldots || y_n, \sigma)$ 为输入, 求逆如下:
  1. 检查 $\mathsf{OTS.Verify}(vk, y_1 || \ldots || y_n, \sigma) \stackrel{?}{=} 1$, 如果签名无效则返回 $\perp$;
  2. 对所有 $i \in [n]$ 计算 $x_i \leftarrow \mathcal{F}.\mathsf{TdInv}(td_{i,b_i}, y_i)$, 其中 $b_i = vk[i]$.
  3. 如果对所有 $i \in [n]$ 均有 $x_i = x_1$ 则返回 x_1, 否则返回 $\perp$.

♣

$|vk| = 3$

$ek_{1,0}$ $td_{1,0}$	$ek_{2,0}$ $td_{2,0}$	$ek_{3,0}$ $td_{3,0}$
$ek_{1,1}$ $td_{1,1}$	$ek_{2,1}$ $td_{2,1}$	$ek_{3,1}$ $td_{3,1}$

图 4.9: $|vk| = 3$ 时的求值公钥和求逆陷门生成图示

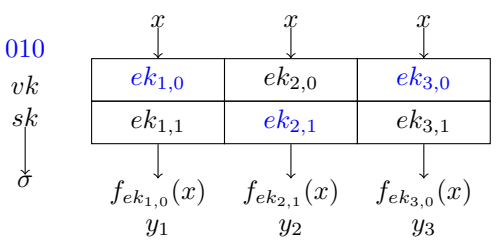


图 4.10: $vk = 010$ 时的采样过程

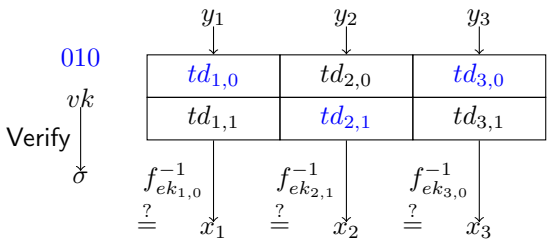


图 4.11: $vk = 010$ 时的求逆过程

构造的正确性显然, 构造的以下三个特性使得归约算法能够成功模拟 $\mathcal{O}_{\mathsf{inv}}$.

- R_{ek} 是单射的并且 $y_1||\ldots||y_n$ 是对原像 x 的 n 重冗余求值.
- vk 是求值公钥的选择比特向量.
- 利用 OTS 的 sEUF-CMA 安全性, vk 在计算意义下绑定了 $(y_1,\ldots,y_n)$.

定理 4.10

如果 OTS 是 sEUF-CMA 安全的, 并且 $\mathcal{F}$ 是 $\mathcal{U}_n$ 相关积单向的, 那么构造 4.8中的二元关系满足自适应单向性.

♡

证明 证明通过以下游戏序列完成.

Game_0: 对应真实的 ATDR 自适应单向性安全试验. 令 $y^*=(vk^*,y_1^*||\ldots||y_n^*,\sigma^*)$ 是挑战的像.

Game_1: 与 Game_0 相同, 唯一的区别是挑战者对于求逆询问 $y=(vk^*,y_1||\ldots||y_n,\sigma)$ 直接返回 $\perp$. 应答的合理性分情况解释如下:

1. $(y_1||\ldots||y_v,\sigma)=(y_1^*||\ldots||y_v^*,\sigma^*)$: 禁询点
2. $(y_1||\ldots||y_v,\sigma)\neq(y_1^*||\ldots||y_v^*,\sigma^*)$: 构成 OTS 的存在性伪造

记敌手发起第二种类型求逆询问的事件为 F, 那么利用 Difference Lemma 可以证明 $|\Pr[S_1]-\Pr[S_0]|\leq\Pr[F]$, 而基于 OTS 的 sEUF-CMA 安全性, 可以推出 $\Pr[F]\leq\mathsf{negl}(\kappa)$, 从而 $|\Pr[S_1]-\Pr[S_0]|\leq\mathsf{negl}(\kappa)$.

断言 4.6

如果 $\mathcal{F}$ 是 $\mathcal{U}_t$ 相关积安全的, 那么对于任意的 PPT 敌手均有 $\Pr[S_1]=\mathsf{negl}(\kappa)$.

♡

证明 论证通过单一归约完成. 假设存在 PPT 的敌手 $\mathcal{A}$ 在 Game_1 中的优势不可忽略, 那么尝试构造 PPT 算法 $\mathcal{B}$, 通过黑盒调用 $\mathcal{A}$ 的能力打破 CP-TDF 相对 $\mathcal{U}_n$ 的相关积单向性. $\mathcal{B}$ 的 CP-TDF 挑战是公开参数 pp_{cptdf}, 求值公钥 $(ek_1,\ldots,ek_n)$ 和像 $(y_1^*,\ldots,y_n^*)$, 其中 $y_i^*\leftarrow f_{ek_i}(x^*)$, $x^*\xleftarrow{\mathsf{R}}X$. $\mathcal{B}$ 并不知晓 x^*, 其攻击目标是求解 x^*.

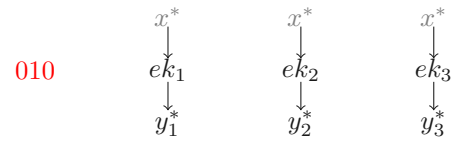


图 4.12: $n=3$ 时 $\mathcal{B}$ 的 CP-TDF 挑战实例

$\mathcal{B}$(扮演挑战者) 与 $\mathcal{A}$ 在 Game_1 中交互如下:

- 初始化: $\mathcal{B}$ 运行 $pp_{\mathsf{ots}}\leftarrow\mathsf{OTS.Setup}(1^\kappa)$, 生成 $(vk^*,sk^*)\leftarrow\mathsf{OTS.KeyGen}(pp_{\mathsf{ots}})$. 令 b_i^* 是 vk^* 的第 i 比特, $\mathcal{B}$ 进行如下操作:
  1. 对 $i\in[n]$ 设定 $ek_{i,b_i^*}:=ek_i$.
  2. 对 $i\in[v]$ 计算 $(ek_{i,1-b_i^*},td_{i,1-b_i^*})\leftarrow\mathcal{F}.\mathsf{KeyGen}(pp_{\mathsf{cptdf}})$.

  $\mathcal{B}$ 发送 $pp=(pp_{\mathsf{cptdf}},pp_{\mathsf{ots}})$ 和 $ek=(ek_{1,0},ek_{1,1},\ldots,ek_{n,0},ek_{n,1})$ 给 $\mathcal{A}$.

vk^*	$ek_{1,0}$	$ek_{2,0}$	$ek_{3,0}$
	$\perp$	$td_{2,0}$	$\perp$
sk^*	$ek_{1,1}$	$ek_{2,1}$	$ek_{3,1}$
	$td_{1,1}$	$\perp$	$td_{3,1}$

图 4.13: $|vk|=010$ 时归约算法设定求值公钥和求逆陷门的过程图示

- 挑战: $\mathcal{B}$ 计算 $\sigma^*\leftarrow\mathsf{OTS.Sign}(sk^*,(y_1^*,\cdots,y_n^*))$, 发送 $(vk^*,y_1^*,\cdots,y_n^*,\sigma^*)$ 给 $\mathcal{A}$ 作为挑战.
- 求逆询问: 对于求逆询问 $y=(vk,y_1||\ldots||y_v,\sigma)$, $\mathcal{B}$ 应答如下:
  1. $vk=vk^*$: 直接返回 $\perp$.

2. $vk \neq vk^*$: 必然存在 $\exists j \in [n]$ s.t. $b_j \neq b_j^*$ 且 $y_j = f_{ek_j,b_j}(x)$, 其中 x 是未知原像. 此时, $\mathcal{B}$ 拥有关于 ek_{j,b_j} 的求逆陷门 td_{j,b_j}, $\mathcal{B}$ 可计算 $x \leftarrow f^{-1}_{ek_j,b_j}(y_j)$
   - 如果 $y_i = f_{ek_{i,b_i}}(x)$ 对所有的 $i \neq j$ 也均成立, 那么返回 x, 否则返回 $\perp$.

由 $\mathcal{F}$ 的单射性可知, $\mathcal{B}$ 完美的模拟了 Game_1 中的 $\mathcal{O}_{\text{inv}}$ 应答.

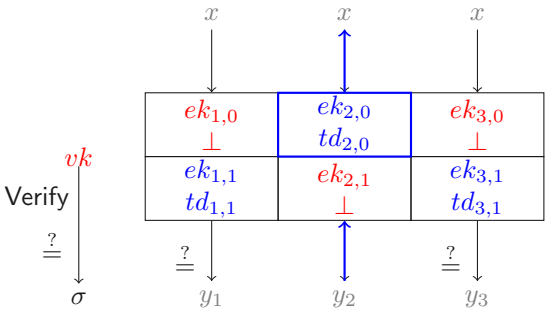


图 4.14: $vk = 010$ 时归约算法求逆过程图示

- 求解: $\mathcal{A}$ 输出 x 作为 Game_1 中 ATDF 的挑战应答, $\mathcal{B}$ 将 x 转发给 CP-TDF 的挑战者.

容易验证, $\mathcal{B}$ 的优势与 $\mathcal{A}$ 的优势相同. 断言得证. □

综上, 定理得证! □

小结

本节中各类单向函数之间的蕴含关系如图 4.15所示. Rosen 和 Segev [32] 证明了 LTDF 与 CP-TDF 之间存在黑盒分离, Kiltz、Mohassel 和 O'Neill [33] 证明了 CP-TDF 与 LTDF 之间也存在黑盒分离. 很长一段时期, ATDF 和 ATDR 是黑盒意义下构造 CCA-KEM 所需的最弱单向陷门函数类组件. 一个重要的公开问题是: 不带有任何增强安全属性的 TDF 是否能蕴涵 CCA 安全的 PKE 方案? Hohenberger 等 [34] 在 2020 年美密会的最佳论文中给出了肯定的答案, 即单射 TDF 即可蕴涵 CCA 安全的 PKE 方案. 这里技术细节不再展开, 感兴趣的读者请查阅论文.

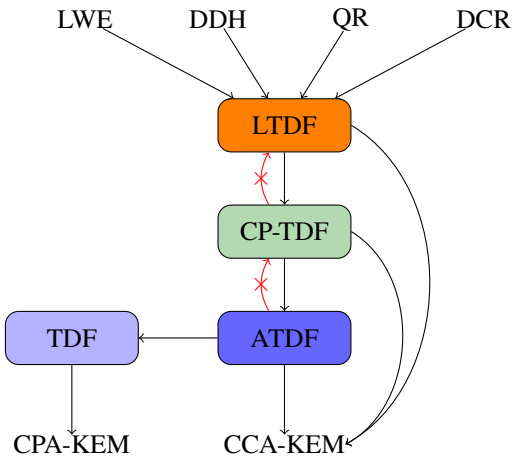


图 4.15: 各类单向函数之间的蕴含关系

4.2 哈希证明系统类

一阴一阳谓之道, 继之者善也, 成之者性也.

—《易经 · 系辞上》

1998 年, Cramer 和 Shoup [109] 基于判定性 Diffie-Hellman 问题构造出首个标准模型下高效的 PKE 方案, 称为 CS98-PKE. 2002 年, Cramer 和 Shoup [27] 再度合作, 提出了哈希证明系统 (HPS, hash proof system) 的概念, 给出了标准模型下构造 CCA 安全 PKE 的全新范式, 完美的阐释了 CS98-PKE 的设计原理. 以下首先介绍 HPS 的定义和相关性质.

定义 4.10 (哈希证明系统 (HPS))

HPS 包含以下 4 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$, 其中 $\mathsf{H} : SK \times X \to \Pi$ 是由私钥集合 SK 索引的一族带密钥哈希函数 (keyed hash function), L 是定义在 X 上的 $\mathcal{NP}$ 语言, W 是对应的证据集合, α 是从私钥集合 SK 到公钥集合 PK 的投射函数 (projection).
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入, 随机采样 $sk \xleftarrow{\mathrm{R}} SK$, 计算 $pk \leftarrow \alpha(sk)$, 输出 (pk, sk).
- $\mathsf{PrivEval}(sk, x)$: 以私钥 sk 和 $x \in X$ 为输入, 输出 $\pi = \mathsf{H}_{sk}(x)$.
- $\mathsf{PubEval}(pk, x, w)$: 以公钥 pk、$x \in L$ 以及相应的 w 为输入, 输出 $\pi = \mathsf{H}_{sk}(x)$, 其中 $\alpha(sk) = pk$. ♣

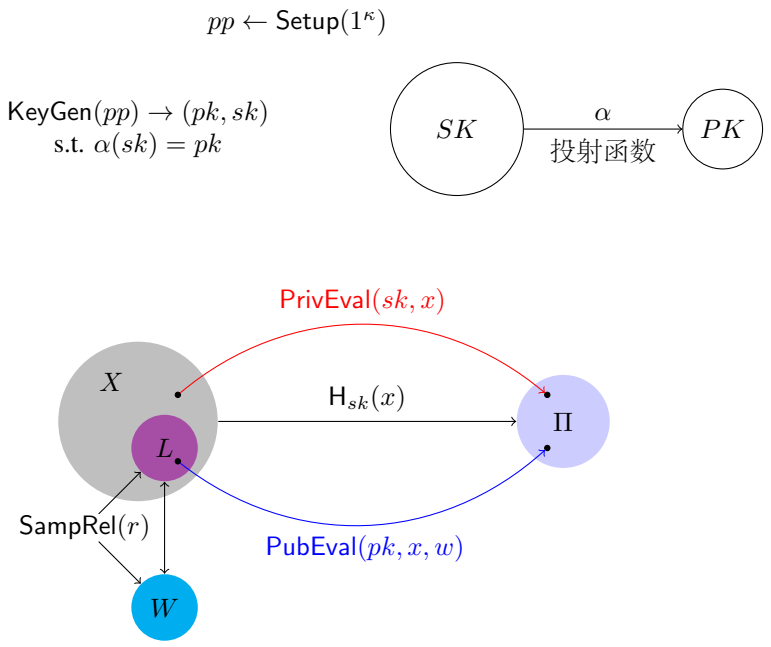


图 4.16: HPS 示意图

HPS 的定义围绕 $L \subset X$ 展开, 引入了 KeyGen, PrivEval 和 PubEval 这三个核心算法. 以下性质刻画了哈希函数在输入 $x \in L$ 上的行为, 用于保证上层密码方案的功能性.

投射性 (projective): $\forall x \in L$, 函数值 $\mathsf{H}_{sk}(x)$ 由 x 和私钥的投射 $pk \leftarrow \alpha(sk)$ 完全确定.

以下性质由弱到强刻画了哈希函数在输入 $x \in X \backslash L$ 上的行为, 用于保证上层密码方案的安全性.

平滑性 (smooth): $\mathsf{H}_{sk}(\cdot)$ 在输入 $x \xleftarrow{\mathrm{R}} X \backslash L$ 时的输出与 Π 上的均匀分布统计接近, 即:

$$(pk, \mathsf{H}_{sk}(x)) \approx_s (pk, \pi)$$

其中 $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$, $\pi \xleftarrow{\mathrm{R}} \Pi$.

1-一致性 (universal$_1$): $\mathsf{H}_{sk}(\cdot)$ 在任意输入的输出与 Π 上的均匀分布统计接近, 即 $\forall x \in X \backslash L$, 有:

$$(pk, \mathsf{H}_{sk}(x)) \approx_s (pk, \pi)$$

其中 $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$, $\pi \xleftarrow{R} \Pi$.

2-一致性 (universal$_2$): 在给定某点 $x^* \in X \backslash L$ 哈希函数值的情形下, $\mathsf{H}_{sk}(\cdot)$ 在任意输入的输出仍与 Π 上的均匀分布统计接近, 即 $\forall x, x^* \in X \backslash L$ 且 $x \neq x^*$, 有:

$$(pk, \mathsf{H}_{sk}(x^*), \mathsf{H}_{sk}(x)) \approx_s (pk, \mathsf{H}_{sk}(x^*), \pi)$$

其中 $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$, $\pi \xleftarrow{R} \Pi$.

笔记 以上三条性质由弱到强. smooth 性质同时建立在 $sk \xleftarrow{R} SK$ 和 $x \xleftarrow{R} X \backslash L$ 两根随机带上, universal$_1$ 性质仅建立在 $sk \xleftarrow{R} SK$ 一根随机带上, 而 universal$_2$ 性质则可解读为要求 universal$_1$ 性质在随机带 $sk \xleftarrow{R} SK$ 有偏时 (将 $\mathsf{H}_{sk}(x^*)$ 理解为关于 sk 的泄漏) 仍然成立. 特别注意, 三条性质均刻画的是输入在语言外时哈希函数的行为.

4.2.1 哈希证明系统的起源释疑

很多读者在阅读 HPS 相关的文献时, 都会对这个范式的命名和引入动机感到疑惑. 事实上, HPS 是一类指定验证者的非交互式零知识证明系统 (designated verifier NIZK), 引入的动机来自以下的思考: Naor-Yung 双重加密范式使用标准的 NIZK 来证明密文的合法性 (well-formedness), 然而密文的合法性并非一定是可公开验证的 (public verifiable), 解密私钥 sk 的持有者可验证即可. 指定可验证弱于公开可验证, 因此 DV-NIZK 的效率通常高于 NIZK. 想必 Cramer 和 Shoup 正是基于以上的思考, 引入了 HPS, 目的是在标准模型下构造高效的 IND-CCA 安全的 PKE.

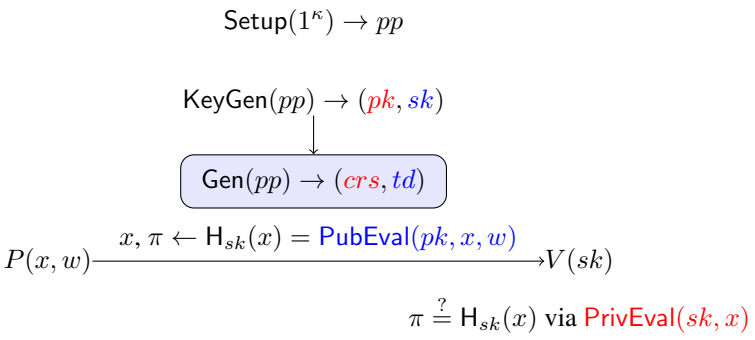


图 4.17: 从 DV-NIZK 的视角解析 HPS

笔记 图 4.17解释了 HPS 的命名渊源, 其本质上是指定验证者零知识证明, 证明的形式是实例的哈希值, 故名哈希证明系统.

- DV-NIZK 的完备性由 $\mathsf{H}_{sk}(\cdot)$ 的投射性保证:

$$\forall x \in L, \mathsf{H}_{sk}(x) = \mathsf{PubEval}(pk, x, w)$$

- DV-NIZK 的合理性由 universal$_1$ 性质保证 $\forall x \notin L, \mathsf{H}_{sk}(x)$ 随机分布, 即使拥有无限计算能力的证明者 P^* 也无法预测, 因此通过验证的概率可忽略. universal$_2$ 性质则保证了更强的合理性, 即敌手在看到一个 No 实例的有效证明后, 也无法为一个新的 No 实例生成有效证明.
- DV-NIZK 的零知识性是显然且平凡的: 指定验证者拥有私钥, 因此可以对任意的 $x \in L$(甚至对于 $x \in X \backslash L$) 生成正确的证明.

此外, 证明系统是有效的, 即证明者在拥有证据时可以高效计算出实例的证明, 这对于基于 HPS 密码方案的功能性至关重要.

4.2.2 哈希证明系统的实例化

以下通过介绍 L_{DDH} 语言的 HPS 实例化协议建立对 HPS 的直观认识. 首先运行 $\mathsf{GenGroup}(1^\kappa) \to (\mathbb{G}, q, g)$, 其中 $\mathbb{G}$ 是阶为素数 p 的群, g 是生成元; 再随机选取 $\mathbb{G}$ 中的两个生成元 g_1, g_2. 令 $pp = (\mathbb{G}, q, g_1, g_2)$ 是公开参数, 定义由 pp 索引的 $\mathcal{NP}$ 语言如下:

$$L_{\text{DDH}} = \{(x_1, x_2) \in X : \exists w \in W \text{ s.t. } x_1 = g_1^w \wedge x_2 = g_2^w\}$$

其中 $X = \mathbb{G} \times \mathbb{G}, W = \mathbb{Z}_q$.

容易验证, 语言中的元素是 DH 对, 语言外的元素是非 DH 对, $(x_1, x_2) \xleftarrow{\text{R}} L_{\text{DDH}}$. DDH 假设蕴含 $L \subset X$ 上的 SMP 困难问题成立, 即:

- $U_L \approx_c U_X$: 随机 DH 对与 X 中的随机二元组计算不可区分
- 由于 $|L|/|X| = 1/q = \mathsf{negl}(\kappa)$, L 在 X 中稀疏, 所以可以进一步得到 $U_L \approx_c U_{X \setminus L}$: 随机 DH 对与随机非 DH 对计算不可区分

构造 4.9 (L_{DDH} 语言的 HPS 构造)

L_{DDH} 的 HPS 构造如下, 如图 4.18所示:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 $pp = (\mathbb{G}, q, g_1, g_2)$. pp 还包括了对 $SK = \mathbb{Z}_q \times \mathbb{Z}_q$, $PK = \mathbb{G}$, L_{DDH}, $X = \mathbb{G} \times \mathbb{G}$ 和 $W = \mathbb{Z}_q$ 的描述.
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入, 随机采样 $sk \xleftarrow{\text{R}} \mathbb{Z}_q^2$, 计算 $pk \leftarrow \alpha(sk) = g_1^{sk_1} g_2^{sk_2}$, 输出 (pk, sk).
- $\mathsf{PrivEval}(sk, x)$: 以私钥 sk 和 $x \in X$ 为输入, 输出 $\pi = \mathsf{H}_{sk}(x) = x_1^{sk_1} x_2^{sk_2}$.
- $\mathsf{PubEval}(pk, x, w)$: 以公钥 pk、$x \in L_{\text{DDH}}$ 以及相应的 w 为输入, 输出 $\pi = pk^w$, 其中 $\alpha(sk) = pk$. 以下等式说明了公开求值算法的正确性:

$$pk^w = (g_1^{sk_1} g_2^{sk_2})^w = x_1^{sk_1} x_2^{sk_2} = \mathsf{H}_{sk}(x)$$

$$pp = (\mathbb{G}, q, g_1, g_2) \leftarrow \mathsf{Setup}(1^\kappa)$$

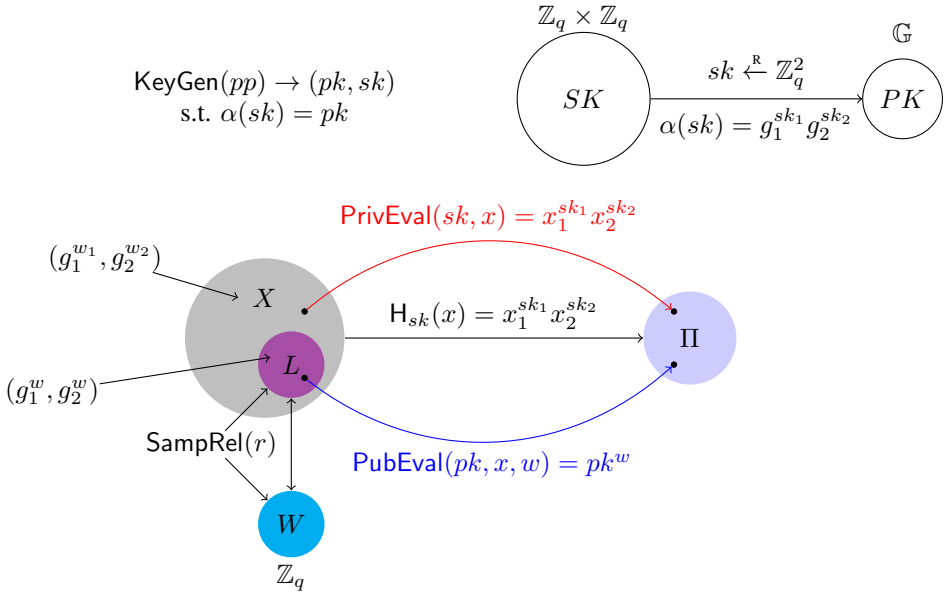


图 4.18: L_{DDH} 的 HPS

引理 4.2

以上关于 L_{DDH} 的 HPS 满足 universal$_1$ 性质.

证明 证明的目标是

$$\forall x \in X \backslash L, (pk, \mathsf{H}_{sk}(x)) \approx_s (pk, \pi)$$

其中 $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$, $\pi \xleftarrow{R} \Pi$.

首先固定 $x = (x_1 = g_1^{w_1}, x_2 = g_2^{w_2}) \in X \backslash L$, 其中 $w_1 \neq w_2$. 将左式表示为关于 sk 函数的形式:

$$(pk, \mathsf{H}_{sk}(x)) = f_{g_1,g_2,x_1,x_2}(sk_1, sk_2) := (g_1^{sk_1} g_2^{sk_2}, x_1^{sk_1} x_2^{sk_2})$$

用矩阵的形式描述函数作用过程:

$$\begin{pmatrix} g_1 & g_2 \\ g_1^{w_1} & g_2^{w_2} \end{pmatrix} \begin{pmatrix} sk_1 \\ sk_2 \end{pmatrix} = \begin{pmatrix} pk \\ \mathsf{H}_{sk}(x) \end{pmatrix}$$

令 $g_2 = g_1^{\beta}$, 其中 $\beta \in \mathbb{Z}_q^*$, 将最左边矩阵进行等价变形:

$$\begin{pmatrix} g_1 & g_2 \\ g_1^{w_1} & g_2^{w_2} \end{pmatrix} = \begin{pmatrix} g_1 & g_1^{\beta} \\ g_1^{w_1} & g_1^{w_2\beta} \end{pmatrix} = g_1 \underbrace{\begin{pmatrix} 1 & \beta \\ w_1 & w_2\beta \end{pmatrix}}_{M}$$

$\det(M) = \beta(w_1 - w_2) \Rightarrow M$ 满秩 $\Rightarrow f$ 单射. 又由于函数的定义域和值域大小相等, 最终得出:

$$\underbrace{\begin{pmatrix} g_1 & g_2 \\ g_1^{w_1} & g_2^{w_2} \end{pmatrix}}_{\text{full rank } 2\times 2} \underbrace{\begin{pmatrix} sk_1 \\ sk_2 \end{pmatrix}}_{\text{uniform over } \mathbb{Z}_q^2} = \underbrace{\begin{pmatrix} pk \\ \mathsf{H}_{sk}(x) \end{pmatrix}}_{\text{uniform over } \mathbb{G}^2}$$

从而 universal_1 性质得证! □

注记 4.11

HPS 并不一定要求 $L \subseteq X$ 之上一定存在 SMP 问题, 但只有当 $L \subseteq X$ 之上存在 SMP 问题时, 相应的 HPS 有密码学意义. 这是因为 HPS 中所有关于哈希函数的性质均是针对输入在语言外时定义的, 只有当 SMP 问题存在时, 才可以间接刻画出哈希函数在输入在语言内时的行为.

HPS 存在两个局限:

- 证明只支持私密验证, 不满足公开验证性
- 证明的表达能力有限, 目前仅能对证明群中的子群成员归属问题, 尚未知能否延伸到任意的 NP 语言.

在很多具体的零知识证明应用场合, 公开验证性和强大的表达能力均不是必须, 因此用标准的零知识证明系统有大材小用之嫌, 哈希证明系统可以做的更快更好, 其中效率的优势恰恰源自局限. 以下展示如何基于 HPS 设计 IND-CPA 和 IND-CCA 的 KEM 方案.

4.2.3 基于哈希证明系统的 KEM 构造

作为暖场应用, 我们首先介绍如何基于 HPS 构造 IND-CPA 安全的 KEM. 设计的思路如下:

- 发送方扮演 HPS 中的证明者, 选择 L 中的随机实例 x 作为密文 c, 利用公钥 pk 和相应的证据 w 计算其哈希证明 π 作为会话密钥 k.
- 接收方扮演 HPS 中的验证者, 使用私钥 sk 计算 x 的哈希证明以恢复会话密钥 k.

构造 4.10 (基于 HPS 的 CPA 安全的 KEM 构造)

从平滑 HPS 出发, 构造 CPA 安全的 KEM 如下:

- $\mathsf{Setup}(\kappa)$: 运行 $pp \leftarrow \mathsf{HPS.Setup}(1^\kappa)$, 输出 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$ 作为公开参数, 其中 X 作为密文空间, Π 作为会话密钥空间.
- $\mathsf{KeyGen}(pp)$: 运行 $(pk, sk) \leftarrow \mathsf{HPS.KeyGen}(pp)$, 输出公钥 pk 和私钥 sk.
- $\mathsf{Encaps}(pk; r)$: 以公钥 pk 和随机数 r 为输入, 执行如下步骤:
  1. 运行 $(x, w) \leftarrow \mathsf{SampRel}(r)$ 生成随机实例和相应的证据;

2. 通过 HPS.PubEval(pk, x, w) 计算实例 x 的哈希证明 $\pi \leftarrow \mathsf{H}_{sk}(x)$;
3. 输出实例 x 作为密文 c, 输出哈希证明 π 作为会话密钥 k.

- Decaps(sk, c): 以私钥 sk 和密文 c 为输入, 通过 HPS.PrivEval(sk, c) 计算 c 的哈希证明 $\pi \leftarrow \mathsf{H}_{sk}(x)$ 以恢复会话密钥 k.

♣

KEM 方案的正确性由 HPS 的完备性保证. 安全性由如下定理保证.

定理 4.11

如果 $L \subseteq X$ 上的 SMP 困难问题成立, 那么构造 4.10中的 KEM 是 IND-CPA 安全的.

♡

证明 我们将通过游戏序列组织证明. 游戏序列的编排次序由如下证明思路指引:

- 将诚实生成的密文分布 $x \xleftarrow{R} L$ 切换为 $x \xleftarrow{R} X \backslash L$
- 论证当 $x \xleftarrow{R} X \backslash L$ 时, $(pk, \pi = \mathsf{H}_{sk}(x))$ 的分布与 $(pk, \pi \xleftarrow{R} \Pi)$ 统计接近.

Game$_0$: 对应真实的游戏, 其中挑战密文 $x^* \xleftarrow{R} L$, 计算会话密钥的方式是对 $\mathsf{H}_{sk}(x^*)$ 进行公开求值

- 初始化: $\mathcal{CH}$ 计算 $pp \leftarrow$ HPS.Setup(1^κ), $(pk, sk) \leftarrow$ HPS.KeyGen(pp), 将 pp 和 pk 发送给 $\mathcal{A}$.
- 挑战: $\mathcal{CH}$ 按照以下步骤生成挑战
  1. 随机采样$(x^*, w^*) \leftarrow \mathsf{SampRel}(r^*)$;
  2. 通过HPS.PubEval(pk, x^*, r^*)公开计算 $\pi^* \leftarrow \mathsf{H}_{sk}(x^*)$;
  3. 令 $c^* = x^*$, $k_0^* = \pi^*$, 随机采样 $k_1^* \xleftarrow{R} \Pi$;
  4. 随机选取 $\beta \xleftarrow{R} \{0, 1\}$, 将 (c^*, k_β^*) 发送给 $\mathcal{A}$ 作为挑战.

  敌手 $\mathcal{A}$ 在游戏中的视图包括 (pp, pk, x^*, k_β^*).
- 应答: $\mathcal{A}$ 输出对 β 的猜测 β', $\mathcal{A}$ 成功当且仅当 $\beta' = \beta$.

为了准备将挑战密文的分布从 $x^* \in L$ 切换到 $x^* \in X \backslash L$, 我们首先需要引入以下的游戏作为过渡, 这是因为分布切换后 x^* 已经不在语言 L 内, $\mathcal{CH}$ 无法再以公开求值的方式计算哈希证明, 所以需要提前改变 $\mathcal{CH}$ 的求值方式.

Game$_1$: 与 Game$_0$ 相比唯一的区别在于挑战阶段的步骤 2, $\mathcal{CH}$ 通过HPS.PrivEval(sk, x^*)秘密计算 $\pi^* \leftarrow \mathsf{H}_{sk}(x^*)$. $\mathsf{H}_{sk}(\cdot)$ 的投射性质保证了当 $x^* \in L$ 时, PubEval$(pk, x^*, w^*) = \mathsf{H}_{sk}(x^*) =$ PrivEval(sk, x^*). 因此在敌手的视角中, $\mathcal{CH}$ 所作出的改变完全不可察觉, 我们有:

$$\text{Game}_0 \equiv \text{Game}_1$$

经过 Game$_1$ 的铺垫, 我们可以顺利过渡到以下的 Game$_2$.

Game$_2$: 与 Game$_1$ 唯一的区别是调用 SampNo(r^*) 采样$x^* \leftarrow X \backslash L$. SMP 问题的困难性保证了敌手在相邻游戏中的视图计算不可区分:

$$\text{Game}_1 \approx_c \text{Game}_2$$

Game$_3$: 与 Game$_2$ 的惟一不同是在挑战阶段随机采样$\pi^* \xleftarrow{R} \Pi$替代 $\pi^* \leftarrow \mathsf{H}_{sk}(x^*)$. 由 $\mathsf{H}_{sk}(\cdot)$ 的平滑性保证:

$$\text{Game}_2 \approx_s \text{Game}_3$$

在 Game$_3$ 中, k_0^* 和 k_1^* 均是 Π 上的均匀分布, 因此即使对于拥有无穷计算能力的敌手 $\mathcal{A}$, 其优势也为 0. 综合以上, 定理得证! □

接下来, 我们将介绍如何基于 HPS 构造 IND-CCA 安全的 KEM. 在此之前, 我们先以自问自答的方式分析构造难点.

构造 4.10中的 KEM 方案是 IND-CCA 安全的么?

- 从归约证明的角度粗略分析似乎并没有技术困难, 因为归约算法 $\mathcal{R}$ 始终掌握私钥 sk, 可以回答任意的解封装询问. 然而细致分析后发现并非如此. 与 IND-CPA 安全游戏相比, 在 IND-CCA 安全游戏中, 敌手的视图

额外包括了对解封装询问的应答. 当解封装询问 $c = x$ 的密文 $x \notin L$ 时, 应答会泄漏更多关于 sk 的信息 (公钥 pk 可以看做关于 sk 的部分泄漏). 因此我们无法再使用平滑性得出 $\text{Game}_2 \approx_s \text{Game}_3$ 的结论.

接上问, 既然当 $x \in X \backslash L$ 时的解封装询问会泄漏 sk 的信息, 那拒绝此类询问是否可以达到 IND-CCA 安全性呢?

- 不可以. 这是因为 SMP 问题的困难性使得 PPT 的解密者无法判定是否 $x \in L$. 善于思考的读者很能发现解密者还拥有解密私钥 sk, 然而解密者 (对应诚实用户) 仅拥有一个解密私钥, 依然无法判定是否 $x \in L$. 那是否有巧妙的方案设计使得解密者拥有多个解密私钥, 从而解密者可以通过检测多个私钥求值的一致性来判定 $x \in ?L$ 了. 答案依然是否定的, 因为 SMP 的困难性否定了此类方案设计算法的存在性. 反过来, 如果解密者拥有了对应 SMP 问题公开参数对应的秘密参数, 那么确实可以设计方案使得解密者拥有多个解密私钥, 比如考虑 L_{DDH} 语言的 HPS 4.9, 如果解密者知晓 α 使得 $g_1^\alpha = g_2$, 那么任取 $\Delta \in \mathbb{Z}_q$, 均有:

$$(sk_1, sk_2) \sim (sk_1' = sk_1 + \alpha\Delta, sk_2' = sk_2 - \Delta) \Leftrightarrow g_1^{sk_1} g_2^{sk_2} = g_1^{sk_1'} g_2^{sk_2'}$$

  上述设计方案已经暗含了 SMP 问题的困难性对解密者不复存在, 这使得安全归约将会在 $\text{Game}_1 \approx_c \text{Game}_2$ 的步骤失败, 原因是归约算法 (针对 SMP 问题的敌手) 不掌握 α, 从而无法模拟解密者的行为.

通过以上的分析, 不难得出基于 HPS 构造 CCA 安全的 KEM 的一种思路是杜绝 "危险" 的解密询问:

- $x \in L$ 属于安全的解密询问, 这是因为应答 $\pi = \text{HPS.PubEval}(pk, x, w)$ 没有额外泄漏关于 sk 的信息, 因此不会破坏平滑性.
- $x \notin L$ 属于危险的解密询问, 杜绝的思路在密文中嵌入"私密认证结构", 使得 PPT 的敌手无法生成有效的 (valid) 危险密文, 同时解密者能够判定密文是否有效. 具体的设计思路是将哈希证明作为信息论意义下的一次性消息验证码 (information-theoretic one-time MAC), 此处需要满足 universal_2 性质的 HPS.

构造 4.11 (基于 HPS 的 CCA 安全的 KEM 构造)

构造的组件是:

- 满足平滑性质的 HPS_1
- 满足一致性质 (universal_2) 的 HPS_2

构造如下:

- $\text{Setup}(1^\kappa)$:
  1. 运行 $pp_1 \leftarrow \text{HPS}_1.\text{Setup}(1^\kappa)$, 其中 $pp_1 = (\mathsf{H}_1, SK_1, PK_1, X, L, W, \Pi_1, \alpha_1)$;
  2. 运行 $pp_2 \leftarrow \text{HPS}_2.\text{Setup}(1^\kappa)$, 其中 $pp_2 = (\mathsf{H}_2, SK_2, PK_2, X, L, W, \Pi_2, \alpha_2)$;
  3. 输出公开参数 $pp = (pp_1, pp_2)$. 公钥空间 $PK = PK_1 \times PK_2$, 私钥空间 $SK = SK_1 \times SK_2$, 密文空间 $C = X \times \Pi_2$, 会话密钥空间 $K = \Pi_1$.
- $\text{KeyGen}(pp)$: 解析 $pp = (pp_1, pp_2)$, 执行以下步骤:
  1. 计算 $(pk_1, sk_1) \leftarrow \text{HPS}_1.\text{KeyGen}(pp_1)$;
  2. 计算 $(pk_2, sk_2) \leftarrow \text{HPS}_2.\text{KeyGen}(pp_2)$;
  3. 输出公钥 $pk = (pk_1, pk_2)$ 和私钥 $sk = (sk_1, sk_2)$.
- $\text{Encaps}(pk; r)$: 以公钥 $pk = (pk_1, pk_2)$ 和随机数 r 为输入, 执行以下步骤:
  1. 运行 $(x, w) \leftarrow \text{SampRel}(r)$ 随机采样语言 L_1 中的实例和相应证据;
  2. 通过 $\text{HPS}_1.\text{PubEval}(pk_1, x, w)$ 计算实例 x 在 HPS_1 中的哈希证明 $\pi_1 \leftarrow \mathsf{H}_1(sk_1, x)$;
  3. 通过 $\text{HPS}_2.\text{PubEval}(pk_2, x, w)$ 计算实例 x 在 HPS_2 中的哈希证明 $\pi_2 \leftarrow \mathsf{H}_2(sk_2, x)$;
  4. 输出实例 x 和 π_2 作为密文 c, 其中 π_2 可以看做 x 的 MAC 值; 输出哈希证明 π_1 作为会话密钥 k.
- $\text{Decap}(sk, c)$: 以私钥 $sk = (sk_1, sk_2)$ 和密文 $c = (x, \pi_2)$ 为输入, 通过 $\text{HPS}_2.\text{PrivEval}(sk_2, x)$ 计算 x 的哈希证明 $\pi_2' \leftarrow \mathsf{H}_2(sk_2, x)$; 如果 $\pi_2 \neq \pi_2'$ 则输出 $\perp$, 否则通过 $\text{HPS}_1.\text{PrivEval}(sk_1, x)$ 计算 x 的哈希证明 $\pi_1 \leftarrow \mathsf{H}_1(sk_1, x)$ 以恢复会话密钥 k.

♣

构造 4.11的正确性由 HPS_1 和 HPS_2 的完备性保证, 安全性由以下定理保证.

定理 4.12

如果 $L \subseteq X$ 上的 SMP 问题成立, 那么构造 4.11 中的 KEM 是 IND-CCA 安全的.

证明 为了便于安全分析, 首先对密文 $c = (x, \pi_2)$ 做如下的分类:

- 良生成的 (well-formed) $\iff x \in L$
- 有效的 (valid) $\iff \mathsf{H}^2_{sk_2}(x) = \pi_2$

根据以上定义, 良生成的密文有可能是无效的, 有效的密文也可能是非良生成的. 在基于 HPS 构造的 KEM 中, 非良生成的密文是"危险的", 因为解封装询问的结果会泄漏关于私钥的信息.

以下通过游戏序列完成定理证明:

Game_0: 对应真实的游戏

- 初始化: $\mathcal{CH}$ 生成 $pp_1 \leftarrow \mathsf{HPS}_1.\mathsf{Setup}(1^\kappa)$, $pp_2 \leftarrow \mathsf{HPS}_2.\mathsf{Setup}(1^\kappa)$, 计算 $(pk_1, sk_1) \leftarrow \mathsf{HPS}_1.\mathsf{KeyGen}(pp_1)$, $(pk_2, sk_2) \leftarrow \mathsf{HPS}_2.\mathsf{KeyGen}(pp_2)$, 发送 $pp = (pp_1, pp_2)$ 和 $pk = (pk_1, pk_2)$ 给敌手 $\mathcal{A}$.
- 挑战: $\mathcal{CH}$ 执行以下操作生成挑战
  1. 运行$(x^*, w^*) \leftarrow (r^*)$随机采样 L 中的实例和证据;
  2. 通过$\mathsf{HPS}_1.\mathsf{PubEval}(pk_1, x^*, w^*)$计算哈希证明 $\pi_1^* \leftarrow \mathsf{H}_1(sk_1, x^*)$;
  3. 通过$\mathsf{HPS}_2.\mathsf{PubEval}(pk_2, x^*, w^*)$计算哈希证明 $\pi_2^* \leftarrow \mathsf{H}_2(sk_2, x^*)$;
  4. 令 $c^* = (x^*, \pi_2^*)$, $k_0^* = \pi_1^*$, $k_1^* \xleftarrow{\mathrm{R}} \Pi$;
  5. 选择随机比特 $\beta \xleftarrow{\mathrm{R}} \{0,1\}$, 发送 (c^*, k_β^*) 给 $\mathcal{A}$ 作为挑战.
- 解封装询问: 当敌手发起解封装询问 $c = (x, \pi_2)$ 时, $\mathcal{CH}$ 分情况应答如下:
  - $c = c^*$: 返回 $\perp$;
  - $c \neq c^*$: 如果$\pi_2 = \mathsf{HPS}_2.\mathsf{PrivEval}(sk_2, x)$ 返回 $\mathsf{HPS}_1.\mathsf{PrivEval}(sk_1, x)$; 否则返回 $\perp$.

Game_1: 与 CPA 构造情形类似, 该游戏的引入是为了将密文 c^* 由语言 L 内切换到语言外. 在挑战阶段, $\mathcal{CH}$ 通过$\mathsf{HPS}_1.\mathsf{PrivEval}(sk_1, x^*)$计算 $\pi_1^* \leftarrow \mathsf{H}_1(sk_1, x^*)$, 通过$\mathsf{HPS}_2.\mathsf{PrivEval}(sk_2, x^*)$计算 $\pi_2^* \leftarrow \mathsf{H}_2(sk_2, x^*)$. HPS 的投射性保证了 $\mathrm{Game}_0 \equiv \mathrm{Game}_1$.

Game_2: 将随机采样 L 中的实例和证据 $(x^*, w^*) \xleftarrow{\mathrm{R}} \mathsf{SampRel}(r^*)$ 切换为随机采样 $X \backslash L$ 中的实例$x^* \leftarrow \mathsf{SampNo}(r^*)$. SMP 问题的困难性保证了敌手在相邻游戏中的视图计算不可区分:

$$\mathrm{Game}_1 \approx_c \mathrm{Game}_2$$

在游戏序列演进过程中, 仅在论证 $\mathrm{Game}_1 \approx_c \mathrm{Game}_2$ 时依赖计算困难假设; 其余的分析均在信息论意义下 (information-theoretic) 完成, 从此刻起挑战者 $\mathcal{CH}$ 拥有无穷计算能力.

Game_3: 微调解密规则, 将直接拒绝非良生成但有效的 (ill-formed but valid) 密文. 对于解封装询问 $c = (x, \pi_2)$, 只要 $x \notin L$, 那么即使 $\pi_2 = \mathsf{H}^2_{sk_2}(x)$ 也直接返回 $\perp$ 表示拒绝. 改变规则的目的是拒绝所有危险密文, 从而确保解封装询问的应答不泄漏关于私钥的信息.

断言 4.7

$|\Pr[S_3] - \Pr[S_2]| \leq \mathsf{negl}(\kappa)$.

证明 注意到正常的解封装算法会对此类密文返回解封装结果, 并不是直接返回 $\perp$ 拒绝. 为了分析规则改变引发的差异, 引入如下事件 E:

- $\mathcal{A}$ 发起非良生成但有效的解封装询问, 即:$x \notin L \wedge \pi_2 = \mathsf{H}^2_{sk_2}(x)$

显然如果事件 E 不发生, 那么 Game_2 与 Game_3 完全相同. 令 Q 表示 $\mathcal{A}$ 发起解封装询问的最大次数, HPS_2 的 universal$_2$ 保证了:

$$\Pr[E] \leq Q/|\Pi_2| = \mathsf{negl}(\kappa)$$

利用差异引理, 断言得证. □

Game_4: 对所有良生成的解封装询问 $c=(x,\pi_2)$ 也即 $x\in L$, $\mathcal{CH}$ 使用公钥 $pk=(pk_1,pk_2)$ 和相应的证据 w 应答. 注意到 $\mathcal{CH}$ 拥有无穷计算能力, 因此能够计算出 $x\in L$ 的证据 w. 该规则变化仅是为了说明对良生成密文的解封装不会额外泄漏关于私钥的信息, 不会引发敌手视图的任何改变, 因此 $\text{Game}_3\equiv\text{Game}_4$.

Game_5: 随机采样$\pi_1^*\xleftarrow{R}\Pi_1$代替 $\pi^*\leftarrow\mathsf{H}_1(sk_1,x^*)$.

断言 4.8

敌手 $\mathcal{A}$ 在 Game_4 和 Game_5 中的视图统计不可区分.

证明 敌手 $\mathcal{A}$ 在 Game_4 和 Game_5 中的视图均由以下部分组成:

- 公开参数: $pp=(pp_1,pp_2)$;
- 公钥: $pk=(pk_1,pk_2)$;
- 挑战: 密文 $c^*=(x^*,\pi_2^*)$ 和会话密钥 k_β^*;
- 解封装询问: 由公钥 pk 和敌手 $\mathcal{A}$ 的询问确定.

接下来, 我们通过递增分布项的方式证明断言:

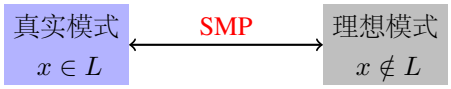
1. 首先由 HPS_1 的平滑性可知, 当 $x^*\xleftarrow{R}X\backslash L$ 是有:
$$(pk_1,x^*,\boxed{\mathsf{H}_1(sk_1,x^*)})\approx_s(pk_1,x^*,\boxed{U_{\Pi_1}})$$
2. 将 (pk_2,π_2^*) 表示为 $g_{sk_2}(x^*)$, 其中 $g_{sk_2}(x):=(\alpha_2(sk_2),\mathsf{H}_2(sk_2,x))$. 复合引理 (composition lemma) 可推出 $X\approx_s Y\Rightarrow f(X)\approx_s f(Y)$, 其中 f 可以是任意 (概率) 函数. 将上面公式左右两边的分布分别看成 X 和 Y, 令 $f(pk_2,x^*,\pi_2)=(g_{sk_2}(x^*),pk_2,x^*,\pi_2)$, 应用复合引理即可得:
$$(pk_2,\pi_2^*,\underline{pk_1,x^*,\mathsf{H}^1_{sk_1}(x^*)})\approx_s(pk_2,\pi_2^*,\underline{pk_1,x^*,U_{\Pi_1}})$$
令 $view'=(pk,x^*,\pi_2^*,k_\beta^*)$, 上面公式可以简写为 $view'_4\approx_s view'_5$.
3. 在左右两边添加解封装结果. $\mathcal{CH}$ 对解封装询问的应答总可以表示为 $f_{\mathsf{decaps}}(view')$, f_{decaps} 编码了敌手 $\mathcal{A}$ 选择密文 $\{c_i\}$ 的策略和解封装算法, 易知 f_{decaps} 是一个 PPT 算法. 再次应用复合引理, 可以得到:
$$(f_{\mathsf{decaps}}(view'_4),view'_4)\approx_s(f_{\mathsf{decaps}}(view'_5),view'_5)$$

根据敌手视图的定义, 可以得到 $\text{Game}_4\approx_s\text{Game}_5$, 断言得证. □

在 Game_5 中, k_0^* 和 k_1^* 均从 Π_1 中随机采样. 因此对于任意敌手均有 $\Pr[S_5]=0$. 综合以上, 定理得证! □

小结

HPS 给出了基于 SMP 类型判定性问题构造公钥加密的范式, 在论证安全性时遵循如下的三步走 (三板斧) 套路:



1. 真实游戏中挑战密文为语言中的随机实例 $x\in L$;
2. 理想游戏中挑战密文为语言外的随机实例 $x\notin L$, 在信息论意义下证明敌手优势可忽略;
3. 利用 SMP 完成语言内外的切换, 论证 PPT 敌手在真实游戏和理想游戏中的优势差可忽略.

论证过程与中国道家的 "阴阳相生" 思想暗合.

在很多情形下, 公钥加密的私钥嵌入于底层困难问题, 因此设计高等级安全公钥加密的一个常见难点是归约证明过程中, 归约算法 $\mathcal{R}$ 需要在未知私钥的情形下模拟与私钥相关的谕言机. 一个具体的例子就是难以证明 ElGamal PKE 具备私钥抗泄漏安全性, 因为私钥嵌入在底层 DDH 困难问题中. Cramer 和 Shoup 另辟蹊径, 绕过了该难点, 关窍是在基于 HPS 的公钥加密设计中, 公钥加密的密文嵌入于底层困难问题, 归约算法 $\mathcal{R}$ 始终掌握私钥, 从而可以完美模拟任意与私钥相关的谕言机. 正是该特性使得 HPS 的用途极为广泛, 远远超越了最初的 CCA 安全的公钥加密, 如 HPS 在基于口令的密钥交换 (password authenticated key exchange, PAKE)、不经意传输 (oblivious transfer, OT) 的构造中均有重要应用, 更是达成密钥泄漏安全、消息依赖密钥安全等高等级安全的主流技术工具.

4.3 可提取哈希证明系统类

事要知其所以然.

—《朱子语类 · 卷九 · 论行知》

1991 年, Rackoff 和 Simon [296] 提出了构造 CCA 安全 PKE 的另一条技术路线:

1. 发送方随机选择会话密钥 k 并使用接收方的公钥对其加密得到密文 c, 同时生成关于 k 的非交互零知识知识证明 π, 将 c 和 π 一起发送给接收方;
2. 接收方先验证 π 的正确性, 若验证通过则利用私钥解密恢复会话密钥;

该条技术路线被称为 Rackoff-Simon 范式, 与 Naor-Yung 范式/Sahai 范式的不同之处是前者需要使用非交互零知识知识的证明 (NIZKPoK), 而后者使用的是非交互零知识证明 (NIZK).

Cramer 和 Shoup 于 2002 年正式提出的哈希证明系统 [27] 是 NIZK 的弱化: 公开可验证弱化为指定验证者, 表达能力由任意 $\mathcal{NP}$ 语言限制为群论语言, 证明的形式特化为哈希值. 2010 年, Wee [28] 提出了可提取哈希证明系统 (extractable hash proof system, EHPS), 并展示了如何基于 EHPS 以一种简洁、模块化的方式构造 CCA 安全的 PKE. 该构造范式统一了几乎所有已知的基于计算性假设的 CCA 安全 PKE 方案. 相对 HPS 是 NIZK 的弱化, EHPS 是 NIZKPoK 的弱化. 以下首先介绍 EHPS 的定义和相关性质.

定义 4.11 (可提取哈希证明系统 (EHPS))

EHPS 包含以下 4 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 $pp = (\mathsf{H}, PK, SK, L, W, \Pi)$, 其中 L 是由困难关系 R_L 定义的平凡 $\mathcal{NP}$ 语言, $\mathsf{H} : PK \times L \to \Pi$ 是由公钥集合 PK 索引的一族带密钥哈希函数. 关系 R_L 支持随机采样, 即存在 PPT 算法 SampRel 以随机数 r 为输入, 输出随机的 "实例-证据" 元组 $(x, w) \in \mathsf{R}_L$. 为了方便后续的应用, SampRel 可以进一步分解为 SampYes 和 SampWit, 前者随机采样语言中的实例, 后者随机采样证据, 对于任意随机数 $r \in R$, 我们有 $(\mathsf{SampYes}(r), \mathsf{SampWit}(r)) \in \mathsf{R}_L$.
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入, 输出公钥 pk 和私钥 sk.
- $\mathsf{PubEval}(pk, x, r)$: 以公钥 pk、$x \in L$ 和随机数 r 为输入, 输出证明 $\pi \in \Pi$. 正确性要求是当 r 是采样 x 的随机数时 (即 $(x, w) \leftarrow \mathsf{SampRel}(r)$), 算法正确计算出哈希证明: $\pi = \mathsf{H}_{pk}(x)$. 注意, 当给定采样随机数 r 时, 可以运行算法 SampRel 恢复 x, 因此算法的第 2 项输入 x 可以省去.
- $\mathsf{Ext}(sk, x, \pi)$: 以私钥 sk, $x \in L$ 和证明 $\pi \in \Pi$ 为输入, 输出证据 $w \in W \cup \perp$. 正确性要求是:

$$\pi = \mathsf{H}_{pk}(x) \iff (x, \mathsf{Ext}(sk, x, \pi)) \in \mathsf{R}_L$$

- $\mathsf{KeyGen}'(pp)$: 以公开参数 pp 为输入, 输出公钥 pk 和私钥 sk'.
- $\mathsf{PrivEval}(sk', x)$: 以私钥 sk 和 $x \in L$ 为输入, 输出证明 $\pi \in \Pi$. 正确性要求是 PrivEval 正确计算出哈希证明: $\pi = \mathsf{H}_{pk}(x)$.

♣

以上算法中, KeyGen、PubEval 和 Ext 工作在真实模式, KeyGen' 和 PrivEval 工作在模拟模式, 两种模式共享同一个 Setup 算法生成公开参数. 两种模式之间的关联是公钥的分布统计不可区分, 即:

$$\mathsf{KeyGen}(pp)[1] \approx_s \mathsf{KeyGen}'(pp)[1]$$

4.3.1 可提取哈希证明系统的起源释疑

笔记 图 4.20解释了 EHPS 的命名渊源, 其本质上是指定验证者零知识知识的证明, 证明的形式是实例的哈希值, 故名可提取哈希证明系统.

- DV-NIZKPoK 的完备性和可提取性由 Ext 的正确性保证, 即在正常模式下,

$$\pi = \mathsf{H}_{pk}(x) \iff (x, \mathsf{Ext}(sk, x, \pi)) \in \mathsf{R}_L$$

其中 $\mathsf{KeyGen}(pp) \to (pk, sk)$.

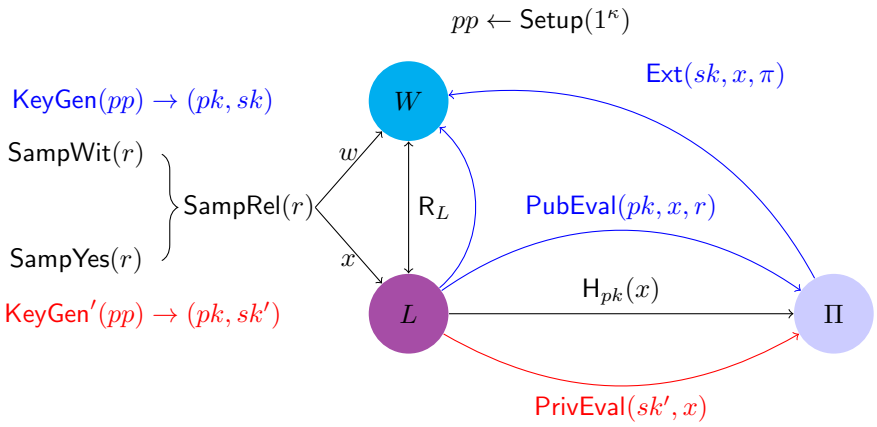


图 4.19: EHPS 的示意图

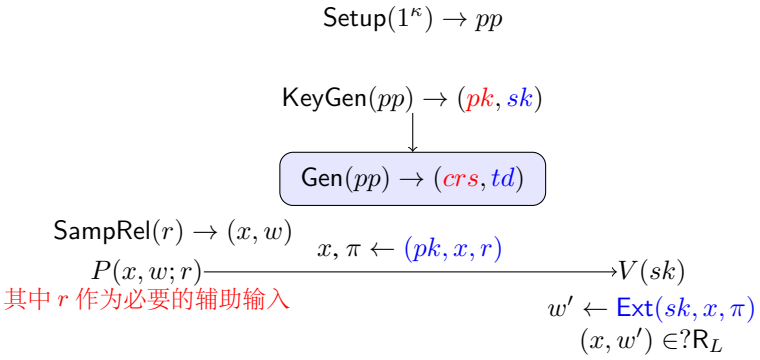


图 4.20: 从 DV-NIZKPoK 的视角解析 EHPS

- DV-NIZKPoK 的零知识性论证如下, 令 $\mathsf{KeyGen}(pp) \to (pk, sk')$, $\mathsf{KeyGen}'(pp) \to (pk, sk')$, 对于 $\forall x \in L$, 我们有:

$$pk \approx_s pk \Rightarrow (pk, \mathsf{H}_{pk}(x)) \approx_s (pk, \mathsf{H}_{pk}(x))$$

再由秘密求值算法的正确性 $\mathsf{H}_{pk}(x) = \mathsf{PrivEval}(sk', x)$ 可以得到:

$$(pk, \mathsf{H}_{pk}(x)) \approx_s (pk, \mathsf{PrivEval}(sk', x))$$

4.3.2 可提取哈希证明系统的实例化

我们以针对 L_{CDH} 语言的 EHPS 构造为例, 获得对 EHPS 设计方式的直观认识. 令 $(\mathbb{G}, q, g)$ 是算法 $\mathsf{GenGroup}(1^\kappa)$ 的输出, 其中 $\mathbb{G}$ 是阶为素数 q 的循环群, g 是生成元. 随机选取 $\mathbb{G}$ 中的另一生成元 g^α, 其中 $\alpha \xleftarrow{\mathrm{R}} \mathbb{Z}_q$. 令 $pp = (\mathbb{G}, q, g, g^\alpha)$ 是公开参数, 定义由 pp 索引的平凡 $\mathcal{NP}$ 语言如下:

$$L_{\mathsf{CDH}} = \{x \in X : \exists w \in W \text{ s.t. } w = x^\alpha\}$$

其中 $L = X = \mathbb{G}, W = \mathbb{G}$. 定义 L_{CDH} 的二元关系为 $\mathsf{R}_{\mathsf{CDH}}$, $(x, w) \in \mathsf{R}_{\mathsf{CDH}} \iff w = x^\alpha$. 容易验证:

- $\mathsf{R}_{\mathsf{CDH}}$ 基于 CDH 假设是困难的.
- $\mathsf{R}_{\mathsf{CDH}}$ 是高效可采样的: 存在 PPT 采样算法 SampRel 随机选取 $r \xleftarrow{\mathrm{R}} \mathbb{Z}_q$, 输出 $(g^r, (g^\alpha)^r) \in \mathsf{R}_{\mathsf{CDH}}$.
- 如果 $\mathbb{G}$ 是双线性映射群, 则 $\mathsf{R}_{\mathsf{CDH}}$ 是公开可验证的.

构造 4.12 (L_{CDH} 语言的 EHPS 构造)

L_{CDH} 的 EHPS 构造如下, 如图所示:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 $pp = (\mathbb{G}, q, g, g^\alpha)$. 其中 pp 还包括了对 $SK = \mathbb{Z}_q$,

$PK = \mathbb{G}$, $L_{\mathrm{CDH}} = X = \mathbb{G}$ 和 $W = \mathbb{G}$ 的描述.

- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入, 随机采样 $sk \xleftarrow{R} \mathbb{Z}_q$, 计算 $pk = g^{sk} \in \mathbb{G}$, 输出 (pk, sk).
- $\mathsf{PubEval}(pk, x, r)$: 以公钥 pk、实例 $x \in L_{\mathrm{CDH}}$ 和 $r \in \mathbb{Z}_q$ 为输入, 输出 $\pi \leftarrow (g^{\alpha} \cdot pk)^r$.
- $\mathsf{Ext}(sk, x, \pi)$: 以私钥 sk、实例 $x \in L_{\mathrm{CDH}}$ 和 π 为输入, 计算 $w \leftarrow \pi / x^{sk}$, 如果 $(x, w) \in \mathsf{R}_L$ 则返回 w, 否则返回 $\perp$. 正确性由以下公式保证:

$$\pi / x^{sk} = (g^{\alpha} \cdot pk)^r / x^{sk} = (g^{\alpha} \cdot g^{sk})^r / g^{r \cdot sk} = (g^{\alpha})^r = w$$

- $\mathsf{KeyGen}'(pp)$: 以公开参数 pp 为输入, 随机采样 $sk' \xleftarrow{R} \mathbb{Z}_q$, 计算 $pk \leftarrow g^{sk'} / g^{\alpha}$.
- $\mathsf{PrivEval}(sk', x)$: 以私钥 sk' 和实例 $x \in L_{\mathrm{CDH}}$ 为输入, 输出 $w \leftarrow x^{sk'}$. 正确性由以下公式保证:

$$\mathsf{H}_{pk}(x) = (g^{\alpha} \cdot pk)^r = (g^{\alpha} \cdot \underline{g^{sk'} / g^{\alpha}})^r = (g^{sk'})^r = x^{sk'}$$

容易验证, 两种模式下生成的 pk 服从同样的分布—$\mathbb{G}$ 上的均匀分布.

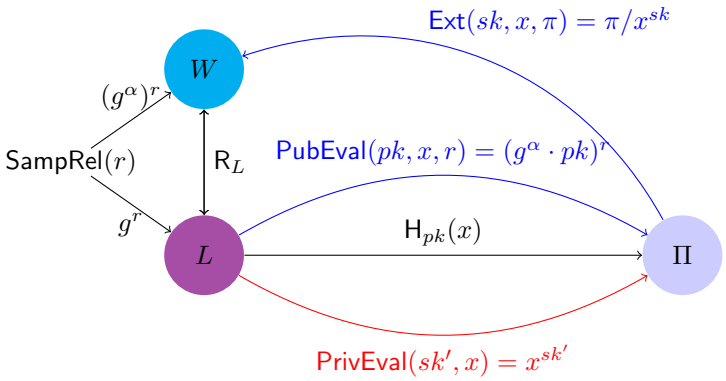


图 4.21: L_{CDH} 的 EHPS

4.3.3 基于可提取哈希证明系统的 KEM 构造

作为暖场应用, 我们首先介绍如何基于 EHPS 构造 CPA 安全的 KEM. 设计的思路源自 Rackoff-Simon 范式. 令 R_L 为定义在 $X \times W$ 上的单向关系, $\mathsf{hc} : W \to K$ 为相应的 hardcore function.

- 发送方扮演 EHPS 中的证明者, 运行 $\mathsf{SampRel}(r)$ 算法随机采样 $(x, w) \in \mathsf{R}_L$, 利用公钥 pk 和随机数 r 计算 x 的哈希证明 π, 生成密文 $c = (x, \pi)$, 计算证据 w 的 hardcore function 值作为会话密钥 k.
- 接收方扮演 EHPS 中的验证者: 使用私钥 sk 从密文 (x, π) 中恢复 w, 进而恢复会话密钥.

构造 4.13 (基于 EHPS 的 CPA 安全的 KEM 构造)

从语言 L 的 EHPS 出发, 构造 CPA 安全的 KEM 如下:

- $\mathsf{Setup}(1^{\kappa})$: 运行 $pp \leftarrow \mathsf{EHPS.Setup}(1^{\kappa})$, 输出 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi)$ 作为公开参数, 其中 $X \times \Pi$ 作为密文空间, 关系 R_L 对应的 hardcore function 值域 K 作为会话密钥空间.
- $\mathsf{KeyGen}(pp)$: 运行 $(pk, sk) \leftarrow \mathsf{EHPS.KeyGen}(pp)$, 输出公钥 pk 和私钥 sk.
- $\mathsf{Encaps}(pk; r)$: 以公钥 pk 和随机数 r 为输入, 执行如下步骤:
  1. 运行 $(x, w) \leftarrow \mathsf{SampRel}(r)$ 生成随机实例和相应证据;
  2. 通过 $\mathsf{EHPS.PubEval}(pk, x, r)$ 计算实例 x 的哈希证明 $\pi \leftarrow \mathsf{H}_{pk}(x)$;
  3. 输出 (x, π) 作为密文, 计算 $k \leftarrow \mathsf{hc}(w)$ 作为会话密钥.
- $\mathsf{Decaps}(sk, c)$: 以私钥 sk 和密文 $c = (x, \pi)$ 为输入, 计算 $w \leftarrow \mathsf{EHPS.Ext}(sk, x, \pi)$, 如果 $(x, w) \notin \mathsf{R}_L$ 则输出 $\perp$, 否则输出 $k \leftarrow \mathsf{hc}(w)$.

KEM 的正确性由 EHPS 的完备性和 R_L 的单射性保证, 安全性由如下定理保证.

定理 4.13

如果 R_L 是单向的, 那么构造 4.13 中的 KEM 是 IND-CPA 安全的. ♡

证明 证明的目标是论证会话密钥 $\mathsf{hc}(w^*)$ 在敌手 $\mathcal{A}$ 的视图中是伪随机的, 其中 $\mathcal{A}$ 的视图包括:

- 公开参数 pp;
- 公钥 pk: 与 w^* 无关;
- 密文 $c^* = (x^*, \pi^*)$: R_L 的单向性保证了 x^* 隐藏了 w^*, EHPS 的零知识性进一步保证了 π^*(相对于 x^*) 不会额外泄漏关于 w^* 的信息.

我们通过以下的游戏序列组织证明.

Game_0: 对应真实的游戏. $\mathcal{CH}$ 在真实模式下运行 EHPS 与敌手 $\mathcal{A}$ 交互.

- 初始化: $\mathcal{CH}$ 计算 $pp \leftarrow \mathsf{EHPS.Setup}(1^\kappa)$, $(pk, sk) \leftarrow \mathsf{EHPS.KeyGen}(pp)$, 将 pp 和 pk 发送给 $\mathcal{A}$.
- 挑战: $\mathcal{CH}$ 按照以下步骤生成挑战
  1. 随机采样 $(x^*, w^*) \leftarrow \mathsf{SampRel}(r^*)$;
  2. 通过$\mathsf{EHPS.PubEval}(pk, x^*, r^*)$公开计算 $\pi^* \leftarrow \mathsf{H}_{pk}(x^*)$;
  3. 计算 $k_0^* \leftarrow \mathsf{hc}(w^*)$, 随机采样 $k_1^* \xleftarrow{\mathrm{R}} K$;
  4. 随机选取 $\beta \xleftarrow{\mathrm{R}} \{0,1\}$, 将 $(c^* = (x^*, \pi^*), k_\beta^*)$ 发送给 $\mathcal{A}$ 作为挑战.
- 应答: $\mathcal{A}$ 输出对 β 的猜测 β', $\mathcal{A}$ 成功当且仅当 $\beta' = \beta$.

为了利用 EHPS 的零知识性论证 π^* 不额外泄漏关于 w^* 的信息, 需要将 EHPS 由真实模式切换到模拟模式.

Game_1: $\mathcal{CH}$ 在模拟模式下运行 EHPS 与敌手 $\mathcal{A}$ 交互.

- 初始化: $\mathcal{CH}$ 计算$(pk, sk') \leftarrow \mathsf{EHPS.KeyGen}'(pp)$.
- 挑战: $\mathcal{CH}$ 在第二步通过$\mathsf{EHPS.PrivEval}(sk', x^*)$计算 $\pi^* \leftarrow \mathsf{H}_{pk}(x^*)$.

敌手 $\mathcal{A}$ 在游戏中的视图为 $(pp, pk, x^*, \pi^*, k_\beta^*)$. 容易验证, EHPS 的零知识性保证了 $\mathrm{Game}_0 \approx_s \mathrm{Game}_1$:

断言 4.9

如果 R_L 是单向的, $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{Game}_1} = \mathsf{negl}(\kappa)$. ♡

证明 证明思路是如果存在 $\mathcal{A}$ 以不可忽略的优势赢得 Game_1, 那么可以构造出 $\mathcal{B}$ 以不可忽略的优势打破 hc 的伪随机性, 从而与单向性假设冲突. 给定关于 hc 的伪随机性挑战 pp 和 (x^*, k_β^*), 其中 $(x^*, w^*) \leftarrow \mathsf{SampRel}(r^*)$, $\mathcal{B}$ 模拟 Game_1 中的挑战者 $\mathcal{CH}$ 与 $\mathcal{A}$ 交互, 目标是猜测 β.

- $\mathcal{B}$ 运行 EHPS 的模拟模式与 $\mathcal{A}$ 在 Game_1 进行交互, 在初始化阶段不再采样 x^* 而是直接嵌入接收到的 x^*, 在挑战阶段将 R_L 的挑战 (x^*, k_β^*) 作为 $\mathcal{A}$ 的 KEM 挑战. 最终, $\mathcal{B}$ 输出 $\mathcal{A}$ 的猜测 β'.

容易验证, $\mathcal{B}$ 在 Game_1 中的模拟是完美的. 因此 $\mathcal{B}$ 打破 R_L 伪随机性的优势与 $\mathsf{Adv}_{\mathcal{A}}^{\mathrm{Game}_1}(\kappa)$ 相同. 断言得证! □

综上, 定理得证! □

在介绍如何基于 EHPS 构造 IND-CCA 安全的 KEM 之前, 我们首先从安全归约的角度分析设计难点. EHPS 模拟模式下的 sk' 可以在不知晓采样实例 x 随机数的情况下正确计算出相应的哈希证明, 但无法提取出证据, 因此归约算法无法应答解密询问. 因此, 为了构造 IND-CCA 的 KEM, 需要赋予 EHPS 更丰富的功能.

PKE/KEM 的选择密文安全游戏是 "全除一"(all-but-one, ABO) 式的—$\mathcal{A}$ 可以发起除挑战密文 x^* 以外的任意解密/解封装询问. Wee [28] 引入了量身定制的 ABO-EHPS.

定义 4.12 (全除一可提取哈希证明系统 (ABO-EHPS))

ABO-EHPS 与 EHPS 的定义差别集中在模拟模式, 真实模式下完全相同. 与 EHPS 相比, ABO-EHPS 在模拟模式下的功能更加丰富.

- $\mathsf{KeyGen}'(pp, x^*)$: 以公开参数 pp 和 $x^* \in L$ 为输入, 输出 (pk, sk').

- $\mathsf{PrivEval}(sk', x^*)$: 以私钥 sk' 和 x^* 为输出, 输出证明 $\pi^* = \mathsf{H}_{pk}(x^*)$.
- $\mathsf{Ext}'(sk', x, \pi)$: 以私钥 sk'、$x \neq x^*$ 和 $\pi \in \Pi$ 为输入, 输出证据 $w \in W$. 正确性的要求是:

$$\pi = \mathsf{H}_{pk}(x) \iff (x, \mathsf{Ext}'(sk', x, \pi)) \in \mathsf{R}_L$$

KeyGen' 算法以预先嵌入的点 x^* 为输入, 输出相应的密钥对 (pk, sk'). ABO 的含义是模拟模式中的 sk' 具备以下功能:

- "一除全" 哈希求值 (one-out-all evaluation): sk' 可以计算 x^* 的哈希值 $\mathsf{H}_{pk}(x^*)$.
- "全除一" 证据抽取 (all-but-one extraction): sk' 可以从除 x^* 以外的点 x 和相应的证明中正确抽取出证据 $\mathsf{Ext}'(sk', x, \pi)$.

♣

注记 4.12

模拟模式下 sk' 的功能在 CCA 安全归约中起到如下作用:

- "一除全" 哈希求值允许归约算法 $\mathcal{R}$ 生成挑战密文 $c^* = (x^*, \pi^*)$.
- "全除一" 证据抽取允许归约算法 $\mathcal{R}$ 回答所有合法的解封装询问 $c \neq c^*$.

♠

Wee [28] 展示了如何基于 EHPS 构造 ABO-EHPS.

构造 4.14 (基于 EHPS 的 ABO-EHPS 构造)

起点: 二元关系 R_L 的 EHPS

设计目标: 二元关系 R_L 的 ABO-EHPS

设计思路: 不妨设 L 中每个实例均可编码为 n 长的比特串, 利用 DDN 结构 [23] 实现 ABO 功能. 具体构造如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp \leftarrow \mathsf{EHPS.Setup}(1^\kappa)$;
- $\mathsf{KeyGen}(pp)$: 独立运行 $\mathsf{EHPS.KeyGen}(pp)$ 算法 $2n$ 次, 生成 $\{(pk_{i,b}, sk_{i,b})\}_{i\in[n],b\in\{0,1\}}$, 输出公钥 $pk = \{pk_{i,0}, pk_{i,1}\}_{i\in[n]}$ 和私钥 $sk = \{sk_{i,0}, sk_{i,1}\}_{i\in[n]}$.
- $\mathsf{PubEval}(pk, x, r)$: 对所有的 $i \in [n]$ 计算 $\pi_i \leftarrow \mathsf{EHPS.PubEval}(pk_{i,x_i}, x, r)$, 输出 $\pi = (\pi_1, \ldots, \pi_n)$.
- $\mathsf{Ext}(sk, x, \pi)$: 对所有的 $i \in [n]$ 计算 $w_i \leftarrow \mathsf{EHPS.Ext}(sk_{i,x_i}, x, \pi_i)$, 如果所有结果一致则输出, 否则返回 $\perp$.
- $\mathsf{KeyGen}'(pp, x^*)$: 独立运行 $\mathsf{EHPS.KeyGen}'(pp)$ 算法 n 次生成 $\{(pk_{i,x_i^*}, sk_{i,x_i^*})\}_{i\in[n]}$, 独立运行 $\mathsf{EHPS.KeyGen}(pp)$ 算法 n 次生成 $\{(pk_{i,1-x_i^*}, sk_{i,1-x_i^*})\}_{i\in[\ell]}$, 输出 $pk = (pk_{i,0}, pk_{i,1})_{i\in[n]}$ and $sk' = (sk_{i,0}, sk_{i,1})_{i\in[n]}$.
- $\mathsf{PrivEval}(sk', x^*)$: 对所有的 $i \in [n]$ 计算 $\pi_i \leftarrow \mathsf{EHPS.PrivEval}'(sk_{i,x_i^*}, x^*)$, 输出 $\pi = (\pi_1, \ldots, \pi_n)$.
- $\mathsf{Ext}'(sk', x, \pi)$: 对所有满足 $x_i^* = x_i$ 的索引 $i \in [n]$ 验证 $\pi_i = \mathsf{EHPS.PrivEval}(sk_{i,x_i}, x)$ 是否成立, 如果否则输出 $\perp$, 如果是则继续对所有满足 $x_i^* \neq x_i$ 的索引 $i \in [n]$ 计算 $\mathsf{EHPS.Ext}(sk_{i,x_i}, x, \pi_i)$, 如果提取结果一致则输出, 否则输出 $\perp$.

♣

ABO-EHPS 真实模式下算法的正确性由 EHPS 对应算法保证.

$n = 3$

$pk_{1,0}$ $sk_{1,0}$	$pk_{2,0}$ $sk_{2,0}$	$pk_{3,0}$ $sk_{3,0}$
$pk_{1,1}$ $sk_{1,1}$	$pk_{2,1}$ $sk_{2,1}$	$pk_{3,1}$ $sk_{3,1}$

图 4.22: 真实模式下 $n = 3$ 时密钥结构图示

ABO-EHPS 模拟模式下算法的正确性由 DDN 结构和 EHPS 对应算法保证. ABO-EHPS 两种模式下公钥分布的统计不可区分性由 EHPS 两种模式下公钥分布的统计不可区分性与各公钥分量生成的独立性保证.

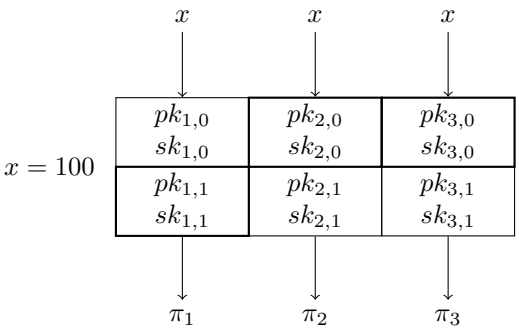


图 4.23: 真实模式下 $x = 100$ 时哈希证明计算图示

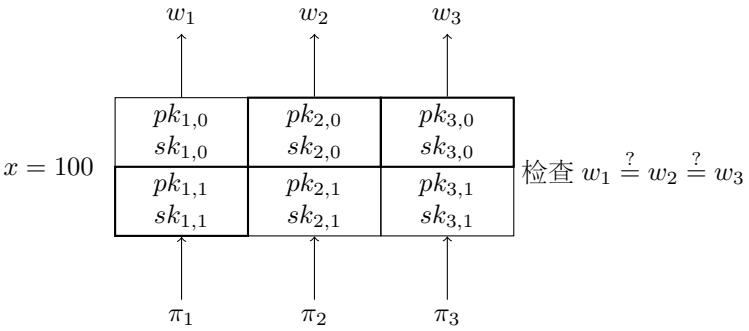


图 4.24: 真实模式下 $x = 100$ 时证据提取图示

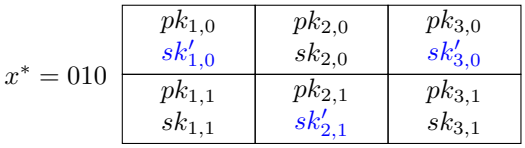


图 4.25: 模拟模式下 $n = 3$, $x^* = 010$ 时的密钥生成

不知晓随机数, 使用 $\mathsf{PrivEval}(sk', x^*)$ 计算

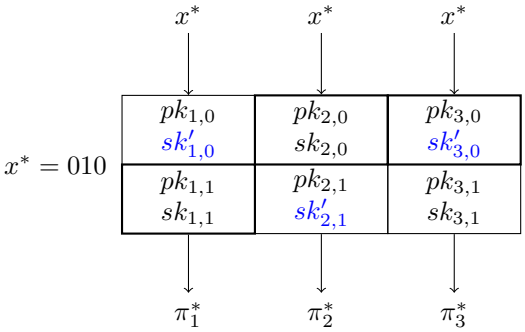


图 4.26: 模拟模式下 $x^* = 010$ 时哈希证明计算

知晓随机数, 使用 $\mathsf{PubEval}(pk, x, r)$ 计算

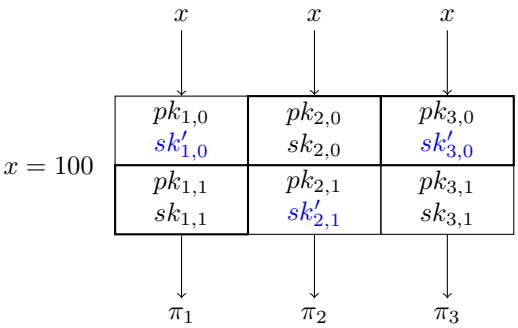
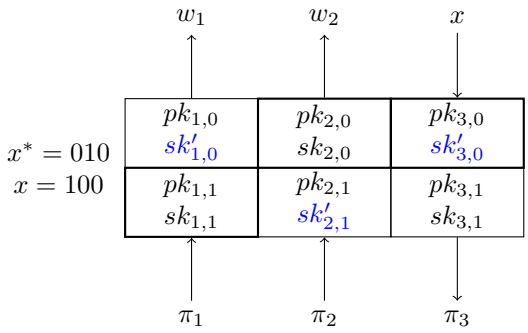


图 4.27: 模拟模式下 $x = 100$ 时哈希证明计算



$w_1 \leftarrow \mathsf{EHPS.Ext}(sk_{1,1}, \pi_1)$
$w_2 \leftarrow \mathsf{EHPS.Ext}(sk_{2,0}, \pi_2)$
检查 $w_1 \stackrel{?}{=} w_2$
检查 $\pi_3 \stackrel{?}{=} \mathsf{EHPS.PrivEval}(sk'_{3,0}, x)$

图 4.28: 模拟模式下 $x = 100$ 时证据提取过程

基于 ABO-EHPS 设计 IND-CCA KEM 的方式与构造 4.13完全相同. KEM 构造的正确性由 ABO-EHPS 的正确性和 R_L 的单射性保证, 安全性由以下定理保证.

定理 4.14

如果 R_L 是单向的, 那么 KEM 是 IND-CCA 安全的. ♡

证明 证明的思路仍然是首先由真实模式切换到模拟模式, 再在模拟模式下利用零知识性证明安全性. 证明的要点是保证两种模式下对解密谕言机 $\mathcal{O}_{\text{decap}}$ 回复的一致性. 以下通过游戏序列完成定理证明:

Game_0: 对应真实的游戏. $\mathcal{CH}$ 在真实模式下运行 ABO-EHPS 与敌手 $\mathcal{A}$ 交互.

- 初始化: $\mathcal{CH}$ 计算 $pp \leftarrow \mathsf{Setup}(1^\kappa)$, $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$, 将 pp 和 pk 发送 $\mathcal{A}$.
- 挑战: $\mathcal{CH}$ 按照以下步骤生成挑战
  1. 随机采样 $(x^*, w^*) \leftarrow \mathsf{SampRel}(r^*)$;
  2. 通过$\mathsf{PubEval}(pk, x^*, r^*)$公开计算 $\pi^* \leftarrow \mathsf{H}_{pk}(x^*)$;
  3. 计算 $k_0^* \leftarrow \mathsf{hc}(w^*)$, 随机采样 $k_1^* \xleftarrow{\text{R}} K$;
  4. 随机选取 $\beta \xleftarrow{\text{R}} \{0,1\}$, 将 $(c^* = (x^*, \pi^*), k_\beta^*)$ 发送给 $\mathcal{A}$ 作为挑战.
- 解封装询问 $c = (x, \pi) \neq c^*$: 计算$w \leftarrow \mathsf{Ext}(sk, x, \pi)$, 如果 $(x, w) \in \mathsf{R}_L$ 则输出 $\mathsf{hc}(w)$, 否则输出 $\perp$.
- 应答: $\mathcal{A}$ 输出对 β 的猜测 β', $\mathcal{A}$ 成功当且仅当 $\beta' = \beta$.

为了利用 ABO-EHPS 的零知识性论证 π^* 和解封装询问不额外泄漏关于 w^* 的信息, 需要将 ABO-EHPS 由真实模式切换到模拟模式. 为此, 先引入以下游戏作为过渡.

Game_1: 与 Game_0 完全相同, 惟一的区别是 $\mathcal{CH}$ 将 $(x^*, w^*) \leftarrow \mathsf{SampRel}(r^*)$ 由挑战阶段提前至初始化阶段. 显然, 该变化不会对敌手的视图有任何改变. 因此有:

$$\text{Game}_0 \equiv \text{Game}_1$$

Game_2: 本游戏对解封装应答方式稍加改动, 以便于后续游戏将密文的 ABO 解封装询问转化为 ABO-EHPS 相对于 x^* 的 ABO 证据抽取. 对于解封装询问 $c = (x, \pi) \neq c^*$, $\mathcal{CH}$ 应答如下:

- $x = x^* \wedge \pi \neq \pi^*$: 直接返回 $\perp$.
- $x \neq x^*$: 计算 $w \leftarrow \mathsf{Ext}(sk, x, \pi)$, 如果 $(x, w) \in \mathsf{R}_L$ 则返回 $\mathsf{hc}(w)$, 否则返回 $\perp$.

由于 H_{pk} 是确定性算法, 因此 Game_2 与 Game_1 中的解封装应答完全相同.

Game_3: $\mathcal{CH}$ 在模拟模式下运行 ABO-EHPS 与敌手 $\mathcal{A}$ 交互.

- 初始化: $\mathcal{CH}$ 与上一游戏的区别在于通过$(pk, sk') \leftarrow \mathsf{KeyGen}'(pp, x^*)$生成密钥对.
- 挑战: $\mathcal{CH}$ 与上一游戏的区别在于通过$\mathsf{PrivEval}(sk', x^*)$ 计算 $\pi^* \leftarrow \mathsf{H}_{pk}(x^*)$.
- 解封装询问 $c = (x, \pi) \neq c^*$: $\mathcal{CH}$ 应答如下
  - $x = x^* \wedge \pi \neq \pi^*$: 直接返回 $\perp$.
  - $x \neq x^*$: 计算$w \leftarrow \mathsf{Ext}'(sk', x, \pi)$, 如果 $(x, w) \in \mathsf{R}_L$ 则返回 $\mathsf{hc}(w)$, 否则返回 $\perp$.

基于以下事实, 我们有: $\text{Game}_2 \approx_s \text{Game}_3$

- $\mathsf{KeyGen}(pp)[1] \approx_s \mathsf{KeyGen}'(pp, x^*)[1]$
- $\mathsf{PubEval}(pk, x^*, r^*) = \mathsf{H}_{pk}(x^*) = \mathsf{PrivEval}(sk', x^*)$
- 对于解封装询问 $c = (x, \pi)$: 当 $x = x^*$ 时, 均返回 $\perp$; 当 $x \neq x^*$ 时, ABO-EHPS 真实模式和模拟模式的正确性以及解封装算法 "提取-检验" 的设计保证了应答一致.

断言 4.10

如果 R_L 是单向的, 那么 $\mathsf{Adv}_{\mathcal{A}}^{\text{Game}_3} = \mathsf{negl}(\kappa)$. ♡

证明 证明思路是如果存在 $\mathcal{A}$ 以不可忽略的优势赢得 Game_3, 那么可以构造出 $\mathcal{B}$ 以不可忽略的优势打破 hc 的伪随机性, 从而与 R_L 的单向性假设冲突. 给定关于 hc 的伪随机性挑战 pp 和 (x^*, k_β^*), 其中 $(x^*, w^*) \leftarrow \mathsf{SampRel}(r^*)$, $\mathcal{B}$ 模拟 Game_3 中的挑战者 $\mathcal{CH}$ 与 $\mathcal{A}$ 交互, 目标是猜测 β.

- $\mathcal{B}$ 运行 ABO-EHPS 的模拟模式与 $\mathcal{A}$ 进行交互, 其在初始化阶段不再采样 x^* 而是直接嵌入接收到的 x^*, 在挑战阶段将 hc 的挑战 (x^*, k^*_β) 作为 $\mathcal{A}$ 的 KEM 挑战. 最终, $\mathcal{B}$ 输出 $\mathcal{A}$ 的猜测 β'.

容易验证, $\mathcal{B}$ 在 Game_3 中的模拟是完美的. 因此 $\mathcal{B}$ 打破 hc 伪随机性的优势与 $\mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{A}}$ 相同. 断言得证!

综上, 定理得证!

Wee [28] 展示了 ABO-EHPS 蕴含 ATDR.

构造 4.15 (基于 ABO-EHPS 的 ATDR 构造)

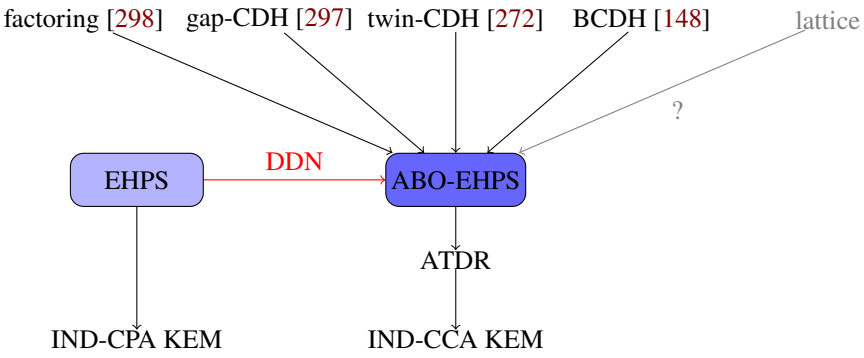
- $\mathsf{Setup}(1^\kappa)$: 运行 $pp \leftarrow \mathsf{ABO\text{-}EHPS.Setup}(1^\kappa)$.
- $\mathsf{KeyGen}(pp)$: 运行 $(pk, sk) \leftarrow \mathsf{ABO\text{-}EHPS.KeyGen}(pp)$, 令 pk 为求值公钥 ek, sk 为求逆陷门 td.
- $\mathsf{Sample}(pk; r)$: 运行 $(x, w) \leftarrow \mathsf{SampRel}(r)$, 通过 $\mathsf{ABO\text{-}EHPS.PubEval}(pk, x, r)$ 计算 $\pi \leftarrow \mathsf{H}_{pk}(x)$, 输出 $(w, (x, \pi))$.
- $\mathsf{TdInv}(td, (x, \pi))$: 计算 $w \leftarrow \mathsf{ABO\text{-}EHPS.Ext}(sk, (x, \pi))$, 如果 $(x, w) \in \mathsf{R}$ 则返回 w, 否则返回 $\perp$.

♣

上述 ATDR 构造的自适应单向性由 ABO-EHPS 的性质 R_L 的单向性保证. 该构造也在更抽象的层面解释了基于 ABO-EHPS 设计 CCA 安全 KEM 的实质是在构造 ATDR.

小结

EHPS 的理论价值在于它阐释统一了一大类标准模型下的基于计算性假设的 IND-CCA 安全的 PKE 方案 [297, 272, 298, 148], 尚未解决的公开问题是能否构造出关于格基困难问题的 EHPS. 目前, 绝大多数标准模型下的 PKE 构造都可纳入 EHPS 和 HPS 的设计范式, 这也从公钥加密的角度展现了零知识证明的强大威力.



HPS 与 EHPS 的对比

相同点

- 均可看成指定验证者的零知识证明系统 (DV-NIZK).
- 证明的形式是哈希值.

不同点

- HPS 是标准的证明系统, 而 EHPS 是知识的证明系统.
- HPS 中哈希函数族 H_{sk} 由私钥索引, EHPS 中哈希函数族 H_{pk} 有公钥索引.
- 在基于 HPS 的 PKE 构造中, 密文 c 是实例 x, 会话密钥 k 是证明 π.
  - HPS 的正确性保证了 PKE 的正确性
  - HPS 的合理性 (哈希函数的平滑性、一致性) 与 SMP 问题的困难性保证了 PKE 的安全性, 在证明过程中, 挑战实例需要从语言 L 上切换到语言外 $X \backslash L$.
- 在基于 EHPS 的 PKE 构造中, 密文 c 由实例 x 和证明 π 组成, 会话密钥 k 是证据 w.
  - EHPS 的知识提取性质保证了 PKE 的正确性
  - EHPS 的零知识性和二元关系的单向性保证了 PKE 的安全性, 在证明过程中, EHPS 需要由真实模式切换为模拟模式.

4.4 程序混淆类

只要代码足够乱, 就没人能看得懂.

— 防御性编程 (代码混淆)

4.4.1 程序混淆的定义与安全性

程序混淆 (program obfuscation) 是一种编译的方法技术, 它将容易理解的源程序转化成难以理解的形式, 同时保持原有功能性不变. 程序混淆概念起源于上世界 70 年代的代码混淆领域, 在软件保护领域 (如软件水印、防逆向工程) 有着广泛的应用, 然而一直缺乏严格的安全定义.

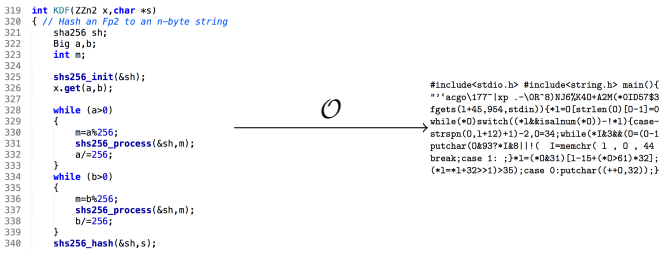


图 4.29: 程序混淆

Barak 等 [299] 首次将程序混淆引入密码学领域, 将程序从狭义的代码泛化为广义的算法, 同时提出了几乎黑盒 (virtual black-box, VBB) 混淆的严格定义, 如图 4.30所示.

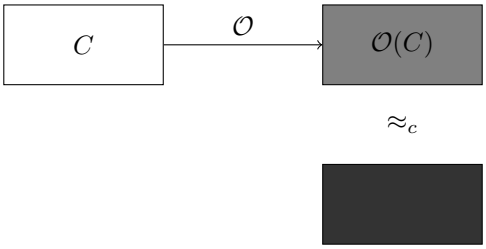


图 4.30: 几乎黑盒混淆

定义 4.13 (几乎黑盒混淆)

我们称一个 PPT 算法 $\mathcal{O}$ 是电路簇 $\{\mathcal{C}_\kappa\}$ 的几乎黑盒混淆器当且仅当其满足以下两个条件:

- 功能保持: 对于任意安全参数 $\kappa \in \mathbb{N}$、任意的 $C \in \mathcal{C}_\kappa$ 和所有输入 $x \in \{0,1\}^*$ 有:

$$\Pr[C'(x) = C(x) : C' \leftarrow \mathcal{O}(\kappa, C)] = 1$$

- 几乎黑盒混淆: 存在 PPT 的模拟器 $\mathcal{S}$, 对于任意 $C \in \{\mathcal{C}_\kappa\}$, 对于任意 PPT 敌手 $\mathcal{A}$, 我们有:

$$\mathcal{A}(\mathcal{O}(C)) \approx_c \mathcal{S}^C$$

其中公式左边表示 $\mathcal{A}$ 的视图, 公式右边表示 $\mathcal{S}$ 在通过对 C 进行黑盒访问所输出的视图.

笔记 几乎黑盒混淆的安全性定义是基于模拟方式, 刻画的是 PPT 敌手从混淆程序 $\mathcal{O}(C)$ 中获取的任何信息不会比黑盒访问 C 获得的信息更多. 换言之, 掌握 $\mathcal{O}(C)$ 的敌手视图可以由模拟器通过黑盒访问 C 模拟得出. VBB 试图隐藏程序 C 的所有细节. 比如 C 以平方差公式计算 x^2-1, 即 $C(x)=(x+1)(x-1)$. 那么敌手在获得 $\mathcal{O}(C)$ 后, 掌握的所有信息与输入输出元组 (x, x^2-1), i.e., $(1,0), (2,3), (3,8), \ldots$ 相同.

VBB 混淆定义强到极致, 因此在密码学中应用起来颇为简单直观. 事实上, 在 1976 年 Diffie 和 Hellman 的划时代论文 [3] 中, 就已经提出了利用混淆器将对称加密方案编译为公钥加密方案的想法 (如图 4.31):

1. 将 SKE 加密算法 $\mathsf{Enc}(sk, m, r)$ 中的第一个输入固化进 (hardwire) 电路, 得到 $\mathsf{Enc}_{sk}(m, r)$;
2. 利用混淆器编译 $\mathsf{Enc}_{sk}(\cdot,\cdot)$, 将得到的混淆程序作为公钥 pk.

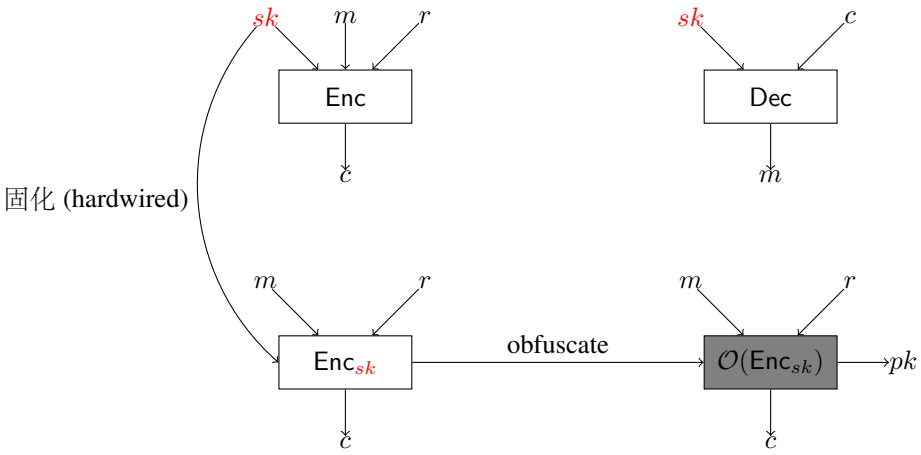


图 4.31: SKE ⇒ PKE via obfuscation

VBB 混淆的定义至强, Barak 等 [299] 指出 VBB"too good to be true!"–不存在针对任意电路 (通用, general-purpose) 的 VBB 混淆. VBB 混淆因为安全太强以至于不存在, Garg 等 [189] 降低了安全性要求, 引入了不可区分混淆 (indistinguishability obfuscator, $i\mathcal{O}$).

定义 4.14 (不可区分混淆 ($i\mathcal{O}$))

我们称一个 PPT 算法 $i\mathcal{O}$ 是电路簇 $\{\mathcal{C}_\kappa\}$ 的不可区分混淆器当且仅当其满足以下两个条件:

- 功能保持: 对于任意安全参数 $\kappa \in \mathbb{N}$、任意的 $C \in \mathcal{C}_\kappa$ 和所有输入 $x \in \{0,1\}^*$ 有:
$$\Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(\kappa, C)] = 1$$
- 不可区分混淆: 对于任意 PPT 敌手 $(\mathcal{S}, \mathcal{D})$, 存在关于安全参数的可忽略函数 α 使得: 如果 $\Pr[\forall x, C_0(x) = C_1(x) : (C_0, C_1, aux) \leftarrow \mathcal{S}(\kappa)] \geq 1 - \alpha(\kappa)$, 那么我们有:
$$|\Pr[\mathcal{D}(aux, i\mathcal{O}(\kappa, C_0)) = 1] - \Pr[\mathcal{D}(aux, i\mathcal{O}(\kappa, C_1)) = 1]| \leq \alpha(\kappa)$$

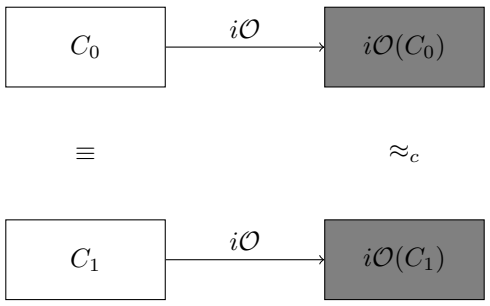


图 4.32: 不可区分混淆示意图

注记 4.13

- 不可区分混淆的定义类似加密方案的不可区分性, 对于任意功能相同的电路 C_0 和 C_1, 均有 $i\mathcal{O}(C_0) \approx_c i\mathcal{O}(C_1)$. 这里, 我们可以把电路 C 类比为消息, $i\mathcal{O}$ 类比为加密算法. 与 VBB 试图隐藏电路的所有信息不同, $i\mathcal{O}$ 只试图隐藏电路的部分信息: 比如 $C_0(x) = (x+1)(x-1)$, $C_1(x) = (x+2)(x-2)+3$, 那么如果混淆后的程序均是 $x^2 - 1$ 即可满足不可区分安全性. 非严格的说, $i\mathcal{O}$ 试图在以统一的方式完成同质的计算.

- 在上述定义中，条件$\Pr[\forall x, C_0(x) = C_1(x) : (C_0, C_1, aux) \leftarrow \mathcal{S}(\kappa)] \geq 1 - \alpha(\kappa)$ 并不意味着 C_0 和 C_1 存在差异输入 (differing-inputs)，而指的是 C_0 和 C_1 以极高的概率功能性完全相同，这一点体现在概率空间定义在 $\mathcal{S}$ 的随机带而与 x 无关.
- aux 表示 $\mathcal{S}$ 在采样 C_0, C_1 过程中得到的任意信息，用于辅助 $\mathcal{D}$ 区分 $i\mathcal{O}(C_0)$ 和 $i\mathcal{O}(C_1)$.

差异输入混淆. 如果在上述 $i\mathcal{O}$ 定义中，将 $\mathcal{S}$ 所采样两个电路的要求由功能性完全相同放宽为允许存在差异输入，则得到的是更强的混淆器，称为差异输入混淆 ($di\mathcal{O}$, differing-input obfuscation). [190] 中给出了正面结果：证明了 $i\mathcal{O}$ 蕴含多项式级别差异输入规模的 $di\mathcal{O}$. [300] 中给出了负面结果：证明了亚指数安全 (sub-exponentially secure) 的单向函数存在，则针对无界输入 Turing 机 (TMs with unbounded inputs) 亚指数安全的 $di\mathcal{O}$ 不存在.

我们再把注意力转回不可区分混淆. 如上所述，VBB 易用但对于通用电路并不存在，$i\mathcal{O}$ 弱化了安全要求，从而有了基于合理困难性假设的构造. 安全性弱化后 $i\mathcal{O}$ 是否还有着强大的威力？如何去应用呢？直观上：混淆后的程序既可以保持功能性，又能够在某种程度上隐藏常量. 常量皆程序. 在密码学中，公钥和私钥均可以看做一段程序，其中硬编码 (hardwired) 原本的公钥和私钥作为常量：比如加密就是以明文和随机数为输入，运行 “公钥程序”，输出密文；解密就是以密文为输入，运行 “解密程序”，输出明文. 混淆在密码学中的一类强大应用就是完成从 Minicrypt 到 Cryptomania 的穿越，因为借助混淆，可以在不泄漏秘密的情况下以公开的方式执行某个任务.

- 保持功能性 ⇒ 确保密码方案的功能性
- 在某种程度上隐藏常量 (对应需要保护的秘密) ⇒ 确保密码方案的安全性

注记 4.14

混淆的威力强大如魔法，其力量的来源在于对底层密码组件的调用方式是非黑盒的 (non-black-box)，因此可以绕过黑盒意义下的不可能结果 (black-box impossibilities).

在 $i\mathcal{O}$ 提出后最初的一段时间，应用只局限于属性加密 (ABE). 原因是应用 $i\mathcal{O}$ 设计密码方案并非易事，需要解决的技术难题是精准的隐藏 “部分信息”. 2014 年，Sahai 和 Waters [35] 创造性的发展了可穿孔编程技术 (puncture program technique)，以此给出了应用 $i\mathcal{O}$ 的范式，展示了 $i\mathcal{O}$ 的巨大威力–结合单向函数和 $i\mathcal{O}$ 重构了几乎所有的密码组件，包括公钥加密/密钥封装、可否认加密、数字签名、单向陷门函数、非交互式零知识证明、不经意传输等.

4.4.2 基于不可区分混淆的 KEM 构造

本章将逐步的展示如何基于 $i\mathcal{O}$ 构造 KEM，实现 Diffie-Hellman 当年的梦想.

起点方案. 首先将对称场景下基于 PRF 的 KEM 表达为程序的形式，如下图所示.

Encaps

Input: 伪随机函数的私钥 $\boxed{sk}$ 和随机数 x

1. output $c = x$, $k \leftarrow \mathsf{PRF}(sk, c)$.

再对程序进行微调，将 sk 由输入变为 hardwired 的常量，如下图所示：

Encaps

Constants: 伪随机函数的私钥 sk

Input: 随机数 x

1. output $c = x, k \leftarrow F(sk, c)$.

由于通用的 VBB 混淆器并不存在, 因此尝试用 $i\mathcal{O}$ 对程序混淆, 将混淆后的结果作为公钥

$$pk \leftarrow i\mathcal{O}(\mathsf{Encaps})$$

技术困难 1. 在将 KEM 的 IND-CCA 安全性归约到 PRF 的伪随机性时, 会遇到以下矛盾点:

- 在构造层面, 归约算法 $\mathcal{R}$ 需要掌握 sk 以生成 pk
- 为了让归约有意义, 归约算法 $\mathcal{R}$ 不能掌握私钥 sk

观察到 KEM 的 IND-CCA 安全仅要求随机挑战密文 c^* 封装的会话密钥是伪随机的, 因此消除矛盾点的核心想法是使用可穿孔伪随机函数替代标准伪随机函数, 在挑战密文 c^* 处穿孔:

- 生成 sk_{c^*} 得以对 c^* 外的所有点求值, 同时保持 $F_{sk}(c^*)$ 的伪随机性.
- 利用 sk_{c^*} 替代 sk 构建程序并混淆生成公钥.

Encaps

Constants: 可穿孔伪随机函数的私钥 sk

Input: 随机数 x

1. 输出 $c \leftarrow x, k \leftarrow F(sk, c)$.

方案构造: $pk \leftarrow i\mathcal{O}(\mathsf{Encaps})$

Encaps^*

Constants: 可穿孔伪随机函数的穿孔私钥sk_{c^*} 和穿孔点 c^*

Input: 随机数 x

1. 输出 $c \leftarrow x, k \leftarrow F(sk_{c^*}, c)$.

归约证明: $pk \leftarrow i\mathcal{O}(\mathsf{Encaps}^*)$

技术困难 2. 我们首先来分析归约证明中将要遇到的困难. 在模拟游戏中, 归约算法 $\mathcal{R}$ 仅需要使用 sk_{c^*} 即可构建程序 Encaps, 因此会话密钥 $k^* \leftarrow F(sk, c^*)$ 的伪随机性可以归约到可穿孔伪随机函数的安全性上. 我们仍需证明敌手在真实游戏与模拟游戏中的视图不可区分. 在此过程中, 遇到的第一个障碍是由于在 $x^* := c^*$ 处穿孔, 敌手可以通过观察程序在 x^* 的输出从而轻易区分真实游戏与模拟游戏:

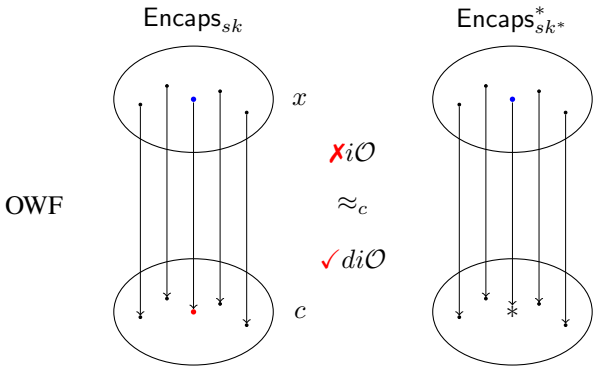
- 真实游戏: $\mathsf{Encaps}(x^*)$ 返回 k^* (已经不安全)
- 模拟游戏: $\mathsf{Encaps}^*(x^*)$ 返回 $\perp$

以上设计不成立的根本原因是

- 密文的设定$\boxed{c = x}$ $\Rightarrow$ 差异输入 x^* 将被挑战密文 c^* 直接暴露

为了隐藏差异输入, 初步的尝试为:

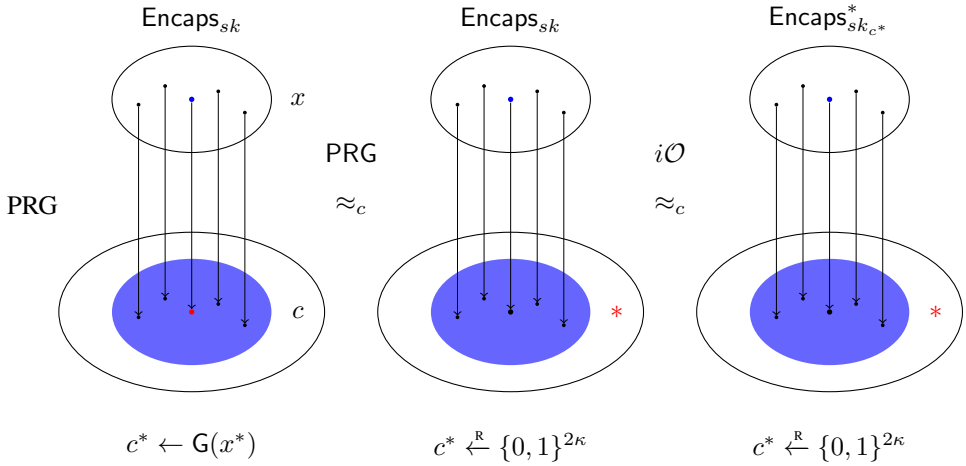
- $c = x \rightsquigarrow \boxed{c = f(x)}$, 其中 f 是单向函数.



使用单向函数对挑战点进行隐藏并没有消除差异输入, Encaps$_{sk}$ 与 Encaps$^*_{sk_{c^*}}$ 的输入输出行为存在不一致, 因此不满足 $i\mathcal{O}$ 的应用条件, 需要使用更强的 $di\mathcal{O}$.

消除差异输入的方法是将穿孔点 c^* 以敌手不可察觉的方式移到输入计算路径之外. 大致的技术路线是:

- 真实构造: $c \leftarrow \mathsf{OWF}(x) \rightsquigarrow \boxed{c \leftarrow \mathsf{G}(x)}$, 其中 G 是伪随机数发生器;
- 过渡游戏: 将 $c^* \leftarrow \mathsf{G}(x^*)$ 切换为 $c^* \xleftarrow{R} \{0,1\}^{2\kappa}$, 利用 PRG 的安全性保证切换不可察觉;
- 最终游戏: 利用 sk_{c^*} 替代 sk, 利用 $i\mathcal{O}$ 的安全性保证替代不可察觉.



综合以上, 最终的构造如下:

构造 4.16 (基于不可区分混淆的 CCA 安全的 KEM 构造)

构造所需的组件是:

- 不可区分混淆 $i\mathcal{O}$
- 伪随机数发生器 PRG $\mathsf{G} : \{0,1\}^\kappa \to \{0,1\}^{2\kappa}$
- 可穿孔伪随机函数 PPRF $F : SK \times \{0,1\}^{2\kappa} \to Y$

构造 KEM 如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp \leftarrow \mathsf{PPRF.Setup}(1^\kappa)$ 生成公开参数, 私钥空间 SK 为可穿孔伪随机函数的密钥空间 K, 密文空间 $C = \{0,1\}^{2\kappa}$, 会话密钥空间 $K = Y$.
- $\mathsf{KeyGen}(pp)$: 随机采样 $sk \xleftarrow{R} SK$, 计算 $pk \leftarrow i\mathcal{O}(\mathsf{Encaps})$.
- $\mathsf{Encaps}(pk; r)$: 运行 $(c, k) \leftarrow pk(r)$.
- $\mathsf{Decaps}(sk, c)$: 输出 $k \leftarrow F(sk, c)$.

Encaps

Constants: 可穿孔伪随机函数的私钥 sk

Input: 随机数 $x \in \{0,1\}^\kappa$

1. 输出 $c \leftarrow \mathsf{G}(x), k \leftarrow F(sk, c)$

构造 4.16的正确性显然, 安全性由以下定理保证.

定理 4.15

如果 F 是安全的可穿孔伪随机函数、G 是安全的伪随机数发生器、$i\mathcal{O}$ 是不可区分混淆, 则构造 4.16满足 IND-CCA 安全性.

♡

证明 以下通过游戏序列完成定理证明.

Game_0: 对应真实游戏

- 初始化: $\mathcal{CH}$ 运行 $\mathsf{PPRF.Setup}(1^\kappa)$ 生成公开参数, 随机采样 $sk \xleftarrow{R} SK$, 生成公钥$pk \leftarrow i\mathcal{O}(\mathsf{Encaps})$
- 挑战阶段: $\mathcal{CH}$ 随机采样$x^* \xleftarrow{R} \{0,1\}^\kappa$, 计算 $c^* \leftarrow \mathsf{G}(x^*)$, $k_0^* \leftarrow F(sk, c^*)$, 随机采样 $k_1^* \xleftarrow{R} K$, $\beta \xleftarrow{R} \{0,1\}$, 将 (c^*, k_β^*) 发送给 $\mathcal{A}$ 作为挑战.
- 解封装询问: $\mathcal{A}$ 发起询问 $c \in C$, $\mathcal{CH}$ 返回$k \leftarrow F(sk, c)$.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β', 攻击成功当且仅当 $\beta = \beta'$.

Game_1: 与 Game_0 的区别是 $\mathcal{CH}$ 在挑战阶段随机采样 $c^* \xleftarrow{R} \{0,1\}^{2\kappa}$ 而非计算 $c^* \leftarrow \mathsf{G}(x^*)$. PRG 的伪随机性保证了:

$$\text{Game}_0 \approx_c \text{Game}_1$$

Game_2: 与 Game_0 的区别是 $\mathcal{CH}$ 将 c^* 的生成从挑战阶段提前到初始化阶段 (为后续使用可穿孔伪随机函数做准备). 该变化完全隐藏于敌手, 因此有:

$$\text{Game}_1 \equiv \text{Game}_2$$

Game_3: $\mathcal{CH}$ 在初始化阶段计算 $pk \leftarrow i\mathcal{O}(\mathsf{Encap}^*)$ 而非之前的 $pk \leftarrow i\mathcal{O}(\mathsf{Encap})$; 在应答解封装询问时, 使用 sk_c^* 计算并返回 $k \leftarrow F(sk_{c^*}, c)$, 代替之前使用 k 计算并返回 $k \leftarrow F(sk, c)$.

Encaps^*

Constants: 可穿孔伪随机函数的穿孔私钥 sk_{c^*} 和穿孔点 c^*

Input: 随机数 $x \in \{0,1\}^\kappa$

1. 输出 $c \leftarrow \mathsf{G}(x), k \leftarrow F(sk_{c^*}, c)$.

- 由于 $\Pr[c^* \in \text{Img}(\mathsf{G})] = 1/2^\kappa$, 因此 c^* 落在 G 的像集中的概率可忽略, 故而穿孔导致程序输入输出行为差异的概率可忽略, 即 $\Pr[\mathsf{Encaps}_{sk} \equiv \mathsf{Encaps}_{sk_{c^*}}] = 1 - 1/2^\kappa$. $i\mathcal{O}$ 的安全性保证了公钥的分布计算不可区分

$$i\mathcal{O}(\mathsf{Encaps}) \approx_c i\mathcal{O}(\mathsf{Encaps}^*)$$

- 对于所有合法的解密询问 $c \neq c^*$, 可穿孔伪随机函数的正确性保证了 $F(sk, c) = F(sk_{c^*}, c)$.

因此, 我们有

$$\text{Game}_2 \approx_c \text{Game}_3$$

Game_4: $\mathcal{CH}$ 随机采样 $k_0^* \xleftarrow{R} K$ 代替上一游戏的 $k_0^* \leftarrow F(sk, c^*)$. 可穿孔伪随机函数的弱伪随机性保证了

$$\text{Game}_3 \approx_c \text{Game}_4$$

在 Game_4, k_0^* 和 k_1^* 均从 K 中均匀随机采样, 因此即使 $\mathcal{A}$ 拥有无穷的计算能力, 其在 Game_4 中的优势也是 0.

综合以上, 定理得证! □

注记 4.15

构造 4.16中的 KEM 也具备可穿孔性质. 该构造充分展示了 $i\mathcal{O}$ 的魔力——使得在不暴露秘密的情况下可以公开执行“内嵌秘密值”的程序:

- 将私钥组件编译为公钥组件

4.5 可公开求值伪随机函数类

明修栈道, 暗渡陈仓.

— 汉 · 司马迁《史记 · 高祖本纪》

前面的章节已经展示了若干种构造公钥加密的通用方法, 包括单向陷门函数、哈希证明系统、可提取哈希证明系统以及不可区分混淆结合可穿孔伪随机函数. 这些通用构造阐释了绝大多数公钥加密方案, 然而令人颇感以外的是, 它们并无法阐释最经典的 ElGamal PKE [37] 和 Goldwasser-Micali PKE [236] 另一方面, 伪随机函数是密码学的核心基本组件之一, 应用范围极其广泛, 特别的, 伪随机函数蕴含了简洁优雅的、也是目前唯一的 IND-CPA SKE 通用构造.

$$\mathsf{Enc}(sk, m; r) \to (r, F(sk, x) \oplus m)$$

然而伪随机函数属于 Minicrypt, 因此在黑盒意义下无法蕴含 PKE.

以上的现象促使我们考虑如下的问题:

思考 4.1

是否存在新型的伪随机函数能够让 PRF-based SKE 延拓到公钥场景? 新型的伪随机函数是否能蕴含统一上述的不同构造, 并阐释经典 PKE 方案的设计机理?

我们首先分析基于 PRF 构造 PKE 的技术难点:

- 密文必须可以公开计算: 显然, PRF-based SKE 的构造正是因为这个原因无法延拓到公钥加密场景中

$$(x, F(sk, x) \oplus m)$$

因为 F 的伪随机性意味着其不可能公开求值.

解决上述问题的关键在于探求伪随机性 (pseudorandomness) 和可公开求值性 (public evaluability) 是否能够共存. 标准的伪随机函数处处伪随机 (universal pseudorandom), 即对于定义域中任意 $x \in X$, PPT 敌手 $\mathcal{A}$ 都无法区分 $F_k(x)$ 和随机值.

- 观察 1: 构造 IND-CPA KEM 仅需要弱伪随机性 (weak pseudorandomness), 即对于挑战者随机选择的挑战输入, 其 PRF 值是伪随机的.
- 观察 2: 如果掌握输入 x 的某些辅助信息 aux (比如采样 x 的随机数), 是有可能在不使用 sk 的情形下对 $F_{sk}(x)$ 公开求值. 如果 aux 在平均意义下是难以抽取的, 则公开求值性与弱伪随机性不冲突.

综合以上, 在 KEM 中由发送方生成 x, 因此其知晓 aux 信息, 从而以下两点成为可能:

- 功能性方面: 发送方可以借助 aux 对 $F_{sk}(x)$ 公开求值从而生成密文.
- 安全性方面: $F_{sk}(x)$ 在 $\mathcal{A}$ 的视图中仍然伪随机.

4.5.1 可公开求值伪随机函数的定义与安全性

正是基于上面的思考, 陈等 [301] 提出了可公开求值伪随机函数 (publicly evaluable PRFs, PEPRF). PEPRF 考虑了定义域 X 包含 $\mathcal{NP}$ 语言 L 的情形, 使用私钥可以对全域求值, 而使用公钥和证据可以对语言 L 内的元素求值. 在安全性上, PEPRF 要求函数在语言 L 上弱伪随机.

定义 4.15 (可公开求值伪随机函数)

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 $pp = (F, PK, SK, X, L, W, Y)$, 其中 $F : SK \times X \to Y \cup \perp$ 是由 SK 索引的一族函数, $L \subseteq X$ 是由困难关系 R_L 定义的 $\mathcal{NP}$ 语言, 其中 W 是相应的证据集合. R_L 是高效可采样的, 存在 PPT 算法 $\mathsf{SampRel}$ 以随机数 r 为输入, 输出实例证据元组 $(x, w) \in \mathsf{R}_L$.
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入, 输出公钥 pk 和私钥 sk.
- $\mathsf{PrivEval}(sk, x)$: 以私钥 sk 和元素 $x \in X$ 为输入, 输出 $y \in Y \cup \perp$.
- $\mathsf{PubEval}(pk, x, w)$: 以公钥 pk、实例 $x \in L$ 以及相应的证据 $w \in W$ 为输入, 输出 $y \in Y$.

注记 4.16

在有些场景中有必要将单一语言 L 泛化为由 PK 索引的一族语言 $\{L_{pk}\}_{pk\in PK}$. 相应的, 采样算法 SampRel 将以 pk 为额外输入, 随机采样 $(x,w)\in \mathsf{R}_{L_{pk}}$.

正确性. 对于任意 $pp\leftarrow \mathsf{Setup}(1^\kappa)$ 和 $(pk,sk)\leftarrow \mathsf{KeyGen}(pp)$, 我们有:

$$\begin{aligned} \forall x\in X: &\quad F_{sk}(x)=\mathsf{PrivEval}(sk,x)\\ \forall x\in L \text{ 以及证据 } w: &\quad F_{sk}(x)=\mathsf{PubEval}(pk,x,w)\end{aligned}$$

(自适应) 弱伪随机性. 定义敌手 $\mathcal{A}$ 的优势函数如下:

$$\Pr\left[\beta=\beta' : \begin{array}{l} pp\leftarrow \mathsf{Setup}(1^\kappa);\\ (pk,sk)\leftarrow \mathsf{KeyGen}(pp);\\ r^*\xleftarrow{R} R,(x^*,w^*)\leftarrow \mathsf{SampRel}(r^*);\\ y_0^*\leftarrow F_{sk}(x^*), y_1^*\leftarrow Y;\\ \beta\leftarrow\{0,1\};\\ \beta'\leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{eval}}(\cdot)}(pp,pk,x^*,y_b^*); \end{array}\right]-\frac{1}{2}.$$

其中 $\mathcal{O}_{\mathsf{eval}}$ 表示求值谕言机, 以 $x\neq x^*\in X$ 为输入, 返回 $F_{sk}(x)$. 如果任意 PPT 敌手 $\mathcal{A}$ 在上述游戏中的优势函数均为可忽略函数, 则称可公开求值伪随机函数是弱伪随机的. 如果敌手在上述游戏中可以访问 $\mathcal{O}_{\mathsf{eval}}$ 谕言机, 则称可公开求值伪随机函数是自适应弱伪随机的.

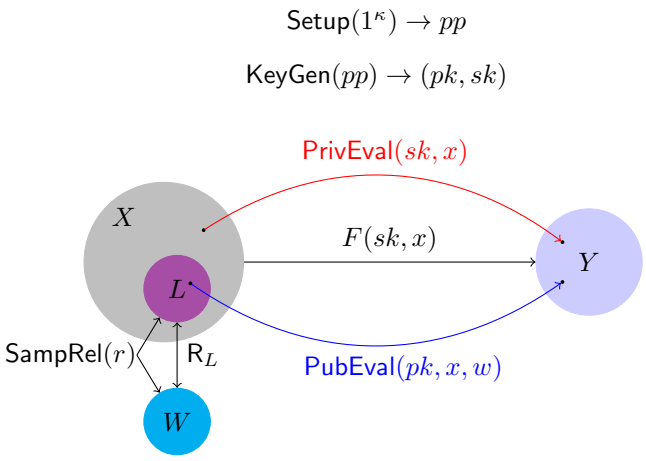


图 4.33: PEPRF 示意图

注记 4.17

在 PEPRF 中, 私钥用于秘密求值, 公钥则可在知晓相应证据时对语言内的元素进行公开求值. 密钥成对出现这一点对于 PEPRF 是自然的, 因为 PEPRF 是作为 PRF 在 Cryptomania 中的对应引入的. 另一方面, 标准的 PRF 也总是可以设置公钥用于发布与私钥相关联但可公开的信息, 例如在基于 DDH 假设的 Naor-Reingold PRF [302] 中, $F_{\vec{a}}(x)=(g^{a_0})^{\prod_{i:x_i=1}a_i}$, 其中 $\vec{a}=(a_0,a_1,\ldots,a_n)\in\mathbb{Z}_q^n$ 是私钥, $\{g^{a_i}\}_{1\le i\le n}$ 则可发布为公钥. 如果没有信息可公开, 可设定 $pk=\{\perp\}$. 如此可保持 PRF 与 PEPRF 的语法定义保持一致.

为什么 PEPRF 只定义了弱伪随机性呢? 这是因为在公开求值算法 PubEval 存在的前提下, 这是可达的最强安全性.

为了加深对概念的理解, 表 4.1对比分析 PRF 与 PEPRF 的异同. 正是由于上述区别, 我们可以基于 PEPRF 构造 KEM.

	PRF	PEPRF
带密钥函数	✓	✓
可公开求值	$\forall x \in X$ ✗	$x \in L$ ✓
安全性	$\forall x \in X$ pseudorandom	$x \xleftarrow{R} L$ weak pseudorandom

表 4.1: PRF 与 PEPRF 的比较

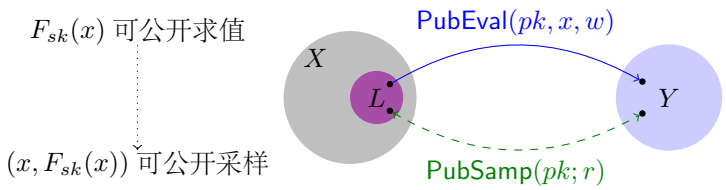
PEPRF 的定义可以进一步泛化以包容更多实例化构造.

定义 4.16 (可公开采样伪随机函数 (PSPRF, Publicly Sampleable PRF))

PSPRF 将 PEPRF 的可公开求值功能可以放宽为可公开采样功能, 即 PubEval 算法由以下的 PPT 随机采样算法替代:

- $\mathsf{PubSamp}(pk; r) \to (x, y) \in L \times Y$ s.t. $y = F_{sk}(x)$

♣



显然, 可以综合关系采样算法和函数公开求值算法构造公开采样算法, 因此 PEPRF 蕴含 PSPRF:

- $\mathsf{PubSamp}(pk; r)$: 运行 $(x, w) \leftarrow \mathsf{SampRel}(r)$, 输出 $(x, \mathsf{PEPRF.PubEval}(pk, x, w))$.

4.5.2 基于可公开求值伪随机函数的 KEM 构造

本章将展示如何基于 PEPRF 构造 KEM.

构造 4.17 (基于 PEPRF 的 KEM 构造)

构造思路: 随机采样语言中的元素作为密文, 计算其函数值作为会话密钥 k.

起点: PEPRF $F : SK \times X \to Y \cup \perp$, 其中 $L \subseteq X$ 是定义在 X 上的 $\mathcal{NP}$ 语言.

构造如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp \leftarrow \mathsf{PEPRF.Setup}(1^\kappa)$, 其中密文空间 $C = X$, 会话密钥空间 $K = Y$.
- $\mathsf{KeyGen}(pp)$: 运行 $(pk, sk) \leftarrow \mathsf{PEPRF.KeyGen}(pp)$.
- $\mathsf{Encaps}(pk; r)$: 随机采样 $(x, w) \leftarrow \mathsf{SampRel}(r)$, 输出 $c = x$ 作为密文, 通过 $\mathsf{PEPRF.PubEval}(pk, x, w)$ 公开计算 $k \leftarrow F_{sk}(x)$ 作为会话密钥.
- $\mathsf{Decaps}(sk, c)$: 通过运行 $\mathsf{PEPRF.PrivEval}(sk, c)$ 秘密计算 $k \leftarrow F_{sk}(x)$ 恢复会话密钥.

♣

构造 4.17的正确性由 PEPRF 的正确性保证, 安全性由以下定理保证.

定理 4.16

如果 PEPRF 是弱伪随机的, 则构造 4.17是 IND-CPA 安全的; 如果 PEPRF 是自适应弱伪随机的, 则构造 4.17是 IND-CCA 安全的.

♡

证明 IND-CPA 安全性的归约是显然的, 建立 IND-CCA 安全性的关键是令归约算法利用 $\mathcal{O}_{\mathsf{eval}}$ 模拟 $\mathcal{O}_{\mathsf{decaps}}$. □

注记 4.18

在上述的 KEM 构造中, 可以将 PEPRF 弱化为 PSPRF.

♠

4.5.3 可公开求值伪随机函数的构造

天下同归而殊途, 一致而百虑.

—《周易 · 系辞下》

本章节展示如何基于具体的困难假设和 (半) 通用的密码组件构造 PEPRF.

基于 DDH 假设的 PEPRF 构造

图 4.34展示了基于 DDH 假设的 PEPRF 构造, 其中可公开求值功能利用了 DH 函数的可交换性, 弱伪随机性建立在 DDH 假设之上. 将实例化代入构造 4.17中, 得到的正是经典的 ElGamal PKE 方案 [37].

构造 4.18 (基于 DDH 假设的 PEPRF)

- $\mathsf{Setup}(1^\kappa)$: 运行 $(\mathbb{G}, q, g) \leftarrow \mathsf{GenGroup}(1^\kappa)$, 生成公开参数 $pp = (F, PK, SK, X, L, W, Y)$, 其中 $X = Y = PK = L = \mathbb{G}$, $SK = W = \mathbb{Z}_q$, $F : SK \times X \to Y$ 定义为 $F_{sk}(x) = x^{sk}$, 语言 $L = \{x : \exists w \in W \text{ s.t. } x = g^w\}$, 相应的采样算法 $\mathsf{SampRel}$ 以随机数 r 为输入, 随机采样证据 $w \xleftarrow{R} \mathbb{Z}_q$, 计算实例 $x = g^w$.
- $\mathsf{KeyGen}(pp)$: 随机采样私钥 $sk \xleftarrow{R} \mathbb{Z}_q$, 计算公钥 $pk = g^{sk}$.
- $\mathsf{PrivEval}(sk, x)$: 输出 x^{sk}.
- $\mathsf{PubEval}(pk, x, w)$: 输出 pk^w.

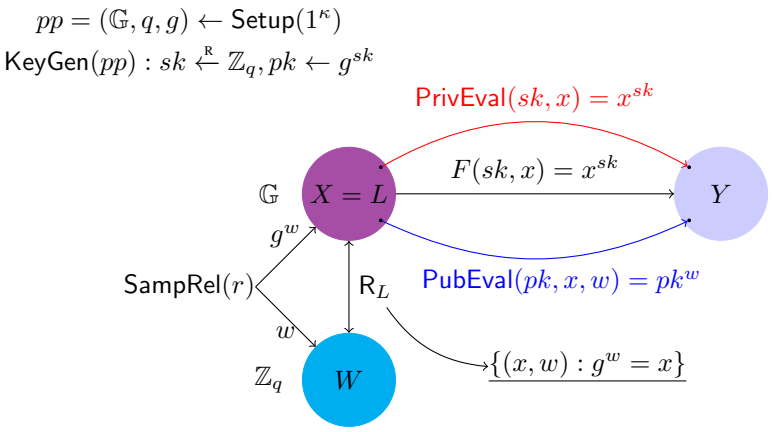


图 4.34: 基于 DDH 假设的 PEPRF

基于 QR 假设的 PEPRF 构造

图 4.35展示了基于 QR 假设的 PEPRF 构造, 其中可公开求值功能利用了语言 L 的 OR 型定义, 弱伪随机性建立在 QR 假设之上. 将实例化代入构造 4.17中, 得到的正是 Goldwasser-Micali PKE 方案 [236] 内蕴的 KEM.

构造 4.19 (QR-based-PEPRF)

- $\mathsf{Setup}(1^\kappa)$: 输出 $pp = \kappa$.
- $\mathsf{KeyGen}(pp)$: 运行 $(Nq) \leftarrow \mathsf{GenModulus}(1^\kappa)$, 选取 $z \in \mathbb{QNR}_N^{+1}$, 输出公钥 $pk = (N, z)$ 和私钥 $sk = (p, q)$. pk 还包含了以下信息: 函数定义域 $X = \mathbb{Z}_N^*$, 值域 $Y = \{0, 1\}$, 证据集合 $W = \mathbb{Z}_N^*$, 语言 $L_{pk} = \{x : \exists w \in W \text{ s.t. } x = w^2 \bmod N \vee x = zw^2 \bmod N\}$($\mathbb{Z}_N^*$ 中 Jacobi 符号为 $+1$ 的元素). 采样算法 $\mathsf{SampRel}$ 以公钥 pk 和随机数 r 为输入, 随机采样 $w \xleftarrow{R} \mathbb{Z}_q$, 随机生成实例 $x = w^2 \bmod N$ 或 $x = zw^2 \bmod N$.

- $\mathsf{PrivEval}(sk, x)$: 如果 $x \in \mathbb{QR}_N$ 则输出 1, 如果 $x \in \mathbb{QNR}_N^{+1}$ 则输出 0.
- $\mathsf{PubEval}(pk, x, w)$: 如果 $x = w^2 \bmod N$ 则输出 1, 如果 $x = zw^2 \bmod N$ 则输出 0.

♣

$$pp = \kappa \leftarrow \mathsf{Setup}(1^\kappa)$$
$$\mathsf{KeyGen}(pp) : sk \leftarrow (p, q), pk \leftarrow (N, z \xleftarrow{R} \mathbb{QNR}_N^{+1})$$

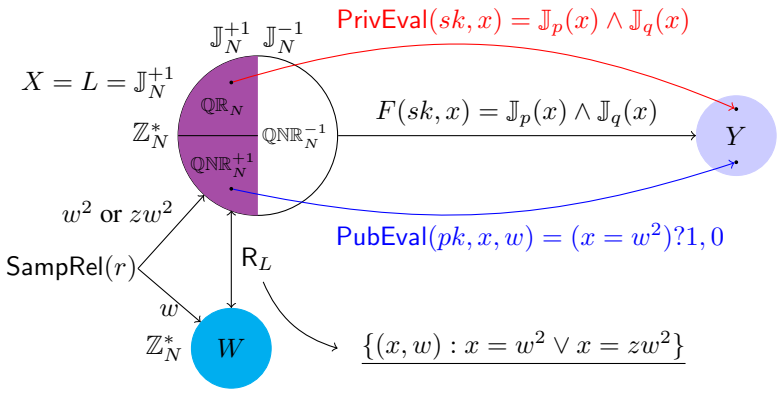


图 4.35: 基于 QR 假设的 PEPRF 构造

基于 TDF 的 PEPRF 构造

通过扭转单射 TDF, 可以构造 PEPRF 如下.

构造 4.20 (基于 TDF 的 PEPRF 构造)

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp = (G, EK, TD, S, U) \leftarrow \mathsf{TDF.Setup}(1^\kappa)$, 令 $\mathsf{hc} : S \to K$ 是相应的 hardcore function; 生成 PEPRF 的公开参数 $pp = (F, PK, SK, X, L, W, Y)$, 其中 $PK = EK$, $SK = TD$, $Y = K$, $X = U$, $W = S$, $F_{sk}(x) = \mathsf{hc}(G_{td}^{-1}(x))$. 算法 TDF.Eval 自然定义了一族定义在 X 上的 $\mathcal{NP}$ 语言 $L = \{L_{pk}\}_{pk \in PK}$, 其中 $L_{pk} = \{x : \exists w \in W \text{ s.t. } x = \mathsf{TDF.Eval}(pk, w)\}$. 采样算法 SampRel 以随机数 r 为输入, 首先随机采样定义域中元素 $s \leftarrow \mathsf{SampDom}(r)$, 再计算 $u \leftarrow \mathsf{TDF.Eval}(pk, s)$, 输出实例 $x = u$ 和证据 $w = s$.
- $\mathsf{KeyGen}(pp)$: 运行 $(ek, td) \leftarrow \mathsf{TDF.KeyGen}(pp)$, 输出 $pk = ek$ 和 $sk = td$.
- $\mathsf{PrivEval}(sk, x)$: 以私钥 sk 和元素 $x \in X$ 为输入, 输出 $y \leftarrow F_{sk}(x) = \mathsf{hc}(\mathsf{TDF.TdInv}(sk, x))$.
- $\mathsf{PubEval}(pk, x, w)$: 以公钥 pk、实例 $x \in L_{pk}$ 和证据 w 为输入, 输出 $y \leftarrow \mathsf{hc}(w)$.

♣

$$pp \leftarrow \mathsf{Setup}(1^\kappa)$$
$$(ek, td) \leftarrow \mathsf{KeyGen}(pp)$$

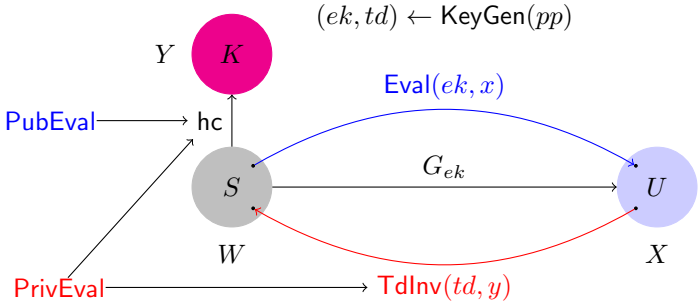


图 4.36: 基于 TDF 的 PEPRF 构造

构造 4.20的正确性由陷门单向函数的正确性和单射性保证, 安全性由如下定理保证.

	HPS	PEPRF
投射性	✓	not necessary
L 与 X 的关系	$L \subset X$	$L \subseteq X$
弱伪随机性	$x \xleftarrow{R} X \backslash L$	$x \xleftarrow{R} L$

表 4.2: HPS 与 PEPRF 的不同

定理 4.17

如果 TDF 是 (自适应) 单向的, 那么构造 4.20 中的 PEPRF 是 (自适应) 弱伪随机的.

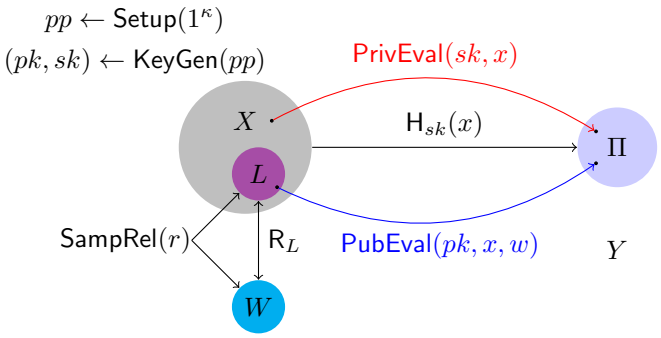
基于 HPS 的 PEPRF 构造

本章节展示如何基于哈希证明系统构造具有不同安全性质的可公开求值伪随机函数.

首先展示如何基于平滑 HPS 构造弱伪随机的 PEPRF.

构造 4.21 (基于平滑 HPS 的 PEPRF 构造)

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{HPS.Setup}(1^\kappa)$ 生成 HPS 的公开参数 $pp = (\mathsf{H}, PK, SK, X, L, W, \Pi, \alpha)$, 输入 PEPRF 的公开参数 $pp = (F, PK, SK, X, L, W, Y)$, 其中 $F = \mathsf{H}, Y = \Pi$.
- $\mathsf{KeyGen}(pp)$: 运行 $(pk, sk) \leftarrow \mathsf{HPS.KeyGen}(pp)$ 生成密钥对.
- $\mathsf{PrivEval}(sk, x)$: 以私钥 sk 和元素 $x \in X$ 为输入, 计算 $y \leftarrow \mathsf{HPS.PrivEval}(sk, x)$.
- $\mathsf{PubEval}(pk, x, w)$: 以公钥 pk、语言中的元素 $x \in L$ 和相应的证据 $w \in W$ 为输入, 计算 $y \leftarrow \mathsf{HPS.PubEval}(pk, x, w)$.



构造 4.21的正确性由平滑 HPS 的正确性保证, 安全性由如下定理保证:

定理 4.18

基于 $L \subset X$ 上的 SMP 假设, 构造 4.21 中的 PEPRF 满足弱伪随机性.

PEPRF 与 HPS 在语法上非常相似, 但存在以下微妙的不同, 如表 4.2所示:

下面展示如何基于平滑和一致 HPS 构造自适应伪随机的 PEPRF.

构造 4.22 (基于平滑和一致 HPS 的 PEPRF 构造)

构造组件: 针对同一语言 $\tilde{L} \subset \tilde{X}$ 的 smooth HPS_1 和 HPS_2:

构造如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp_1 = (\mathsf{H}_1, PK_1, SK_1, \tilde{X}, \tilde{L}, W, \Pi_1, \alpha_1) \leftarrow \mathsf{HPS}_1.\mathsf{Setup}(1^\kappa)$ 生成 smooth HPS 的公开参数, 运行 $pp_2 = (\mathsf{H}_2, PK_2, SK_2, \tilde{X}, \tilde{L}, \tilde{W}, \Pi_2, \alpha_2) \leftarrow \mathsf{HPS}_2.\mathsf{Setup}(1^\kappa)$ 生成 universal$_2$ HPS 的公开参数, 基于 pp_1 和 pp_2 生成 PEPRF 的公开参数 $pp = (F, PK, SK, X, L, W, Y)$, 其中 $X = \tilde{X} \times \Pi_2$, $Y = \Pi_1 \cup \perp$, $PK = PK_1 \times PK_2$, $L = \{L_{pk}\}_{pk \in PK}$ 定义在 $X = \tilde{X} \times \Pi_2$ 上, 其中 $L_{pk} = \{x =$

$(\tilde{x},\pi_2) : \exists w \in W \text{ s.t. } \tilde{x} \in \tilde{L} \wedge \pi_2 = \mathsf{HPS}_2.\mathsf{PubEval}(pk_2,\tilde{x},w)\}$, 相应的采样算法 SampRel 以公钥 $pk = (pk_1,pk_2)$ 和随机数 r 为输入, 首先随机采样语言 $\tilde{L}$ 的随机实例证据元组 $(\tilde{x},w)$, 计算 $\pi_2 \leftarrow \mathsf{HPS}_2.\mathsf{PubEval}(pk_2,\tilde{x},\tilde{w})$, 输出语言 L 的实例 $x = (\tilde{x},\pi_2)$ 和证据 $w = \tilde{w}$. 不失一般性, 令 pp 包含 pp_1 和 pp_2 中的所有信息.

- $\mathsf{KeyGen}(pp)$: 从 pp 中解析出 pp_1 和 pp_2, 运行 $(pk_1,sk_1) \leftarrow \mathsf{HPS}_1.\mathsf{KeyGen}(pp_1)$ 和 $(pk_2,sk_) \leftarrow \mathsf{HPS}_2.\mathsf{KeyGen}(pp_2)$, 输出 $pk = (pk_1,pk_2)$ 和 $sk = (sk_1,sk_2)$.
- $\mathsf{PrivEval}(sk,x)$: 以私钥 $sk = (sk_1,sk_2)$ 和 $x = (\tilde{x},\pi_2)$ 为输入, 如果 $\pi_2 = \mathsf{HPS}_2.\mathsf{PrivEval}(sk_2,\tilde{x})$ 则返回 $\perp$ 否则返回 $y \leftarrow \mathsf{HPS}_1.\mathsf{PrivEval}(sk_1,\tilde{x})$. 该算法定义了 $F : SK \times X \to Y \cup \perp$.
- $\mathsf{PubEval}(pk,x,w)$: 以公钥 $pk = (pk_1,pk_2)$、元素 $x = (\tilde{x},\pi_2) \in L_{pk}$ 以及证据 w 为输入, 输出 $y \leftarrow \mathsf{HPS}_1.\mathsf{PubEval}(pk_1,\tilde{x},w)$.

♣

定理 4.19

基于 $\tilde{L} \subset \tilde{X}$ 上的 SMP 假设, 构造 4.22中的 PEPRF 是自适应弱伪随机的.

♡

注记 4.19

构造 4.21相对直接, 构造 4.22稍显复杂, 其中蕴含的设计思想与基于哈希证明系统构造 CCA 安全的 PKE 相似: 使用 "弱"HPS 封装随机会话密钥, 使用 "强"HPS 生成证明以杜绝 "危险" 解密询问.

♠

基于 EHPS 的 PEPRF 构造

本章节展示如何基于 (ABO-)EHPS 构造 PEPRF.

构造 4.23 (基于 (ABO-)EHPS 的 PEPRF 构造)

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp = (\mathsf{H},PK,SK,\tilde{L},\tilde{W},\Pi) \leftarrow \mathsf{EHPS.Setup}(1^\kappa)$ 生成 EHPS 的公开参数, 令 $\mathsf{hc} : \tilde{W} \to Z$ 是单向关系 $\mathsf{R}_{\tilde{L}}$ 的 hardcore function; 生成 PEPRF 的公开参数 $pp = (F,PK,SK,X,L,W,Y)$, 其中 $X = \tilde{L} \times \Pi, Y = Z, W = R, L = \{L_{pk}\}_{pk \in PK}$ 定义在 $X = \tilde{L} \times \Pi$ 上, 其中 $L_{pk} = \{x = (\tilde{x},\pi) : \exists w \in W \text{ s.t. } \tilde{x} = \mathsf{SampYes}(w) \wedge \pi = \mathsf{EHPS.PubEval}(pk,\tilde{x},w)\}$, 相应的采样算法以公钥 pk 和随机数 w 为输入, 首先以 w 作为证据生成实例 $\tilde{x} \xleftarrow{R} \tilde{L}$, 再计算 $\pi \leftarrow \mathsf{EHPS.PubEval}(pk,\tilde{x},w)$, 输出实例 $x = (\tilde{x},\pi)$ 和证据 w. $F_{sk}(x) := \mathsf{hc}(\mathsf{EHPS.Ext}(sk,x))$.
- $\mathsf{KeyGen}(pp)$: 运行 $(pk,sk) \leftarrow \mathsf{EHPS.KeyGen}(pp)$ 生成密钥对.
- $\mathsf{PrivEval}(sk,x)$: 以私钥 sk 和 $x \in X$ 为输入, 将 x 解析为 $(\tilde{x},\pi)$, 计算 $\tilde{w} \leftarrow \mathsf{EHPS.Ext}(sk,\tilde{x},\pi)$, 输出 $y \leftarrow \mathsf{hc}(\tilde{w})$.
- $\mathsf{PubEval}(pk,x,w)$: 以公钥 pk、$x \in L_{pk}$ 以及相应的证据 w 为输入, 计算 $\tilde{w} \leftarrow \mathsf{SampWit}(w)$, 输出 $y \leftarrow \mathsf{hc}(\tilde{w})$.

♣

定理 4.20

如果 $\mathsf{R}_{\tilde{L}}$ 是单向的, 基于 (ABO-)EHPS 的 PEPRF 是 (自适应) 弱伪随机的.

♡

基于程序混淆的 PEPRF 构造

本章节展示如何基于不可区分程序混淆构造 PEPRF.

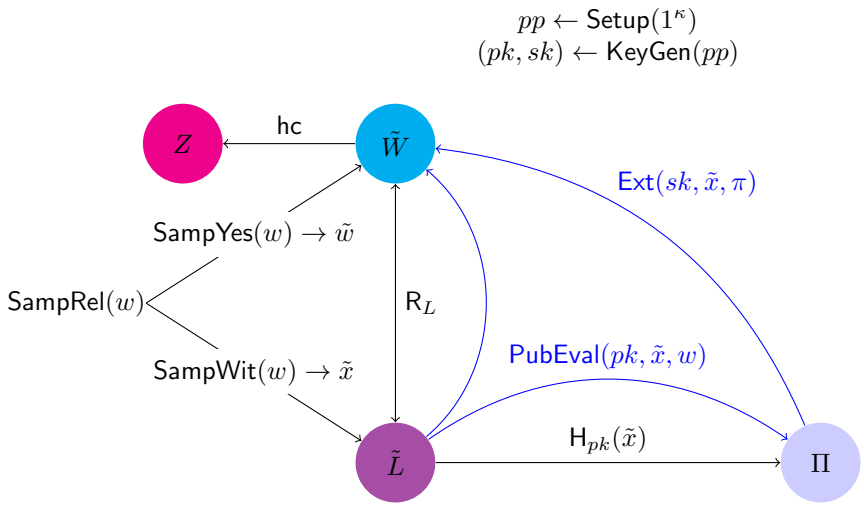


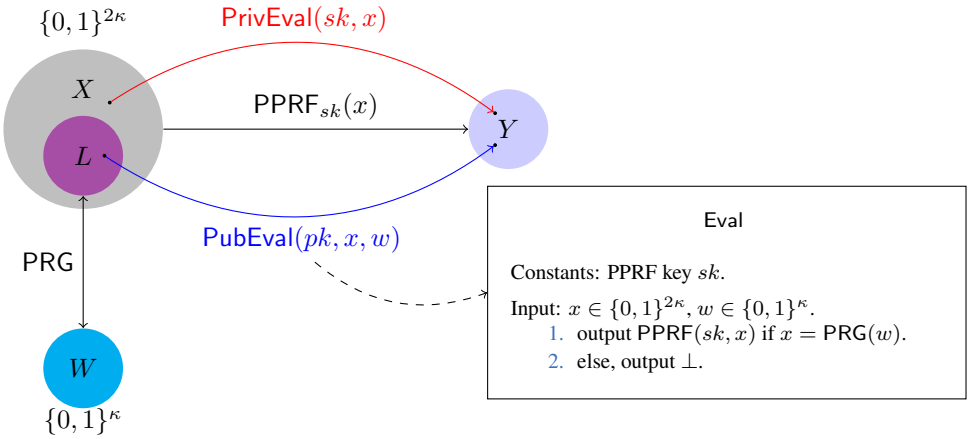
图 4.37: 基于 EHPS 的 PEPRF 构造

构造 4.24 (基于 $i\mathcal{O}$ 和 PPRF 的 PEPRF 构造)

构造组件: 不可区分程序混淆 $i\mathcal{O}$、伪随机数发生器和可穿孔伪随机函数

构造如下:

- $\mathsf{Setup}(1^\kappa)$: 生成对电路族 $\mathcal{C}_\kappa$ 的不可区分程序混淆 $i\mathcal{O}$, 选取长度倍增的伪随机数发生器 $\mathsf{PRG} : \{0,1\}^\kappa \to \{0,1\}^{2\kappa}$ 和可穿孔伪随机函数 $\mathsf{PPRF} : K \times \{0,1\}^{2\kappa} \to Y$; 生成 PEPRF 的公开参数 $pp = (F, PK, SK, X, L, W, Y)$, 其中 $PK = i\mathcal{O}(\mathcal{C}_\kappa)$, $SK = K$, $X = \{0,1\}^{2\kappa}$, 其中 $F := \mathsf{PPRF}$, $W = \{0,1\}^\kappa$, $L = \{x \in X : \exists w \in W \text{ s.t. } x = \mathsf{PRG}(w)\}$, 相应的采样算法 $\mathsf{SampRel}$ 以随机数 $r \in \{0,1\}^\kappa$ 为输入, 输出实例 $x \leftarrow \mathsf{PRG}(r)$ 和证据 $w = r$.
- $\mathsf{KeyGen}(pp)$: 随机采样 $k \in K$ 作为私钥 sk, 计算 $pk \leftarrow i\mathcal{O}(\mathsf{Eval})$ 作为公钥.
- $\mathsf{PrivEval}(sk, x)$: 输出 $y \leftarrow \mathsf{PPRF}(sk, x)$.
- $\mathsf{PubEval}(pk, x, w)$: 将公钥 pk 解析为程序, 计算 $y \leftarrow pk(x, w)$.



定理 4.21

基于不可区分程序混淆、伪随机数发生器和可穿孔伪随机函数的安全性, 构造 4.24 中的 PEPRF 满足自适应弱伪随机性.

注记 4.20

上述构造实质上展示了 $i\mathcal{O}$ 可以将 Minicrypt 中的可穿孔伪随机函数编译为 Cryptomania 中的可公开求值伪随机函数.

小结

本章中引入了 PEPRF 这一全新的密码组件, 展示了它与已有密码组件之间的联系以及它的应用, 如图 4.38所示. 引入 PEPRF 最大的理论意义在于它不仅首次阐明了经典的 Goldwasser-Micali PKE 和 ElGamal PKE 的构造机理, 还统一了几乎所有已知的构造范式. 作为首个实用的公钥加密, RSA PKE 影响深远, 令单向陷门函数的概念深入人心, 使得人们常有 “构造公钥加密必须有陷门” 的错觉. PEPRF 树立了正确的认知, 指出构造公钥加密的实质在于构造可公开求值的伪随机函数, 核心技术是 “令同一函数存在两种求值方法”. 基于 PEPRF 的 PKE 构造恰与 Minicrypt 中基于 PRF 的 SKE 构造形成完美的形式契合与思想共鸣.

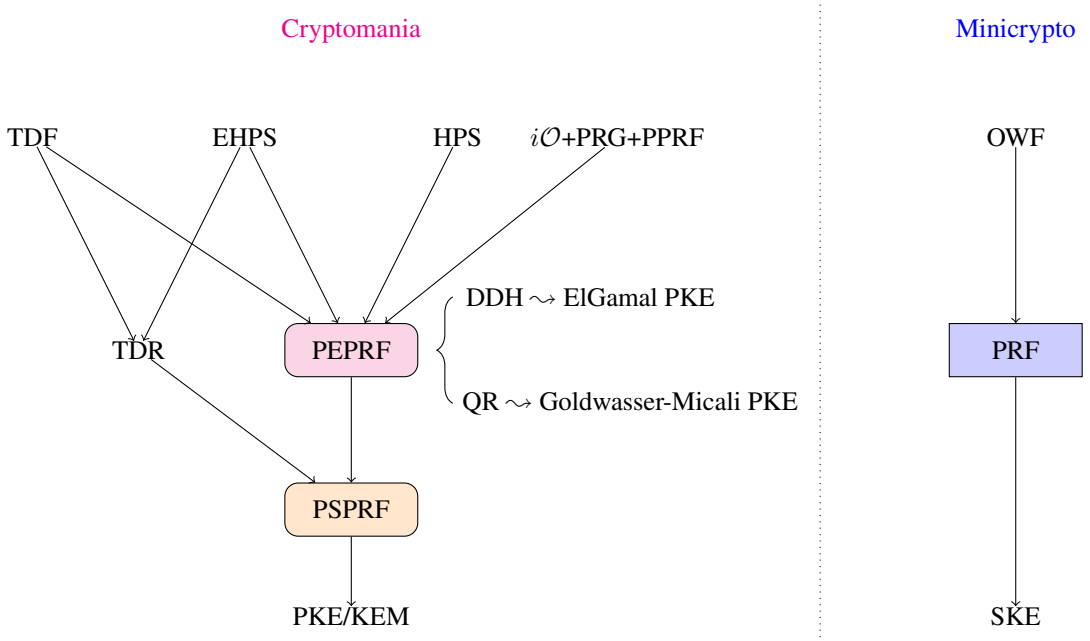


图 4.38: PEPRF 的构造与应用

PEPRF 的强大威力来源于其高度抽象, 它诠释了公钥加密设计的 “万法同源, 殊途同归”.

笔记 抽象的概念是美妙的, 相信读者能够通过 PEPRF 感受到 “大繁至简” 的优雅与 “高屋建瓴” 的力量. 然而抽象概念是果, 具体构造是因. 切不能刻意过度的抽象而忽视具体构造, 正是多种多样具体构造才让我们能够有机缘洞见事物本质, 使得高度凝练的概念内涵丰富、意义深刻.

第五章 公钥加密的安全性增强

章前概述

内容提要

- 抗泄漏安全
- 抗篡改安全
- 消息依赖密钥安全

本章开始介绍公钥加密的安全性增强方法. 5.1节介绍了抗泄漏公钥加密的基本概念、安全模型和基于哈希证明系统、一次有损过滤器、规则有损函数等技术的通用构造方法, 5.2节介绍了抗篡改公钥加密的基本概念、安全模型和基于自适应单向陷门关系、不可延展函数等技术的通用构造方法, 5.3节介绍了消息依赖公钥加密的基本概念、安全模型和基于同态哈希证明系统的通用构造方法.

5.1 抗泄漏安全

夫事以密成, 语以泄败.

—《韩非子 · 说难第十二》

侧信道攻击, 又称边信道攻击或旁路攻击, 利用密码算法在实现过程中泄漏的物理信息, 如运行时间 [303]、电磁辐射 [304]、能量功耗 [305] 等来攻击密码算法的安全性. 图 5.1展示了侧信道攻击的方法, 其中, F 代表一种密码算法函数, 如签名算法、解密算法等. 除了私钥 sk 外, 算法的输入可能还包括签名消息或解密密文 x. 敌手可以利用上述侧信道攻击方法在算法 F 执行过程中获取私钥 sk 的部分信息, 记作 $\mathsf{leak}(sk)$. 2008 年, Halderman 等 [306] 还发现另一类特殊的侧信道攻击方法, 即 "冷启动" 攻击 (又称内存攻击). 简单来说, 计算机断电后内存中存储的信息并不是立即被擦除掉, 通过短暂的物理访问可以恢复动态随机存取存储器中的数据或密钥. 上述这些物理信息都有可能泄漏私钥的部分信息, 因此这类侧信道攻击统称为密钥泄漏攻击. 与传统的数学分析方法相比, 这类新型分析技术更有效, 对密码算法的安全性构成巨大的威胁. 早期抵御侧信道攻击的方法主要通过在算法实现过程中引入一些随机信息以减少泄漏的物理信息中含有的密钥信息, 可参考文献 [307] 第 29 章及其引文. 然而这种方式难以同时抵抗多种类型的侧信道攻击技术并且这些方法缺少严格的安全性证明. 类似不可区分选择密文攻击模型, 如何建立合理的抗密钥泄漏攻击的安全模型, 并从算法角度设计可证明安全的抗密钥泄漏密码方案是抗泄漏密码学研究的主要问题.

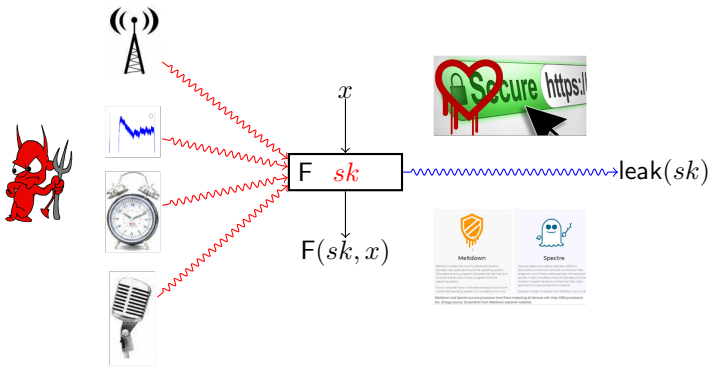


图 5.1: 侧信道攻击方法示意图

根据泄漏函数的形式不同, 密钥泄漏模型可以分为有界密钥泄漏模型和无界密钥泄漏模型. 2009 年, Akavia、Goldwasser 和 Vaikuntanathan [66] 受 "冷启动" 攻击的启发, 提出一种非常实用的刻画密钥泄漏的模型. 该模型允许攻击者获得密钥的任意信息, 只要所有获取的信息比特长度不超过某一阈值 ℓ 即可. 因此, 该模型一般称为有界密钥泄漏模型 (bounded-leakage model, BLM). 具体来讲, 攻击者可以通过一系列有效、可计算函数 f, 称之为泄漏函数, 适应性地访问私钥 sk, 并获取相应的泄漏信息 $f(sk)$, 而对攻击者的要求是所有泄漏函数的输出长度之和不超过该阈值. 由于 BLM 模型既简单又能涵盖广泛的侧信道攻击方法, 近年来, 该模型受到了密码学界的广泛关注. 特别地, 有界密钥泄漏模型可以涵盖以下两种密钥泄漏情形: 相对泄漏 (relative leakage) 和绝对泄漏 (absolute leakage).

- 相对泄漏: 总体泄漏量与私钥长度的比率是相对固定的. 这一比率通常称为相对泄漏比率. 例如, 攻击者得到的泄漏信息长度不超过私钥长度的一半. 相对泄漏能够刻画多种侧信道攻击的情景, 包括 Halderman 等的 "冷启动" 攻击、针对智能卡的微波攻击等. 因此, 很多抗密钥泄漏方案都是在相对泄漏模型下设计的.
- 绝对泄漏: 相对泄漏量可以非常巨大. 这种模型在一些场合是非常实用的. 例如, 当系统中存有恶意软件时, 病毒程序可能会将用户大量的敏感数据传送给远程的控制服务器. 但是在很多情况下, 病毒程序下载巨量数据消耗的时间和代价很大. 因此, 抵御这种类型侧信道攻击最好的方法是将私钥变得巨大, 以至于攻击者无法获取超过阈值的信息量. Crescenzo 等 [308] 和 Dziembowski [309] 将这一模型称为有界恢复模型 (bounded retrieval model, BRM). 在有界恢复模型中, 设计密码算法的基本方式是通过增加敏感数据的存储

空间来实现安全性, 但是不能影响系统其他方面的性能. 特别地, 合法用户仅需要访问很小一部分的密钥信息, 而他的计算和通信开销并不会有太大的增加.

有界恢复模型可以看做是内存泄漏模型的推广. 在 BRM 模型中设计方案的困难性主要在于方案效率仅能依赖方案的安全参数, 而不能依赖私钥的大小. 在相对泄漏模型中, 方案的效率一般与私钥大小有关. 通过扩大私钥空间来提高私钥泄漏量的同时, 方案的效率往往会显著下降. 尽管如此, 在 BRM 模型中设计方案通常先在相对泄漏模型中进行设计. 为此, 本节重点介绍相对泄漏模型下的抗泄漏公钥加密方案的几种典型设计方法.

5.1.1 抗泄漏安全模型

本节主要介绍公钥加密方案在相对密钥泄漏攻击和自适应选择密文攻击下的安全模型, 简称 ℓ–LR-CCA 安全模型, 其中 ℓ 表示密钥泄漏量的上界. 在该模型中, 敌手不仅可以访问解密谕言机, 而且可以获得密钥的部分信息. 密钥泄漏查询由任意一组输出长度之和不超过泄漏上界 ℓ 的函数组成. 敌手可以适应性地选择函数 f 并获得密钥的函数值 $f(sk)$. 很显然, 如果函数 f 的输出没有任何限制, 则任何 (公钥) 加密方案都不可能抵抗密钥泄漏攻击.

LR-CCA 安全性. 定义公钥加密方案敌手 $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa)=\left|\Pr\left[\beta'=\beta:\begin{array}{l} pp\leftarrow\mathsf{Setup}(1^{\kappa});\\ (pk,sk)\leftarrow\mathsf{KeyGen}(pp);\\ (m_0,m_1,state)\leftarrow\mathcal{A}_1^{\mathcal{O}_{\mathsf{decrypt}},\mathcal{O}_{\mathsf{leak}}}(pp,pk);\\ \beta\xleftarrow{\mathrm{R}}\{0,1\};\\ c^*\leftarrow\mathsf{Encrypt}(pk,m_\beta);\\ \beta'\leftarrow\mathcal{A}_2^{\mathcal{O}_{\mathsf{decrypt}}}(pp,pk,state,c^*);\end{array}\right]-\frac{1}{2}\right|$$

在上述定义中, $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ 的含义类似 IND-CCA 安全模型中的敌手, 表示敌手 $\mathcal{A}$ 可划分为两个阶段, 划分界线是接收到挑战密文 c^* 前后, $state$ 表示 $\mathcal{A}_1$ 向 $\mathcal{A}_2$ 传递的信息, 记录部分攻击进展. $\mathcal{O}_{\mathsf{decrypt}}$ 表示解密谕言机, 其在接收到密文 c 的询问后输出 $\mathsf{Decrypt}(sk,c)$. $\mathcal{O}_{\mathsf{leak}}$ 表示密钥泄漏谕言机, 其在接收到泄漏函数 $f_i:\{0,1\}^*\to\{0,1\}^{\ell_i}$ 的询问后输出 $f_i(sk)$ 且所有泄漏函数输出长度之和满足 $\sum_i\ell_i\le\ell$. 如果任意的 PPT 敌手 $\mathcal{A}$ 在上述游戏中的优势函数 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 均为可忽略函数, 则称公钥加密方案是 ℓ-LR-CCA 安全的. 如果不允许敌手访问解密谕言机, 则称公钥加密方案是 ℓ-LR-CPA 安全的.

笔记 在 LR-CCA 模型中, 敌手在获得挑战密文后是不允许再访问密钥泄漏谕言机的. 否则, 敌手可以将挑战密文的解密函数作为一种特殊的密钥泄漏函数, 通过访问密钥泄漏谕言机获得明文的部分比特信息, 从而区分挑战密文加密的是 m_0 还是 m_1. 如果允许敌手在看到挑战密文后继续访问密钥泄漏函数, 则模型的安全目标必然会降低. 为此, 2011 年 Halevi 和 Lin [310] 提出"after-the-fact"密钥泄漏模型, 利用明文的剩余熵来刻画方案的安全性. 在上述定义中, 若令 $\ell=0$, 即敌手没有访问密钥泄漏谕言机, 则上述定义即是标准 IND-CCA 安全性的定义. 图 5.2展示了 PKE 的抗泄漏模型与传统安全模型之间的关系.

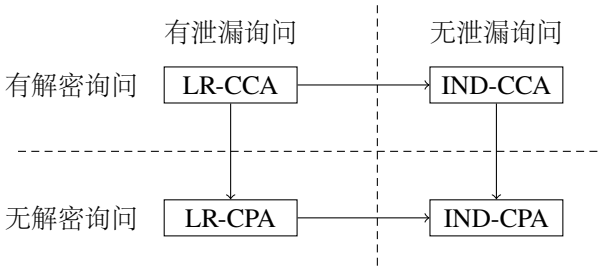


图 5.2: 公钥加密的抗泄漏模型与传统安全模型之间的关系

5.1.2 LR-CPA 安全 PKE 的通用构造方法

5.1.2.1 基于哈希证明系统的 LR-CPA 安全 PKE

2009 年, Naor 和 Segev [67] 基于哈希证明系统提出一种抗泄漏 PKE 方案的通用构造. 该构造结构简单, 是哈希证明系统在抗泄漏密码学中的一个经典应用案例.

构造 5.1 (基于 HPS 的 LR-CPA 安全 PKE)

令 $\ell = \ell(\kappa)$ 为密钥泄漏量的上界, ϵ_1 和 ϵ_2 是两个可忽略量. 构造所需组件是:

- ϵ_1-universal$_1$ 哈希证明系统 $\mathsf{HPS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PubEval}, \mathsf{PrivEval})$.
- 一个平均意义 $(\log \Pi - \ell, \epsilon_2)$-强随机性提取器 $\mathsf{ext} : \Pi \times \{0,1\}^d \to \{0,1\}^\kappa$.

构造 PKE 如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{HPS.Setup}(1^\kappa)$, 输出 HPS 的一个实例参数 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$. 选择一个平均意义 $(\log \Pi - \ell, \epsilon_2)$-强随机性提取器 $\mathsf{ext} : \Pi \times \{0,1\}^d \to \{0,1\}^\kappa$. 将 $\hat{pp} = (pp, \mathsf{ext})$ 作为公开参数, 其中 $\{0,1\}^\kappa$ 作为明文空间 M, $X \times \{0,1\}^d \times \{0,1\}^\kappa$ 作为密文空间 C.
- $\mathsf{KeyGen}(pp)$: 运行 $(pk, sk) \leftarrow \mathsf{HPS.KeyGen}(pp)$, 输出公钥 pk 和私钥 sk.
- $\mathsf{Encrypt}(pk, m)$: 以公钥 pk 和明文 $m \in \{0,1\}^\kappa$ 为输入, 执行如下步骤:
  1. 运行 $(x, w) \leftarrow \mathsf{SampRel}(r)$ 生成随机实例和相应的证据, 其中 r 是采样算法使用的随机数;
  2. 通过 $\mathsf{HPS.PubEval}(pk, x, w)$ 计算实例 x 的哈希证明 $\pi \leftarrow \mathsf{H}_{sk}(x)$;
  3. 随机选择 $s \xleftarrow{R} \{0,1\}^d$, 计算 $\psi = \mathsf{ext}(\pi, s) \oplus m$;
  4. 输出 (x, s, ψ) 作为密文 c.
- $\mathsf{Decrypt}(sk, c)$: 以私钥 sk 和密文 $c = (x, s, \psi)$ 为输入, 通过 $\mathsf{HPS.PrivEval}(sk, x)$ 计算 x 的哈希证明 $\pi \leftarrow \mathsf{H}_{sk}(x)$, 再恢复明文 $m' = \psi \oplus \mathsf{ext}(\pi, s)$.

♣

正确性. 根据哈希证明系统的正确性, 即 $\mathsf{HPS.PrivEval}(sk, x) = \mathsf{HPS.PubEval}(pk, x, w) = \mathsf{H}_{sk}(x)$, 以下公式说明方案具有完美正确性:

$$\begin{aligned} m' &= \psi \oplus \mathsf{ext}(\mathsf{HPS.PrivEval}(sk, x), s) \\ &= \mathsf{ext}(\mathsf{HPS.PubEval}(pk, x, w), s) \oplus m \oplus \mathsf{ext}(\mathsf{HPS.PrivEval}(sk, x), s) \\ &= \mathsf{ext}(\mathsf{H}_{sk}(x), s) \oplus m \oplus \mathsf{ext}(\mathsf{H}_{sk}(x), s) \\ &= m \end{aligned}$$

安全性. 构造 5.1中的 PKE 方案的抗密钥泄漏安全性可由以下定理保证.

定理 5.1

如果 HPS 是一个 ϵ_1-universal$_1$ 哈希证明系统且 ext 是一个平均意义 $(\log \Pi - \ell, \epsilon_2)$-强随机性提取器, 那么构造 5.1是一个 ℓ-LR-CPA 安全的公钥加密方案, 其中 $\ell \le \log |\Pi| - \omega(\kappa) - \kappa$.

♡

笔记 Naor-Segev PKE 方案的设计思路和安全性证明思路几乎是完美统一的. 如图 5.3所示, 给定一个 HPS 公钥 pk, 存在若干个私钥 sk 满足 $\alpha(sk) = pk$, 记 $SK_{pk} = \{sk | \alpha(sk) = pk\}$ 表示与公钥 pk 对应的所有可能私钥组成的空间. 当 $x \in L$ 时, 哈希证明系统可以看作是一个从私钥空间 SK_{pk} 到哈希证明空间上的多对一投射函数, 即对于任意两个不同的私钥 $sk_0, sk_1 \in SK_{pk}$, 都有 $\mathsf{H}_{sk_0}(x) = \mathsf{H}_{sk_1}(x)$. 当 $x \in X \setminus L$ 时, 哈希证明系统可以看作是一个从私钥空间 SK_{pk} 到哈希证明空间 Π 上的一对一映射函数, 即对于任意两个不同的私钥 $sk_0, sk_1 \in SK_{pk}$, 则有 $\mathsf{H}_{sk_0}(x) \neq \mathsf{H}_{sk_1}(x)$. 基于子集成员判断问题困难性, 即使在知道私钥 sk 的情况下, 这两种映射函数在计算意义下也是不可区分的. 对于第二种情况, 在公钥确定的情况下, 封装的哈希证明 π 依然具有一定的信息熵. 当私钥 sk 泄漏部分信息时, 由于映射函数 H 是一一映射, 所以只会降低 π 的信息熵, 从而利用一个平均强随机性提取器 ext

依然可以提取出具有均匀随机性的比特串用于掩盖真实的消息 m. 这一证明思路同时解释了为什么 Naor-Segev 方案需要引入一个平均强随机性提取器来掩盖消息, 从而能够容忍密钥泄漏.

具体地, 定理 5.1可通过一系列不可区分游戏来实现. 在原始游戏的基础, 第一步, 利用 HPS 公开计算和私有计算两种模式的等价性可以将哈希证明的计算方式从公开计算模式转化为私有计算模式. 第二步, 利用子集成员判定问题的困难性, 可将随机实例 x 的采样从集合 L 转化为集合 $X \setminus L$. 第三步, 利用哈希证明 π 具有信息熵的性质, 也就说 HPS 的 universal$_1$ 性质, 进一步将 π 从私有计算转化为随机选取. 最后, 证明在私钥泄漏部分信息的情况下, 利用强随机性提取器 ext 的性质依然可以提取出均匀随机比特串, 从而掩盖消息 m_b 的信息, 实现密文不可区分性.

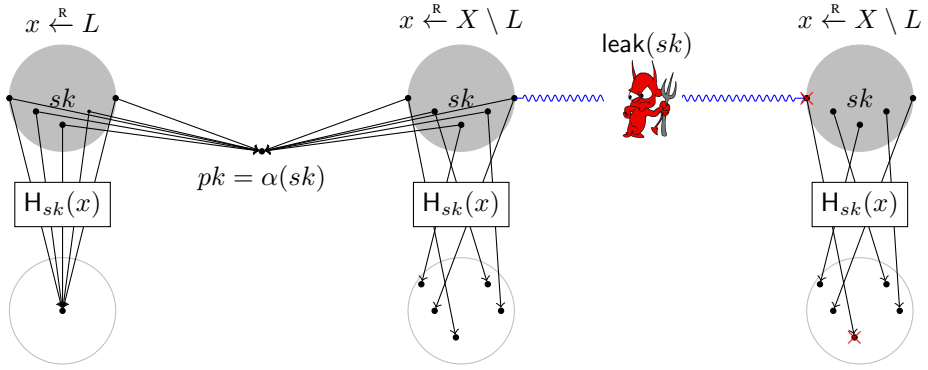


图 5.3: 基于 HPS 的 LR-CPA 安全 PKE 的设计和证明思路示意图

证明 令 S_i 表示事件 "$\mathcal{A}$ 在 Game$_i$ 中成功". 以游戏序列的方式组织证明如下:

Game$_0$: 该游戏是标准的 LR-CPA 游戏, 挑战者 $\mathcal{CH}$ 和敌手 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 运行 $\mathsf{Setup}(1^\kappa)$ 生成公开参数 $\hat{pp} = (pp, \mathsf{ext})$, 同时运行 $\mathsf{KeyGen}(pp)$ 生成公私钥对 (pk, sk). $\mathcal{CH}$ 将 $(\hat{pp}, pk)$ 发送给 $\mathcal{A}$.
- 询问: 假设 $f_i : \{0,1\}^* \to \{0,1\}^{\ell_i}$ 是 $\mathcal{A}$ 的第 i 次泄漏谕言机 $\mathcal{O}_{\mathsf{leak}}$ 查询. $\mathcal{CH}$ 首先判断 $\sum_i \ell_i \le \ell$ 是否成立, 若成立则返回 $f_i(sk)$, 否则返回 $\perp$.
- 挑战: $\mathcal{A}$ 选择 $m_0, m_1 \in \{0,1\}^\kappa$ 并发送给 $\mathcal{CH}$. $\mathcal{CH}$ 选择随机比特 $\beta \in \{0,1\}$, 作如下计算:
  1. 运行 $(x, w) \leftarrow \mathsf{SampRel}(r)$ 生成随机实例 x 及其证据 w;
  2. 通过 $\mathsf{HPS.PubEval}(pk, x, w)$ 计算实例 x 的哈希证明 $\pi \leftarrow \mathsf{H}_{sk}(x)$;
  3. 随机选择 $s \xleftarrow{R} \{0,1\}^d$, 计算 $\psi = \mathsf{ext}(\pi, s) \oplus m_b$;
  4. 输出 (x, s, ψ) 作为挑战密文 c^* 并发送给 $\mathcal{A}$.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. $\mathcal{A}$ 成功当且仅当 $\beta' = \beta$.

根据定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game$_1$: 该游戏与 Game$_0$ 的唯一不同在于挑战密文中哈希证明的生成方式. $\mathcal{CH}$ 不再通过 $\mathsf{HPS.PubEval}(pk, x, w)$ 计算哈希证明, 而是通过$\mathsf{HPS.PrivEval}(sk, x)$计算 x 的哈希证明 $\pi \leftarrow \mathsf{H}_{sk}(x)$. 根据 HPS 两种工作模式的等价性可知, 敌手 $\mathcal{A}$ 在游戏 Game$_0$ 和 Game$_1$ 中的视图是一样的, 则有:

$$\text{Game}_0 \equiv \text{Game}_1$$

Game$_2$: 该游戏与 Game$_1$ 的唯一不同在于挑战密文中随机实例 x 的选取方式. $\mathcal{CH}$ 调用 $\mathsf{SampNo}(pp)$ 采样$x \xleftarrow{R} X \setminus L$. 根据 SMP 问题的困难性, 敌手 $\mathcal{A}$ 在游戏 Game$_1$ 和 Game$_2$ 中的视图计算不可区分, 则有:

$$\text{Game}_1 \approx_c \text{Game}_2$$

Game$_3$: 该游戏与 Game$_2$ 的唯一不同在于挑战密文中哈希证明 π 的计算方式. $\mathcal{CH}$ 随机选取$\pi \xleftarrow{R} \Pi$. 根据 HPS 的

universal$_1$ 性质, 可以证明敌手 $\mathcal{A}$ 在游戏 Game_2 和 Game_3 中的视图统计上不可区分, 即:

$$\text{Game}_2 \approx_s \text{Game}_3$$

这是因为, 在没有任何密钥泄漏的情况下, 根据 HPS 的 ϵ_1-universal$_1$ 性质, 则有:

$$\Delta((pk, x, \mathsf{H}_{sk}(x)), (pk, x, \pi)) \leq \epsilon_1$$

令密钥泄漏谕言机输出的信息为 leak. 由于 leak 的分布由公钥 pk, 随机实例 x 和哈希证明 $\mathsf{H}_{sk}(x)$ 完全确定, 即 $\mathsf{leak} = \mathsf{leak}(pk, x, \mathsf{H}_{sk}(x))$, 根据统计距离的性质, 可得:

$$\Delta((pk, x, \mathsf{H}_{sk}(x), \mathsf{leak}(pk, x, \mathsf{H}_{sk}(x))), (pk, x, \pi, \mathsf{leak}(pk, x, \pi))) \leq \epsilon_1$$

由于强随机性提取器 ext 作用在上述两个分布上不会增加它们的统计距离, 故有:

$$\Delta((pk, x, \mathsf{ext}(\mathsf{H}_{sk}(x), s), s, \mathsf{leak}), (pk, x, \mathsf{ext}(\pi, s), s, \mathsf{leak})) \leq \epsilon_1$$

通过上述分析可知, 敌手 $\mathcal{A}$ 在上述两个游戏中的视图统计距离相差不超过 ϵ_1, 即 $\text{Game}_2 \approx_s \text{Game}_3$.

Game_4: 该游戏与 Game_3 的唯一不同在于挑战密文中强随机性提取器 $\mathsf{ext}(\pi, s)$ 的选取方式. $\mathcal{CH}$ 随机选择$k \xleftarrow{R} \{0,1\}^\kappa$, 再计算 $\psi = k \oplus m_b$. 由于 k 是随机且独立于消息 m_b 的选取的, 所以在该游戏中敌手没有任何优势猜测挑战消息, 即 $\mathcal{A}$ 在该游戏中成功的概率为:

$$\Pr[S_4] = 1/2$$

最后, 证明即使在泄漏 ℓ 比特密钥信息的情况下, 敌手在游戏 Game_3 和 Game_4 两个游戏中的视图仍然是不可区分的.

对于分布 $(pk, x, k, \mathsf{ext}(\pi, s), s, \mathsf{leak})$, ℓ 比特的密钥泄漏量 leak 最多使 π 的平均最小熵减少 ℓ, 即

$$\mathsf{H}_\infty(\pi | (pk, x, \mathsf{leak})) \geq \tilde{\mathsf{H}}_\infty(\pi | (pk, x)) - \ell = \log \Pi - \ell$$

利用平均强随机性提取器 ext 的性质, 可得:

$$\Delta((pk, x, \mathsf{ext}(\pi, s), s, \mathsf{leak}), (pk, x, k, s, \mathsf{leak})) \leq \epsilon_2$$

其中, $k \in \{0,1\}^\kappa$ 是独立且随机选取的. 由此可知, 敌手 $\mathcal{A}$ 在上述两个游戏中的视图统计距离相差不超过 ϵ_2.

综上, 定理 5.1得证. □

将第 4.2节关于 L_{DDH} 的 $\frac{1}{q}$-universal$_1$ 哈希证明系统 (构造 4.9) 应用于 Naor-Segev 的通用构造中, 即可得到一个基于 DDH 问题的 LR-CPA 安全的 PKE 方案.

构造 5.2 (基于 DDH 问题的 LR-CPA 安全 PKE)

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{GenGroup}(1^\kappa)$, 生成一个循环群 $(\mathbb{G}, q, g)$. 令 $\ell = \ell(\kappa)$ 是泄漏参数 (泄漏量上界). 选择一个平均意义 $(\log q - \ell, \epsilon)$-强随机性提取器 $\mathsf{ext} : \mathbb{G} \times \{0,1\}^d \to \{0,1\}^\kappa$. 输出公开参数 $pp = (\mathbb{G}, q, g, \mathsf{ext})$.
- $\mathsf{KeyGen}(pp)$: 随机选择 $x_1, x_2 \in \mathbb{Z}_q$ 和 $g_1, g_2 \in \mathbb{G}$ 并计算 $h = g_1^{x_1} g_2^{x_2}$. 输出公钥 $pk = (g_1, g_2, h)$ 和私钥 $sk = (x_1, x_2)$.
- $\mathsf{Encrypt}(pk, m)$: 输入公钥 pk 和明文 $m \in \{0,1\}^\kappa$, 随机选择 $r \in \mathbb{Z}_q$ 和 $s \in \{0,1\}^d$, 输出密文 $c = (g_1^r, g_2^r, s, \mathsf{ext}(h^r, s) \oplus m)$.
- $\mathsf{Decrypt}(sk, c)$: 输入私钥 $sk = (x_1, x_2)$ 和密文 $c = (u_1, u_2, s, e)$, 输出明文 $m' := e \oplus \mathsf{ext}(u_1^{x_1} u_2^{x_2}, s)$. ♣

根据定理 5.1, 可以直接得到下述结论:

引理 5.1

如果 DDH 假设成立, 那么构造 5.2中的 PKE 是 ℓ-LR-CPA 安全的, 其中 $\ell = (\log q - \omega(\log \kappa) - \kappa)$. ♡

笔记 由上述引理可知, 实例化方案的密钥泄漏量可达 $(1/2 - o(1))|sk|$, 其中 $|sk|$ 表示私钥的比特长度. 然而, 该方案仅是 LR-CPA 安全的. 为了实现 LR-CCA 安全性, 一种直接的方法是将 Naor-Yung"双密钥加密范式" 应用与一

个 ℓ-LR-CPA 安全的公钥加密方案上，从而得到一个密钥泄漏比率不变且抗选择密文攻击安全的 LR-CCA 安全公钥加密方案. 该方法需要引入适应性安全的非交互零知识证明系统，然而，标准模型下构造的非交互零知识证明系统在效率上具有一定的局限性. 另一种方法是结合一个 universal$_2$ HPS 实现 CCA 安全性，如图 5.4. 其中 HPS$_1$ 满足 universal$_1$ 性质，用于掩盖消息，而 HPS$_2$ 满足 universal$_2$ 性质，用于验证密文的合法性. 在加密阶段，当 $x \in L$ 时，只需要利用 HPS 的公开计算模式，而不需要输入 HPS 的私钥 sk_1 和 sk_2. HPS 的私钥仅在解密和安全性证明中使用. 由于任意一个 HPS 的私钥不可能完全泄漏，否则系统无任何安全性保障. 对于整个私钥 $sk = (sk_1, sk_2)$ 而言，能够容忍的密钥泄漏量不会超过 sk 的一半. 因此，在理论上，这种构造方式能够容忍的密钥泄漏比率不会超过 $1/2 - o(1)$. 实际上，基于该方法设计的方案容忍密钥泄漏的比率更小，如在文献 [67] 中，基于 DDH 问题设计的 CCA 安全 PKE 方案的密钥泄漏比率仅为 $1/6 - o(1)$，而在优化的设计方案 [311, 312] 中，密钥泄漏比率也只能提升至 $1/4 - o(1)$. 为了提升密钥泄漏比率，同时保证方案的效率，一种可行的方法是将第二个 HPS 替换为一种类似非交互零知识证明系统的无密钥、信息泄漏量少 (固定) 且高效的密码原语，如一次有损过滤器 [70, 91]、规则有损函数 [72, 313] 等. 在第 5.1.3节将介绍这种具体的密码原语.

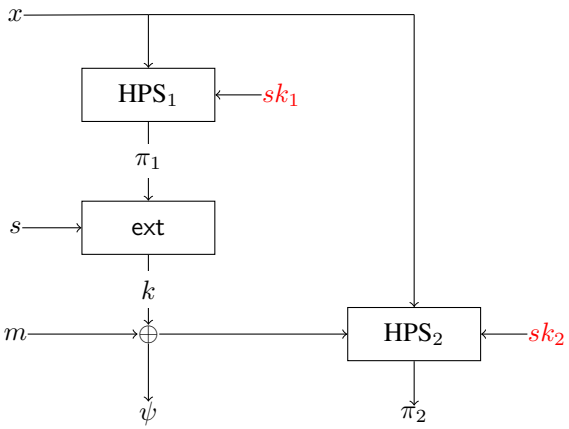


图 5.4: 基于 HPS 的 LR-CCA 安全 PKE 的构造思路

5.1.2.2 基于可公开求值伪随机函数的 LR-CPA 安全 PKE

前面构造的抗泄漏密码方案主要利用抗泄漏密码原语或者结合特殊的方案结构，使得挑战者在回答密钥泄漏询问时都可以转化为知道真实私钥的情况. 例如基于哈希证明系统的技术 [67]，利用 SMP 问题假设将良生成密文转化为非良生成密文过程中，挑战者在知道真实私钥的情况下，依然可以利用 SMP 假设将集合 L 中的元素转化为集合 $X \setminus L$ 中的元素. 再结合 Leftover Hash Lemma 2.3 ，即使在泄漏部分私钥的情况下依然能够提取出伪随机比特串用于掩盖真实消息. 2018 年，陈等 [73] 提出一种在计算意义下利用模拟泄漏来构造泄漏谕言机的方法. 该方法引入抗泄漏可公开求值伪随机函数 (leakage-resilient publicly evaluable PRF, LR-PEPRF)，证明了 LR-PEPRF 直接蕴含了抗泄漏公钥加密方案. 下面主要介绍 LR-PEPRF 的定义、构造及应用.

LR-PEPRF 的定义. 可公开求值伪随机函数 (PEPRF) 的概念是陈等 [301] 于 2014 年提出的一种具有特殊性质的伪随机函数，可以看作是公钥密码学领域与弱伪随机函数对应的一种密码学原语. 在 PEPRF 中，每个私钥都对应了一个公钥，以及一个 $\mathcal{NP}$ 语言集. 对于语言中的任意元素，除了利用私钥计算其 PRF 值外，也可以利用公钥及该元素相应的证据来计算. PEPRF 可以由特殊假设或更通用假设构造，例如 (可提取) 哈希证明系统和陷门函数. PEPRF 的形式化定义及其性质见第 4.5节定义 4.15. 下面主要介绍 LR-PEPRF 的定义.

定义 5.1 (抗泄漏可公开求值伪随机函数)

令 PEPRF $=$ (Setup, KeyGen, PrivEval, PubEval) 是一个可公开求值伪随机函数，$\mathcal{A}$ 是一个攻击 PEPRF 伪随

机性的敌手. 定义 PEPRF 的抗泄漏伪随机性敌手的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa)=\left|\Pr\left[\beta'=\beta:\begin{array}{l} pp\leftarrow\mathsf{Setup}(1^{\kappa});\\ (pk,sk)\leftarrow\mathsf{KeyGen}(pp);\\ state\leftarrow\mathcal{A}^{\mathcal{O}_{\mathsf{leak}}}(pk);\\ (x^*,w^*)\leftarrow\mathsf{SampR}(pp);\\ y_0^*\leftarrow\mathsf{PrivEval}(sk,x^*),y_1^*\xleftarrow{\mathrm{R}}Y;\\ \beta\xleftarrow{\mathrm{R}}\{0,1\};\\ \beta'\leftarrow\mathcal{A}(pk,x^*,y_\beta^*); \end{array}\right]-\frac{1}{2}\right|$$

其中, $\mathcal{O}_{\mathsf{leak}}$ 是泄漏谕言机, 输入泄漏函数 $f:SK\to\{0,1\}^*$, 返回 $f(sk)$, 且所有泄漏函数输出的比特长度之和不超过 ℓ. 如果对于任意 PPT 敌手 $\mathcal{A}$, 上述实验中的优势函数是可忽略的, 则称 PEPRF 是抗 ℓ 泄漏弱伪随机的. 陈等在文献 [301] 中指出, 由于 PEPRF 的可公开求值的性质, LR-PEPRF 不可能具有完全伪随机性.

♣

LR-PEPRF 的构造. 利用可穿孔可公开求值伪随机函数 (puncturable publicly evaluable PRF, PPEPRF) 和不可区分混淆器 ($i\mathcal{O}$) 可以构造一个 LR-PEPRF. 在介绍 LR-PEPRF 的构造之前, 先回顾 PPEPRF 的定义.

定义 5.2 (可穿孔可公开求值伪随机函数)

令 $L=\{L_{pk}\}$ 是一个定义在 X 上的 $\mathcal{NP}$ 语言集. 一个可穿孔可公开求值伪随机函数 $F:SK\times X\to Y\cup\perp$ 包含以下 4 个 PPT 算法:

- $\mathsf{Setup}(1^{\kappa})$: 以安全参数 1^{κ} 为输入, 输出公开参数 $pp=(F,PK,SK,X,L,W,Y)$.
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入, 输出公钥 pk 和私钥 sk.
- $\mathsf{PrivEval}(sk,x)$: 以私钥 sk 和元素 $x\in X$ 为输入, 输出 $y\leftarrow F_{sk}(x)\in Y\cup\perp$.
- $\mathsf{Puncture}(sk,x^*)$: 以私钥 sk 和 $x^*\in L_{pk}$ 为输入, 输出可密钥 sk_{x^*}.
- $\mathsf{PuncEval}(sk_{x^*},x)$: 以可密钥 sk_{x^*} 和 $x\neq x^*$ 为输入, 输出 $y\leftarrow F_{sk}(x)\in Y\cup\perp$.
- $\mathsf{PubEval}(pk,x,w)$: 以公钥 pk 和元素 $x\in L_{pk}$ 及证据 w 为输入, 输出 $y\leftarrow F_{sk}(x)\in Y$.
- (弱伪随机性) 令 $\mathcal{A}$ 是一个攻击 PPEPRFs 的敌手, 定义下面的优势函数:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa)=\left|\Pr\left[\beta'=\beta:\begin{array}{l} pp\leftarrow\mathsf{Setup}(1^{\kappa});\\ (pk,sk)\leftarrow\mathsf{KeyGen}(pp);\\ (x^*,w^*)\leftarrow\mathsf{SampRel}(r^*);\\ sk_{x^*}\leftarrow\mathsf{Puncture}(sk,x^*);\\ y_0^*\leftarrow F_{sk}(x^*),y_1^*\xleftarrow{\mathrm{R}}Y;\\ \beta\xleftarrow{\mathrm{R}}\{0,1\};\\ \beta'\leftarrow\mathcal{A}(pk,sk_{x^*},x^*,y_\beta^*); \end{array}\right]-\frac{1}{2}\right|$$

如果对于任意 PPT 敌手 $\mathcal{A}$, 在上述实验中的优势函数都是可忽略的, 则称该 PPEPRF 是弱伪随机的.

♣

下面介绍 LR-PEPRF 的构造.

构造 5.3 (基于 PPEPRF 和 $i\mathcal{O}$ 的 LR-PEPRF)

构造所需组件是:

- 一个关于 $L=\{L_{pk}\}_{pk\in PK}$ 可穿孔可公开求值伪随机函数 $F:SK\times X\to Y\cup\perp$. 不失一般性, 假设 $Y=\{0,1\}^{\rho}$.
- 一个不可区分混淆器 $i\mathcal{O}$.
- 一个平均意义 (n,ϵ)-强随机性提取器 $\mathsf{ext}:Y\times S\to Z$.

令 $\hat{X}=X\times S$, $\hat{L}_{pk}=\{\hat{x}=(x,s):x\in L_{pk}\wedge s\in S\}$ [a]. 则构造 LR-PEPRF $\hat{F}:\hat{SK}\times\hat{X}\to Z\cup\perp$ 如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp \leftarrow F.\mathsf{Setup}(1^\kappa)$, 输出公开参数 pp.
- $\mathsf{KeyGen}(pp)$: 运行 $F.\mathsf{KeyGen}(pp)$ 以生成 (pk, sk), 构造 $\hat{sk} \leftarrow i\mathcal{O}(\text{PrivEval})$, 其中程序 PrivEval 的定义见图 5.5; 输出 $(pk, \hat{sk})$.
- $\mathsf{PrivEval}(\hat{sk}, \hat{x})$: 输入 $\hat{sk}$ 和 $\hat{x} = (x, s) \in \hat{X}$, 输出 $\hat{y} \leftarrow \hat{sk}(\hat{x})$. 这里实际上定义了 $\hat{F}_{\hat{sk}}(\hat{x}) := \mathsf{ext}(F_{sk}(x), s)$, 其中 $\hat{x} = (x, s)$.
- $\mathsf{PubEval}(pk, \hat{x}, w)$: 输入 pk, $\hat{x} = (x, s) \in \hat{L}_{pk}$ 和 $\hat{x}$ 的证据 w, 利用 $F.\mathsf{PubEval}(pk, x, w)$ 计算 $y \leftarrow F_{sk}(x)$, 输出 $z \leftarrow \mathsf{ext}(y, s)$.

[a]根据 $\hat{L}$ 的定义, $x \in L_{pk}$ 的证据 w 也是 $\hat{x} = (x, s) \in \hat{L}_{pk}$ 的证据, 其中 s 是 S 中任意种子.

♣

PrivEval

Constants: PPEPRF secret key sk

Input: $\hat{x} = (x, s)$

1. Output $\mathsf{ext}(F_{sk}(x), s)$.

图 5.5: 程序 PrivEval 的描述

PrivEval*

Constants: PPEPRF punctured secret key sk_{x^*}, x^* and y^*

Input: $\hat{x} = (x, s)$

1. If $x = x^*$, output $\mathsf{ext}(y^*, s)$.
2. Else, output $\mathsf{ext}(F_{sk_{x^*}}(x), s)$.

图 5.6: 程序 PrivEval* 的描述

定理 5.2

如果 F 是一个安全的可穿孔可公开求值伪随机函数, $i\mathcal{O}$ 是一个不可区分混淆器, ext 是一个平均意义 (n, ϵ)-强随机性提取器, 则构造 5.3 中的 PEPRF 是抗 ℓ 比特泄漏弱伪随机的, 其中 $\ell \leq \rho - n$.

♡

证明 通过游戏的方式组织证明. 令 S_i 表示事件 "$\mathcal{A}$ 在游戏 i 中成功".

Game_0: 这是标准的 PEPRF 抗泄漏弱伪随机性游戏. 挑战者 $\mathcal{CH}$ 按以下方式同敌手 $\mathcal{A}$ 交互执行游戏.

1. 初始化: $\mathcal{CH}$ 运行 $pp \leftarrow F.\mathsf{Setup}(1^\kappa)$, $(pk, sk) \leftarrow F.\mathsf{KeyGen}(pp)$, 创建 $\hat{sk} \leftarrow i\mathcal{O}(\text{PrivEval})$, 然后将 pp 和 pk 发送给 $\mathcal{A}$.
2. 询问: 当收到泄漏询问函数 $\langle f \rangle$ 时, 只要泄漏总量不超过 ℓ, $\mathcal{CH}$ 将 $f(\hat{sk})$ 发送给 $\mathcal{A}$.
3. 挑战: $\mathcal{CH}$ 随机采样 $(x^*, w^*) \leftarrow \mathsf{SampR}(pp)$ 和 $s^* \xleftarrow{R} S$, 利用 $F.\mathsf{PubEval}(pk, x^*, w^*)$ 计算 $y^* \leftarrow F_{sk}(x^*)$ 和 $z_0^* \leftarrow \mathsf{ext}(y^*, s^*)$. 接下来, 随机选择 $z_1^* \xleftarrow{R} Z$ 和 $\beta \xleftarrow{R} \{0, 1\}$. 最后, $\mathcal{CH}$ 将 $\hat{x}^* = (x^*, s^*)$ 和 z_β^* 发送给 $\mathcal{A}$.
4. 猜测: $\mathcal{A}$ 输出一比特 β' 作为对 β 的猜测结果. 如果 $\beta' = \beta$, 则 $\mathcal{A}$ 成功.

根据上述定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game_1: 该游戏与 Game_0 的不同之处是 $\mathcal{CH}$ 在初始化阶段就随机采样 x^* 和 w^* 并计算 $y^* \leftarrow F_{sk}(x^*)$. 该变化仅是概念上的不同, 因此:

$$\Pr[S_1] = \Pr[S_0]$$

Game_2: 该游戏与 Game_1 的不同之处是 $\mathcal{CH}$ 在初始化阶段同时计算 $sk_{x^*} \leftarrow F.\text{Puncture}(sk, x^*)$, 并创建 $\hat{sk} \leftarrow i\mathcal{O}(\text{PuncPriv})$. 此处的程序 PrivEval* 通过常量 sk_{x^*}, x^* 和 y^* 构建, 见图 5.6中的定义. 显而易见, 对于所有输入, 程序 PrivEval 和程序 PrivEval* 输出的结果是一致的. 因此, 可以将这两个游戏之间的区别直接归约到 $i\mathcal{O}$ 安全性上, 所以:

$$|\Pr[S_1] - \Pr[S_2]| \leq \text{negl}(\kappa)$$

Game_3: 该游戏与 Game_2 的不同之处在于 $\mathcal{CH}$ 在初始化阶段随机选择 $y^* \xleftarrow{R} Y$, 而不再通过 $y^* \leftarrow F_{sk}(x^*)$ 计算生成. 根据 PPEPRF 的弱伪随机性, 这一变化对于任意 PPT 敌手是不可区分的, 所以:

$$|\Pr[S_2] - \Pr[S_3]| \leq \text{negl}(\kappa)$$

Game_4: 该游戏与 Game_3 的不同之处在于 $\mathcal{CH}$ 在挑战阶段随机选择 $z_0^* \xleftarrow{R} Z$, 而不再通过 $z_0^* \leftarrow \text{ext}(y^*, s^*)$ 计算生成.

令 V 表示由公钥 pk, x^* 和 s^* 组成的集合. 在 Game_3 和 Game_4 中, y^* 是从 Y 上均匀选取且独立于 V, 因此 $\mathsf{H}_\infty(y^*|V) = \rho$. 注意到 $\mathcal{A}$ 最多可以获取 $\hat{sk}$ 的 ℓ 比特泄漏量, 记作 leak. 该泄漏量与 y^* 有关. 根据引理 2.2有 $\tilde{\mathsf{H}}_\infty(y^*|(V, \text{leak})) \geq \mathsf{H}_\infty(y^*|V) - \ell = \rho - \ell$, 且大于 n. 由于 ext 是一个平均意义 (n, ϵ)-强随机性提取器, 由此可得即使在 V 和泄漏信息已知的条件下, $\text{ext}(y^*, s^*)$ 与 $z_0^* \in Z$ 的统计距离不超过 ϵ. 注意到 $\mathcal{A}$ 在 Game_3 和 Game_4 中的视图完全由 z_0^*, z_1^*, β^*, V 和 leak 确定, 而 z_1^*, β^*, V 和 leak 的分布在两个游戏中是一样的. 所以 $\mathcal{A}$ 在这两个游戏中的视图差异不超过 $\epsilon/2$. 因此:

$$|\Pr[S_3] - \Pr[S_4]| \leq \epsilon/2 \leq \text{negl}(\kappa)$$

在 Game_4 中, z_0^* 和 z_1^* 都是从 Z 中随机选取的. 所以:

$$\Pr[S_4] = 1/2$$

综上, 定理 5.2得证! □

下面进一步介绍一种提高泄漏率至最优的构造方法.

构造 5.4 (泄漏率最优的 LR-PEPRF)

构造所需组件是:

- 一个关于 $L = \{L_{pk}\}_{pk \in PK}$ 可穿孔可公开求值伪随机函数 $F : SK \times X \to Y \cup \perp$. 不失一般性, 假设 $Y = \{0,1\}^\rho$.
- 一个不可区分混淆器 $i\mathcal{O}$.
- 一个平均意义 (n, ϵ)-强随机性提取器 $\text{ext} : Y \times S \to Z$.
- 一个 IND-CPA 安全的对称加密方案 SKE, 其消息空间为 $\{0,1\}^\rho$、密文空间为 $\{0,1\}^v$.
- 一个 (v, τ)-有损函数.

构造 LR-PEPRF 如下:

- $\text{Setup}(1^\kappa)$: 运行 $pp \leftarrow F.\text{Setup}(1^\kappa)$, $pp' \leftarrow \text{LF.Setup}(1^\kappa)$ 和 $pp'' \leftarrow \text{SKE.Setup}(1^\kappa)$, 输出公开参数 $\hat{pp} = (pp, pp', pp'')$.
- $\text{KeyGen}(\hat{pp})$: 运行 $(pk, sk) \leftarrow F.\text{KeyGen}(pp)$, $h \leftarrow \text{LF.GenInj}(pp')$, $k_e \leftarrow \text{SKE.keyGen}(pp'')$, 创建一个冗余密文 $ct \leftarrow \text{SKE.Enc}(k_e, 0^\rho)$ 作为 $\hat{sk}$, 计算 $\eta^* \leftarrow h(ct)$, 创建 $C_{\text{eval}} \leftarrow i\mathcal{O}(\text{PrivEval})$ (程序 PrivEval 的定义见图 5.7, η^* 看作是该程序的触发器), 设置 $\hat{pk} = (pk, C_{\text{eval}})$, 输出 $(\hat{pk}, \hat{sk})$.
- $\text{PrivEval}(\hat{sk}, \hat{x})$: 输入 $\hat{sk}$ 和 $\hat{x} = (x, s) \in \hat{X}$, 输出 $\hat{y} \leftarrow C_{\text{eval}}(\hat{sk}, \hat{x})$. 这相当于定了 $\hat{F}_{\hat{sk}}(\hat{x}) := \text{ext}(F_{sk}(x), s)$, 其中 $\hat{x} = (x, s)$.
- $\text{PubEval}(\hat{pk}, \hat{x}, w)$: 输入 $\hat{pk} = (pk, C_{\text{eval}}, t)$, $\hat{x} = (x, s) \in \hat{L}_{pk}$ 和 $\hat{x}$ 的证据 w, 利用 $F.\text{PubEval}(pk, x, w)$ 计算 $y \leftarrow F_{sk}(x)$. 输出 $\hat{y} \leftarrow \text{ext}(y, s)$.

♣

PrivEval

Constants: PPEPRF secret key sk, η^*

Input: $\hat{sk}$, $\hat{x} = (x, s)$

1. If $h(\hat{sk}) \neq \eta^*$, output $\perp$.
2. Else, output $\mathsf{ext}(F_{sk}(x), s)$.

图 5.7: 程序 PrivEval 的描述

PrivEval*

Constants: PPEPRF punctured secret key sk_{x^*}, k_e, x^* and η^*

Input: $\hat{sk}$, $\hat{x} = (x, s)$

1. If $h(\hat{sk}) \neq \eta^*$, output $\perp$.
2. If $x = x^*$, set $y^* \leftarrow \mathsf{SKE.Dec}(k_e, \hat{sk})$, output $\mathsf{ext}(y^*, s)$.
3. Else, output $\mathsf{ext}(F_{sk_{x^*}}(x), s)$.

图 5.8: 程序 PuncEval 的描述

定理 5.3

如果 F 是一个安全的可孔可公开求值伪随机函数, $i\mathcal{O}$ 是一个不可区分混淆器, SKE 是一个 IND-CPA 安全的对称加密方案, LF 是一个 (v, τ)-有损函数族, ext 是一个平均意义 (n, ϵ)-强随机性提取器, 那么构造 5.4 中的 PEPRF 是抗 ℓ 比特泄漏弱伪随机的, 其中 $\ell \leq \rho - n - \tau$.

通过设置合适的参数, 如设置 $v = \rho + o(\rho)$, $n = o(\rho)$, $\tau = o(v)$, 则有 $|\hat{sk}| = v = \rho + o(\rho)$, $\ell = \rho - o(\rho)$, 并且泄漏率达到最优.

证明 下面通过游戏的方式组织证明. 令 S_i 表示事件 "$\mathcal{A}$ 在游戏 i 中成功".

Game_0: 该游戏是标准的 PEPRFs 抗泄漏弱伪随机性游戏. $\mathcal{CH}$ 与 $\mathcal{A}$ 按以下方式交互完成游戏.

1. 初始化: $\mathcal{CH}$ 运行 $\hat{pp} = (pp, pp', pp'') \leftarrow \mathsf{Setup}(1^\kappa)$, $(pk, sk) \leftarrow F.\mathsf{KeyGen}(pp)$, $h \leftarrow \mathsf{LF.GenInj}(pp')$, 随机选取 $k_e \leftarrow \mathsf{SKE.KeyGen}(pp'')$, 生成一个冗余密文 $ct \leftarrow \mathsf{SKE.Enc}(k_e, 0^\rho)$ 作为 $\hat{sk}$, 计算 $\eta^* \leftarrow h(ct)$ 和 $C_{\mathrm{eval}} \leftarrow i\mathcal{O}(\mathrm{PrivEval})$. $\mathcal{CH}$ 设置 $\hat{pk} = (pk, C_{\mathrm{eval}})$ 并发送给敌手 $\mathcal{A}$.
2. 询问: 当收到泄漏询问函数 $\langle f \rangle$ 时, 只要泄漏总量不超过 ℓ, 则 $\mathcal{CH}$ 将 $f(\hat{sk})$ 发送给 $\mathcal{A}$.
3. 挑战: $\mathcal{CH}$ 随机采样 $(x^*, w^*) \leftarrow \mathsf{SampR}(pp_1)$, 选择 $s^* \xleftarrow{R} S$, 利用 $F.\mathsf{PubEval}(pk, x^*, w^*)$ 计算 $y^* \leftarrow F_{sk}(x^*)$, $z_0^* \leftarrow \mathsf{ext}(y^*, s^*)$, 随机选择 $z_1^* \xleftarrow{R} Z$, $\beta \xleftarrow{R} \{0,1\}$. 最后, $\mathcal{CH}$ 将 $\hat{x}^* = (x^*, s^*)$ 和 z_β^* 发送给 $\mathcal{A}$.
4. 猜测: $\mathcal{A}$ 输出一比特 β' 作为对 β 的猜测结果. 如果 $\beta' = \beta$, 则 $\mathcal{A}$ 成功.

根据定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game_1: 该游戏与 Game_0 的不同之处在于 $\mathcal{CH}$ 在初始化阶段就随机选择 x^*, w^* 并计算 $y^* \leftarrow F_{sk}(x^*)$. 该变化仅是概念上的不同, 因此:

$$\Pr[S_1] = \Pr[S_0]$$

Game_2: 该游戏与 Game_1 的不同之处在于 $\mathcal{CH}$ 在初始化阶段利用 $ct \leftarrow \mathsf{SKE.Enc}(k_e, y^*)$ 替代 $\mathsf{SKE.Enc}(k_e, 0^\rho)$ 计算冗余密文 ct. 这一区别可以直接归约到 SKE 的 IND-CPA 安全性上, 所以:

$$|\Pr[S_1] - \Pr[S_2]| \leq \mathsf{negl}(\kappa)$$

Game_3: 该游戏与 Game_2 的不同之处在于 $\mathcal{CH}$ 在初始化阶段同时计算$sk_{x^*} \leftarrow F.\mathsf{Puncture}(sk, x^*)$并创建 $C_{\mathrm{eval}} \leftarrow$

$i\mathcal{O}(\text{PrivEval}^*)$. 程序 PrivEval^* 的定义见图 5.8.

根据 h 的单射性和 SKE、PPEPRF 的正确性, 对于任意输入, 两个程序 PrivEval 和 PuncPriv 的输出结果是一致的. 因此, 这两个游戏之间的区别可以直接归约到 $i\mathcal{O}$ 安全性上, 所以:

$$|\Pr[S_2] - \Pr[S_3]| \leq \mathsf{negl}(\kappa)$$

Game_4: 该游戏与 Game_3 的不同之处在于在初始化阶段, $\mathcal{CH}$随机选择 $y^* \xleftarrow{R} Y$而不再是通过 $y^* \leftarrow F_{sk}(x^*)$ 计算所得.

基于 PPEPRF 的伪随机性假设, 这两个游戏之间的变化对于任意 PPT 敌手是不可区分的, 即:

$$|\Pr[S_3] - \Pr[S_4]| \leq \mathsf{negl}(\kappa)$$

Game_5: 该游戏与 Game_4 的不同之处在于 $\mathcal{CH}$利用 $\text{LF.GenLossy}(\ell)$选择函数 h, 而不再选择一个单射函数. 这两个游戏之间的区别可以直接归约到有损函数的安全性上, 所以:

$$|\Pr[S_4] - \Pr[S_5]| \leq \mathsf{negl}(\kappa)$$

Game_6: 该游戏与 Game_5 的区别在于 $\mathcal{CH}$ 在挑战阶段随机选择 $z_0^* \xleftarrow{R} Z$, 而不再设置 $z_0^* \leftarrow \mathsf{ext}(y^*, s^*)$.

令 V 表示公钥 $\hat{pk} = (pk, C_{\text{eval}})$, x^* 和 s^* 组成的集合. 在 Game_5 和 Game_6 中, y^* 是从 Y 中均匀选取且与 sk_{x^*}, x^* 和 s^* 完全独立, 而 η^* 最多有 2^τ 种取值, 根据引理 2.2, 则有 $\mathsf{H}_\infty(y^*|V) \geq \rho - \tau$. 注意到 $\mathcal{A}$ 最多可以获取 $\hat{sk}$ 的 ℓ 比特泄漏量, 记作 leak. 该泄漏量与 y^* 有关. 根据引理 2.2可得 $\tilde{\mathsf{H}}_\infty(y^*|(V, \mathsf{leak})) \geq \mathsf{H}_\infty(y^*|V) - \ell = \rho - \tau - \ell$, 且该值大于 n. 由于 ext 是一个平均意义 (n, ϵ)-强随机性提取器, 由此可得即使在 V 和泄漏信息已知的条件下, $\mathsf{ext}(y^*, s^*)$ 与 $z_0^* \in Z$ 的统计距离不超过 ϵ. 注意到 $\mathcal{A}$ 在 Game_5 和 Game_6 中的视图完全由 z_0^*, z_1^*, β^*, V 和 $leak$ 确定, 而 z_0^*, z_1^*, β^*, V 和 leak 的分布在两个游戏中是一样的. 所以 $\mathcal{A}$ 在这两个游戏中的视图差异不超过 $\epsilon/2$. 因此:

$$|\Pr[S_6] - \Pr[S_5]| \leq \epsilon/2 \leq \mathsf{negl}(\kappa)$$

在 Game_6 中, z_0^* 和 z_1^* 都是从 Z 中随机选取的. 所以:

$$\Pr[S_6] = 1/2$$

综上, 定理 5.3得证! □

LR-PEPRF 的应用. 弱伪随机的可公开求值伪随机函数直接蕴含了 IND-CPA 安全的密钥封装/公钥加密方案 [301]. 该结论在密钥泄漏环境下依然成立 [73]. 由此, 利用抗泄漏的可公开求值伪随机函数, 按照下面的方式可以构造一个抗泄漏公钥加密方案.

假设 $F : SK \times X \to Y$ 是一个 LR-PEPRF, 其中 $L = \{L_{pk}\}_{pk \in PK}$, Y 是一个加法群. PKE 的密钥同 PEPRF 的密钥. 当加密消息 $m \in Y$ 时, 随机选择 $x \xleftarrow{R} L_{pk}$ 及其证据 w, 计算 $k \leftarrow \mathsf{PubEval}(pk, x, w)$, 输出密文 $(x, k + m)$. 解密过程是利用 $\mathsf{PrivEval}(sk, x)$ 重新计算 k. PKE 方案的 LR-CPA 安全性依赖于 PEPRF 的抗泄漏弱伪随机性.

构造 5.5 (基于 LR-PEPRF 的 LR-CPA 安全 PKE)

构造所需组件是:

- 一个抗泄漏弱随机安全的伪随机函数 $F : SK \times X \to Y$, 其中 $L = \{L_{pk}\}_{pk \in PK}$, Y 是一个加法群.

构造 LR-CPA PKE 如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp \leftarrow F.\mathsf{Setup}(1^\kappa)$, 输出公开参数 pp.
- $\mathsf{KeyGen}(pp)$: 运行 $(pk, sk) \leftarrow F.\mathsf{KeyGen}(pp)$, 输出公钥 pk 和私钥 sk.
- $\mathsf{Encrypt}(pk, m)$: 以公钥 pk 和明文 $m \in \{0,1\}^\kappa$, 执行如下步骤:
  1. 随机选择 $x \xleftarrow{R} L_{pk}$ 及其证据 w;
  2. 计算 $k \leftarrow \mathsf{PubEval}(pk, x, w)$;
  3. 输出 $(x, k + m)$ 作为密文 c.
- $\mathsf{Decrypt}(sk, c)$: 以私钥 sk 和密文 $c = (x, \psi)$ 为输入, 计算 $k \leftarrow \mathsf{PrivEval}(sk, x)$, 输出明文 $m' := \psi \oplus k$. ♣

5.1.3 LR-CCA 安全 PKE 的通用构造方法

5.1.3.1 基于一次有损过滤器的 LR-CCA 安全 PKE

2013 年, Qin 和 Liu [70] 提出一次有损过滤器 (one-time lossy filter, OT-LF) 的概念. 类似 LTF, OT-LF 也具有单射和有损两种在计算上不可区分的工作模式. 不同之处在于, OT-LF 不需要求逆计算, 并且有损模式的像空间大小固定, 不会随着原像空间的增大而增大, 从而保证 OT-LF 输出的结果仅泄漏固定量的原像信息.

将 HPS 和 OT-LF 结合可以构造高效的 LR-CCA 安全的 PKE 方案, 基本思路是将基于哈希证明系统构造模式中的第二个用于验证密文有效性的 universal$_2$ 哈希证明系统 HPS$_2$ 换为 OT-LF. 通过 OT-LF 验证 universal$_1$ 哈希证明系统 HPS$_1$ 的输出结果 π_1 的正确性来验证密文的有效性, 其构造思路如图 5.9所示. 在解密阶段, 由于 OT-LF 不需要求逆操作, 所以整个方案的私钥仅包含 HPS$_1$ 的私钥. 尽管 OT-LF 不需要输入私钥, 但是需要输入一个标签 t_c 用于控制 OT-LF 的工作模式. 这种构造方法的密钥泄漏比率主要取决于所基于的 HPS 容忍密钥泄漏的性质. 在理论上, 泄漏比率可以达到 $1-o(1)$ 的最优结果, 但是需要基于特殊的困难问题构造具有 universal$_1$ 性质的哈希证明系统, 如文献 [91] 基于子群不可区分问题的实例化方案. 基于 DDH 或 DCR 等标准困难问题的实例化方案能够达到 $1/2-o(1)$ 的泄漏比率, 要优于基于两个哈希证明系统的实例化方案.

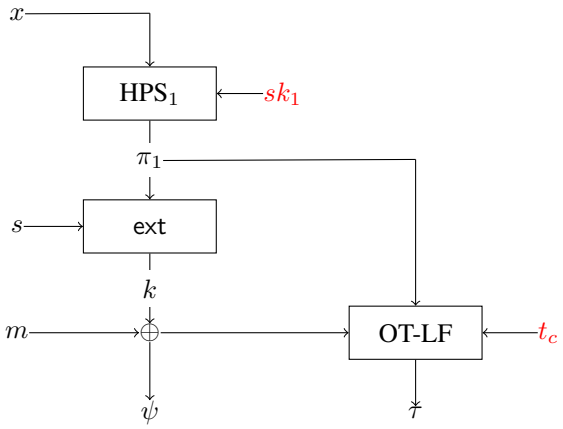


图 5.9: 基于 OT-LF 的 LR-CCA 安全 PKE 方案构造思路

下面介绍 OT-LF 的形式化定义及应用.

一个 (X,τ)-OT-LF 是一族以公钥 ek 和标签 t 为指标的函数: $\{\mathsf{LF}_{ek,t}: X\to Y\}$. 函数族中的任意函数 $\mathsf{LF}_{ek,t}$ 将 $x\in X$ 映射到 $\mathsf{LF}_{ek,t}(x)$. 给定公钥 ek, 标签集合 T 可以分解为两个计算上不可区分的子集合: 单射标签集合 T_{inj} 和有损标签集合 T_{loss}. 如果 t 属于单射标签, 则函数 $\mathsf{LF}_{ek,t}$ 也是单射的并且像的大小为 $|X|$; 如果 t 是有损的, 则函数最多有 2^{τ} 个可能的输出结果. 因此, 若 t 是有损标签, 则 $\mathsf{LF}_{ek,t}(x)$ 最多泄漏 x 的 τ 比特信息. 这一性质在方案证明中是至关重要的.

定义 5.3 (一次有损过滤器)

一个 (X,τ)-OT-LF 包含以下 4 个 PPT 算法并满足以下性质:

- $\mathsf{Setup}(1^{\kappa})$: 输入安全参数 1^{κ}, 输出公开参数 pp.
- $\mathsf{KeyGen}(pp)$: 输入公开参数 pp, 输出一对密钥 (ek,td). 其中, 公钥 ek 定义了标签集合 $T=\{0,1\}^*\times T_c$, 它由两个不相交的有损标签集合 $T_{loss}\subseteq T$ 和单射标签集合 $T_{inj}\subseteq T$ 构成. 每个标签 $t=(t_a,t_c)\in T$ 由辅助标签 $t_a\in\{0,1\}^*$ 和核心标签 $t_c\in T_c$ 两部分组成. td 是一个陷门, 利用它可以有效地从有损标签集合中进行抽样.
- $\mathsf{Eval}(ek,t,x)$: 给定公钥 ek 和标签 t, 将 $x\in X$ 映射到 $\mathsf{LF}_{ek,t}(x)\in Y$.
- $\mathsf{LTag}(td,t_a)$: 利用陷门 td 计算辅助标签 t_a 对应的核心标签 t_c, 使其满足 $t=(t_a,t_c)\in T_{loss}$.
- 有损性: 如果 t 是单射的, 则函数 $\mathsf{LF}_{ek,t}(\cdot)$ 也是单射的; 如果 t 是有损的, 则 $\mathsf{LF}_{ek,t}(x)$ 的像集合最多包含 2^{τ} 个元素. (在应用中, 通过调整公钥参数, 原像集合可以逐渐增大而参数 τ 始终保持不变.)

- 不可区分性: 对于任意 PPT 算法 $\mathcal{A}$, 区分有损标签和随机选取的标签是困难的. 严格来说, 对于任意 PPT 算法 $\mathcal{A}$, 下面的优势函数是可忽略的

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left|\Pr[\mathcal{A}(ek,(t_a,t_c^{(0)}))=1] - \Pr[\mathcal{A}(ek,(t_a,t_c^{(1)}))=1]\right|$$

  其中 $pp \leftarrow \mathsf{Setup}(1^\kappa)$, $(ek,td) \leftarrow \mathsf{KeyGen}(pp)$, $t_a \leftarrow \mathcal{A}(ek)$, $t_c^{(0)} \leftarrow \mathsf{LTag}(td,t_a)$, $t_c^{(1)} \xleftarrow{R} T_c$.
- 隐没性: 对于任意 PPT 敌手 $\mathcal{A}$, 即使给定一个有损标签情况下, 也无法计算一个新的非单射标签 (在有些情况下, 一个标签可能既不是单射的也不是有损的). 严格来说, 对于任意 PPT 算法 $\mathcal{A}$, 下面的优势函数是可忽略的:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[\begin{array}{l|l} (t_a',t_c') \neq (t_a,t_c) \ \wedge (t_a',t_c') \in T \setminus T_{inj} & \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (ek,td) \leftarrow \mathsf{KeyGen}(pp); \\ t_a \leftarrow \mathcal{A}(ek); t_c \leftarrow \mathsf{LTag}(td,t_a); \\ (t_a',t_c') \leftarrow \mathcal{A}(ek,(t_a,t_c)); \end{array} \end{array}\right]$$

笔记 一次有损过滤器也可以看作是一种简化的有损代数过滤器 [314]. 两者存在以下不同之处: 一是前者要求敌手最多知道一个有损标签; 而后者要求敌手可以获得多个有损标签, 这导致后者比前者实现难度大且实现的方案效率非常差. 二是前者不需要具有特定的代数结构, 而后者必须有特定的代数结构, 可用于多挑战密文的环境, 例如 KDM-CCA 安全性. 此外, 一次有损过滤器也可看作是一种不带求逆陷门的全除一有损陷门函数. 在单射模式下, 一次有损过滤器没有求逆陷门, 而全除一有损陷门函数则需要一个陷门能够用于求逆. 因此, 在相同定义域下, 一次有损过滤器的计算效率一般比全除一有损陷门函数高.

前面已经介绍了利用 OT-LF 构造 LR-CCA 安全的公钥加密方案的设计思路, 下面给出通用构造的细节.

构造 5.6 (基于 OT-LF 的 LR-CCA 安全 PKE)

令 $\ell = \ell(\kappa)$ 为密钥泄漏量的上界, ϵ_1 和 ϵ_2 是两个可忽略的量. 构造所需组件是:

- 一个 ϵ_1-universal$_1$ 哈希证明系统 $\mathsf{HPS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PubEval}, \mathsf{PrivEval})$.
- 一个 (Π,τ)-一次有损过滤器 $\mathsf{OTLF} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Eval}, \mathsf{LTag})$.
- 一个平均意义 $(\log\Pi - \ell, \epsilon_2)$-强随机性提取器 $\mathsf{ext} : \Pi \times \{0,1\}^d \to \{0,1\}^\kappa$.

构造 LR-CCA PKE 如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{HPS.Setup}(1^\kappa)$, 输出 HPS 的一个实例参数 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$, 运行 $\mathsf{OTLF.Setup}(1^\kappa)$, 输出 OT-LF 的公开参数 pp'. 选择一个平均意义 $(\log\Pi - \ell, \epsilon_2)$-强随机性提取器 $\mathsf{ext} : \Pi \times \{0,1\}^d \to \{0,1\}^\kappa$. 将 $\hat{pp} = (pp, pp', \mathsf{ext})$ 作为公开参数, 其中 $M = \{0,1\}^\kappa$ 作为明文空间, $C = X \times \{0,1\}^d \times \{0,1\}^\kappa \times Y \times t_c$ 作为密文空间 C.
- $\mathsf{KeyGen}(\hat{pp})$: 运行 $(pk,sk) \leftarrow \mathsf{HPS.KeyGen}(pp)$ 和 $(ek,td) \leftarrow \mathsf{OTLF.KeyGen}(pp')$. 输出公钥 $\hat{pk} = (pk, ek)$ 和私钥 $\hat{sk} = sk$.
- $\mathsf{Encrypt}(\hat{pk}, m)$: 以公钥 $\hat{pk} = (pk, ek)$ 和明文 $m \in \{0,1\}^\kappa$ 为输入, 执行如下步骤:
  1. 运行 $(x,w) \leftarrow \mathsf{SampR}(r)$ 生成随机实例 x 和相应的证据 w;
  2. 通过 $\mathsf{HPS.PubEval}(pk,x,w)$ 计算实例 x 的哈希证明 $\pi \leftarrow \mathsf{H}_{sk}(x)$;
  3. 随机选择 $s \xleftarrow{R} \{0,1\}^d$, 计算 $\psi = \mathsf{ext}(\pi,s) \oplus m$;
  4. 随机选择 $t_c \xleftarrow{R} T_c$, 计算 $y \leftarrow \mathsf{LF}_{ek,t}(\pi)$, 其中 $t = (t_a, t_c)$, $t_a = (x,s,\psi)$;
  5. 输出密文 $c = (x,s,\psi,y,t_c)$.
- $\mathsf{Decrypt}(\hat{sk}, c)$: 以私钥 $\hat{sk} = sk$ 和密文 $c = (x,s,\psi,y,t_c)$ 为输入, 执行以下步骤:
  1. 计算 $\pi' \leftarrow \mathsf{H}_{sk}(x)$ 和 $y' \leftarrow \mathsf{LF}_{ek,t}(\pi')$, 其中 $t = ((x,s,\psi),t_c)$;
  2. 验证 $y' = y$ 是否成立. 如果不成立, 则返回 $\perp$; 否则输出明文 $m' = \psi \oplus \mathsf{ext}(\pi', s)$.

正确性. 方案的正确性可以通过哈希证明系统和一次有损过滤器的正确性直接验证. 由于 $\mathsf{HPS.PrivEval}(sk,x) =$

$\mathsf{HPS.PubEval}(pk, x, w) = \mathsf{H}_{sk}(x)$, 所以 $\mathsf{LF}_{ek,t}(\pi') = \mathsf{LF}_{ek,t}(\pi)$. 从而, 解密算法中的验证等式 $y' = y$ 成立. 进一步, 根据如下公式可以说明方案具有完美正确性.

$$\begin{aligned} m' &= \psi \oplus \mathsf{ext}(\mathsf{HPS.PrivEval}(sk, x), s) \\ &= \mathsf{ext}(\mathsf{HPS.PubEval}(pk, x, w), s) \oplus m \oplus \mathsf{ext}(\mathsf{HPS.PrivEval}(sk, x), s) \\ &= \mathsf{ext}(\mathsf{H}_{sk}(x), s) \oplus m \oplus \mathsf{ext}(\mathsf{H}_{sk}(x), s) \\ &= m \end{aligned}$$

定理 5.4

如果 HPS 是一个 ϵ_1-universal$_1$ 哈希证明系统, OTLF 是一个 (Π, τ)-一次有损过滤器, $\mathsf{ext} : \Pi \times \{0,1\}^d \to \{0,1\}^\kappa$ 是一个平均意义 $(\nu - \ell - \tau, \epsilon_2)$-强随机性提取器, 则构造 5.6中的 PKE 是 ℓ-LR-CCA 安全的, 其中 $\nu = \log(1/\epsilon_1)$, $\ell \le \nu - \kappa - \tau - \omega(\log \kappa)$.

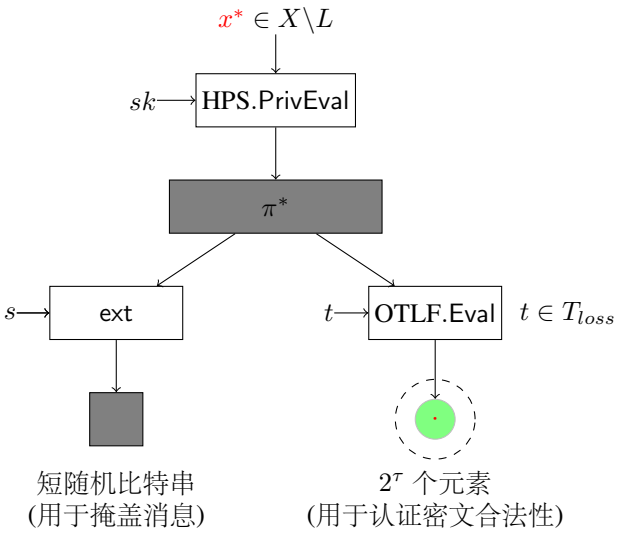


图 5.10: 挑战密文的基本特征

证明思路: 在给出定理 5.4的证明之前, 先概括地介绍一下构造 5.6的安全性证明思路. 该方案首先使用哈希证明系统生成一个对称密钥 π, 它既用作隐藏明文又用作验证密文的正确性. 图 5.10展示了挑战密文的一些基本特征及挑战密文可能泄漏 π^* 的信息熵. 在图中, 哈希证明 π^* 的熵较高且 OT-LF 工作在有损模式下. 当挑战密文中元素 $x^* \in X \backslash L$ 时, 根据 HPS 的 universal$_1$ 性质, 此时 π^* 具有至少 $\log(1/\epsilon_1)$ 的最小熵. 利用 OT-LF 的性质, 挑战密文中的 OT-LF 标签可以转化为有损标签并且用于验证密文有效性的元素 y^* 的空间大小仅为 2^τ. 因此, 挑战密文仅泄漏了 π^* 固定量的信息, 这相当于敌手看到挑战密文后, HPS 的私钥仅泄漏很少一部分信息. 而当敌手进行解密查询时, 如果 $x \in X \backslash L$, 由于 OT-LF 以压倒性的概率工作在单射模式下, 所以敌手必须完全知道 x 对应的哈希证明 π, 才能计算出正确的 y. 除去挑战密文和通过密钥泄漏查询获得的私钥信息, 如果 π 依然有足够多的最小熵, 那么敌手的解密查询通过密文有效性验证的概率是可忽略的, 从而使得解密查询对于敌手来说是没有帮助的.

证明 下面通过不可区分游戏组织证明. 每个游戏的参与者包括挑战者 (模拟者)$\mathcal{CH}$ 和一个 PPT 敌手 (算法)$\mathcal{A}$, $\mathcal{CH}$ 通过与 $\mathcal{A}$ 交互通信, 最终输出一比特信息 β' 作为对 $\mathcal{CH}$ 选择的随机比特 β 的猜测. 在游戏 Game$_i$ 中, 用 S_i 表示事件 $\beta' = \beta$, 用 $c^* = (x^*, s^*, \psi^*, y^*, t_c^*)$ 表示挑战密文. 如果密文 $c = (x, s, \psi, y, t_c)$ 中 $x \in X \backslash L$, 则称该密文是非良生成的; 如果 $y = \mathsf{LF}_{ek,t}(\mathsf{H}_{sk}(x))$ 成立, 则称该密文是有效的. 初始游戏 Game$_0$ 及后续游戏的定义如下:

Game$_0$: 该游戏是标准的 LR-CCA 游戏, 挑战者 $\mathcal{CH}$ 和敌手 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 运行 $\mathsf{Setup}(1^\kappa)$ 生成公开参数 $\hat{pp} = (pp, pp', \mathsf{ext})$, 同时运行 $\mathsf{KeyGen}(\hat{pp})$ 生成公钥 $\hat{pk} = (pk, ek)$ 和私钥 $\hat{sk} = sk$, 其中 (pk, sk) 是 HPS 的密钥, ek 是 OT-LF 的密钥. $\mathcal{CH}$ 将 $(\hat{pp}, \hat{pk})$ 发送给 $\mathcal{A}$.
- 阶段 1 询问: 对于每个解密询问 c 或私钥泄漏询问 f_i, $\mathcal{CH}$ 利用私钥 $\hat{sk}$ 作出回答 $\mathsf{Decrypt}(\hat{sk}, c)$ 或 $f_i(\hat{sk})$.

- 挑战: $\mathcal{A}$ 选择两个等长消息 m_0 和 m_1 并发送给 $\mathcal{CH}$. $\mathcal{CH}$ 随机选择比特 $\beta \xleftarrow{R} \{0,1\}$, 作如下计算:
  1. 运行 $(x^*, w^*) \leftarrow \mathsf{SampRel}(r^*)$ 生成随机实例 x^* 和相应的证据 w^*;
  2. 通过 $\mathsf{HPS.PubEval}(pk, x^*, w^*)$ 计算实例 x 的哈希证明 $\pi^* \leftarrow \mathsf{H}_{sk}(x^*)$;
  3. 随机选择 $s^* \xleftarrow{R} \{0,1\}^d$, 计算 $\psi^* = \mathsf{ext}(\pi^*, s^*) \oplus m$;
  4. 随机选择 $t_c^* \xleftarrow{R} T_c$, 计算 $y \leftarrow \mathsf{LF}_{ek,t^*}(\pi)$, 其中 $t^* = (t_a^*, t_c^*)$, $t_a^* = (x^*, s^*, \psi^*)$;
  5. 输出挑战密文 $c^* = (x^*, s^*, \psi^*, y^*, t_c^*)$ 并发送给 $\mathcal{A}$.
- 阶段 2 询问: $\mathcal{A}$ 可以继续访问解密谕言机但不能访问密钥泄漏谕言机. 此外, 解密询问需满足 $c \neq c^*$.
- 猜测: $\mathcal{A}$ 输出一个比特 β', 作为对 β 的猜测. $\mathcal{A}$ 成功当且仅当 $\beta' = \beta$.

根据 LR-CCA 安全性的定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game_1: 该游戏与 Game_0 的不同之处在于密钥生成方式和挑战密文中核心标签的选取方式. 具体地, 当运行 $\mathsf{KeyGen}(\hat{pp})$ 生成加密方案的公钥/私钥对时, 挑战者除了保留解密私钥 sk 外, 还保留一次有损过滤器 OTLF 的陷门 td. 在选择核心标签时, 模拟者计算$t_c^* \leftarrow \mathsf{OTLF.LTag}(td, t_a^*)$, 其中 $t_a^* = (x^*, s^*, \psi^*)$, 而不是随机选取 $t_c^* \xleftarrow{R} T_c$. 根据 OT-LF 有损标签和随机标签的不可区分性, 则有:

$$\Pr[S_0] - \Pr[S_1] \leq \mathsf{Adv}_{\mathcal{B}_1}(\kappa)$$

其中 $\mathcal{B}_1$ 是一个攻击 OT-LF 标签不可区分性的敌手.

Game_2: 该游戏与 Game_1 的唯一不同之处在于增加了一条特殊规则用于拒绝解密. 具体来讲, 当敌手解密询问的密文 $c = (x, s, \psi, y, t_c)$ 满足$t = (t_a, t_c) = (t_a^*, t_c^*) = t^*$时, 解密谕言机立即返回 $\perp$ 并终止查询. 为简便起见, 将这种标签称为重用的有损过滤器标签. 下面证明, 若解密询问中的标签是重用的, 则在 Game_1 和 Game_2 中, 这种解密询问都将被拒绝查询. 考虑下面两种情况:

- 情形 1: $y = y^*$. 这意味着 $c = c^*$, 由于 $\mathcal{A}$ 是不允许对挑战密文进行解密询问的, 这种情况在两个游戏中都将被拒绝解密.
- 情形 2: $y \neq y^*$. 由 $t = ((x, s, \psi), t_c) = ((x^*, s^*, \psi^*), t_c^*) = t^*$, 可知 $\pi = \pi^*$, $\mathsf{LF}_{ek,t}(\pi) = \mathsf{LF}_{ek,t^*}(\pi^*) = y^*$. 因此, 这种解密询问在 Game_1 中已经被拒绝了.

根据以上分析, 可知敌手 $\mathcal{A}$ 在游戏 Game_1 和 Game_2 中的视图是一样的. 故有:

$$\Pr[S_1] = \Pr[S_2]$$

Game_3: 该游戏与 Game_2 的唯一不同之处在于挑战密文中 π^* 的生成方式. 在该游戏中, 挑战者通过哈希证明系统的私有计算模式$\mathsf{HPS.PrivEval}(sk, x^*)$替代公开计算模式 $\mathsf{HPS.PubEval}(pk, x^*, w^*)$ 来计算 π^*. 根据 HPS 的投射性质, 这只是一种概念上的改变, 对计算结果没有任何影响. 故有:

$$\Pr[S_3] = \Pr[S_2]$$

Game_4: 该游戏与 Game_3 的唯一不同之处在于挑战密文中随机实例 x^* 的选取方式. 在该游戏中, 挑战者调用 $\mathsf{SampNo}(pp)$ 采样$x^* \xleftarrow{R} X \backslash L$. 根据 SMP 问题的困难性, 敌手 $\mathcal{A}$ 在游戏 Game_4 和 Game_3 中的视图计算不可区分, 即

$$\Pr[S_3] - \Pr[S_4] \leq \mathsf{Adv}_{\mathcal{B}_2}(\kappa)$$

其中 $\mathcal{B}_2$ 为攻击 SMP 问题的敌手.

Game_5: 该游戏与 Game_4 的唯一不同之处在于增加了一种特殊的解密规则. 该规则为: 如果敌手解密询问的密文$c = (x, s, \psi, y, t_c)$ 满足 $x \in X \backslash L$, 则解密谕言机立即返回 $\perp$ 并终止查询. 令事件 bad_x 表示一个解密查询在游戏 Game_5 中被拒绝, 而在游戏 Game_4 中可通过解密规则的验证. 因此, 当且仅当事件 bad_x 发生, 敌手在游戏 Game_5 和 Game_4 中的视图不同. 根据差异引理 2.1, 可知:

$$\Pr[S_4] - \Pr[S_5] \leq \Pr[\mathsf{bad}_x]$$

下面的结论保证了事件 bad_x 发生的概率是可忽略的.

引理 5.2

假设敌手最多询问 $Q(\kappa)$ 次解密谕言机, 则

$$\Pr[\mathrm{bad}_x] \leq Q(\kappa) \cdot \mathsf{Adv}_{\mathcal{B}_3}(\kappa) + \frac{Q(\kappa)2^{\ell+\tau+\kappa}}{2^{\nu} - Q(\kappa)}$$

其中 $\mathcal{B}_3$ 是一个攻击一次有损过滤器“隐没性”的敌手.

Game_6: 该游戏与 Game_5 的唯一不同之处在于挑战密文中 ψ^* 的生成方式. 挑战者随机选择$\psi^* \xleftarrow{R} \{0,1\}^{\kappa}$, 而不是通过计算 $\psi^* \leftarrow \mathsf{ext}(\mathsf{H}_{sk}(x^*), s^*) \oplus m_b$ 所得.

对于敌手来说, 游戏 Game_5 和 Game_6 定义的环境是不可区分的. 首先, 从敌手的视图角度 (记做 $\mathsf{view}'_{\mathcal{A}}$) 分析 $\mathsf{H}_{sk}(x^*)$ 的最小熵. 因为非良生成密文直接被拒绝解密, 所以敌手利用解密谕言机不可能获得关于 $\mathsf{H}_{sk}(x^*)$ 的信息. 所有关于私钥的信息只可能来自公钥, 挑战密文和私钥泄漏谕言机, 即 pk, x^*, π^* 和 ℓ 比特的私钥泄漏量. 根据平均最小熵的性质并结合 y^* 仅有 2^{τ} 个可能取值和 $\tilde{\mathsf{H}}_{\infty}(\mathsf{H}_{sk}(x^*) \mid (pk, x^*)) \geq \nu$ (对于所有 pk 和 $x^* \in X \backslash L$ 都成立) 这一事实, 可得:

$$\begin{aligned}
\tilde{\mathsf{H}}_{\infty}(\mathsf{H}_{sk}(x^*)|\mathsf{view}'_{\mathcal{A}}) &= \tilde{\mathsf{H}}_{\infty}(\mathsf{H}_{sk}(x^*)|pk, x^*, \ell\text{-leakage}, y^*) \\
&\geq \tilde{\mathsf{H}}_{\infty}(\mathsf{H}_{sk}(x^*)|pk, x^*) - \ell - \tau \\
&\geq \nu - \ell - \tau
\end{aligned}$$

因此, 利用 $(\nu - \ell - \tau, \epsilon_2)$-平均强随机性提取器 $\mathsf{ext} : \Pi \times \{0,1\}^d \to \{0,1\}^{\kappa}$ 从信息源 $\mathsf{H}_{sk_1}(x^*)$ 中提取的随机串 $\mathsf{ext}(\mathsf{H}_{sk}(x^*), s^*)$ 与均匀分布的统计距离不超过可忽略量 ϵ_2. 故有:

$$\Pr[S_5] - \Pr[S_6] \leq \epsilon_2$$

在游戏 Game_6 中, 挑战密文与加密的消息是完全独立的. 因此:

$$\Pr[S_6] = 1/2$$

综上, 定理 5.4得证! □

引理 5.2说明敌手选择一个非良生成密文进行解密查询并通过 OT-LF 验证的概率是可忽略的. 根据 OT-LF 的“隐没性”, 在解密查询 $c = (x, s, \psi, y, t_c)$ 中, OT-LF 的标签 $t = ((x, s, \psi), t_c)$ 是一个非单射标签的概率是可忽略的. 若 $x \in X \backslash L$, 根据 HPS 的 $\mathsf{universal}_1$ 性质, $\pi = \mathsf{H}_{sk_1}(x)$ 的信息熵至少为 $\log(1/\epsilon_1)$. 由于挑战密文和解密查询泄漏私钥的信息量是有限的, 从而保证了在非良生成密文的解密查询中, π 的平均最小熵依然是很高的. 由于 OT-LF 工作在单射模式, 所以敌手能够计算出正确的 y 并通过 OT-LF 验证的概率是可忽略的, 从而拒绝回答非良生成密文的解密结果. 这与直接对 $x \in X \backslash L$ 的密文拒绝解密查询的效果是一样的, 从而保证了 Game_4 和 Game_5 两个游戏的不可区分性.

下面介绍引理 5.2的详细证明过程.

证明 令事件 F 表示在游戏 Game_4 中, 存在一个解密询问 $c = (x, s, \psi, y, t_c)$ 使得 $t = ((x, s, \psi), t_c)$ 既不是单射标签也不是重用标签. 则:

$$\Pr[\mathrm{bad}_x] = \Pr[\mathrm{bad}_x \wedge F] + \Pr[\mathrm{bad}_x \wedge \overline{F}] \leq \Pr[F] + \Pr[\mathrm{bad}_x|\overline{F}]$$

假设敌手 $\mathcal{A}$ 最多询问 $Q(\kappa)$ 次解密谕言机, OT-LF 是一次有损过滤器, HPS 是 ϵ_1-$\mathsf{universal}_1$ 哈希证明系统. 下面分别证明公式 (5.1)和公式 (5.2)成立.

$$\Pr[F] \leq Q(\kappa) \cdot \mathsf{Adv}_{\mathcal{B}_3}(\kappa) \tag{5.1}$$

$$\Pr[\mathrm{bad}_x|\overline{F}] \leq \frac{Q(\kappa)2^{\ell+\tau+\kappa}}{2^{\nu} - Q(\kappa)} \tag{5.2}$$

其中 $\nu = \log(1/\epsilon_1)$.

给定有损过滤器的公钥 ek^*, $\mathcal{B}_3$ 通过模拟 $\mathcal{A}$ 在游戏 Game_4 中的环境来攻击有损过滤器的“隐没性”. 除了令 $ek = ek^*$ 外, $\mathcal{B}_3$ 按照游戏 Game_4 中的方式来生成公钥 pk 的其他参数. 值得注意的是, $\mathcal{B}_3$ 知道 PKE 的私钥, 因此可以正确回答敌手 $\mathcal{A}$ 的解密查询和私钥泄漏查询. 为了模拟挑战密文 (其中过滤器的标签必须是有损

的), $\mathcal{B}_3$ 通过一次有损过滤器提供的谕言机获得辅助标签 $t_a^* = (x^*, s^*, \psi^*)$ 对应的有损标签 t_c^*. 最后, $\mathcal{B}_3$ 随机选择 $i \in \{1, \ldots, Q(k)\}$, 并从 $\mathcal{A}$ 的第 i 个解密询问中提取相应的过滤器标签 $t = ((x, s, \psi), t_c)$. 很显然, 如果事件 F 发生了, 则至少以 $1/Q(\kappa)$ 的概率, t 是一个非单射标签. 也就是说 $\Pr[F] \leq Q(\kappa) \cdot \mathsf{Adv}_{\mathcal{B}_3}(\kappa)$. 因此, 公式 (5.1)成立.

在事件 F 未发生的前提下, 假设 $c = (x, s, \psi, y, t_c)$ 是第一个令事件 bad_x 发生的解密查询, 即 $x \in X \backslash L$, $\Pi = \mathsf{LF}_{ek,t}(\mathsf{H}_{sk_1}(x))$ 且 $t = ((x, s, \psi), t_c)$ 是单射标签. 将敌手提交第一个非良生成密文之前获得的所有信息记做 $\mathsf{view}_{\mathcal{A}}$. 注意到, 敌手只可能从公钥 pk_1, 挑战密文 c^* 和 ℓ 比特的私钥泄漏中获得有关私钥的信息. 由此可得

$$\begin{aligned} \tilde{\mathsf{H}}_\infty(\mathsf{H}_{sk}(x)|\mathsf{view}_{\mathcal{A}}) &= \tilde{\mathsf{H}}_\infty(\mathsf{H}_{sk}(x)|pk, x, c^*, \ell\text{-leakage}) \\ &\geq \tilde{\mathsf{H}}_\infty(\mathsf{H}_{sk}(x)|pk, x, c^*) - \ell \\ &\geq \mathsf{H}_\infty(\mathsf{H}_{sk}(x)|pk) - \ell - \tau - \kappa \qquad (5.3) \\ &\geq \nu - \ell - \tau - \kappa \qquad (5.4) \end{aligned}$$

其中公式 (5.3)的结论依据以下事实: 在挑战密文 c^* 中, 仅有 ψ^* 和 y^* 两部分可能泄漏私钥信息, 而 ψ^* 和 y^* 分别只有 2^κ 和 2^τ 个可能的取值. 特别指出, t_c^* 可能泄漏私钥的信息完全取决于 ψ^*, 因为 $t_c^* = \mathsf{OTLF.LTag}(td, (x^*, s^*, \psi^*))$ 可以看做是 ψ^* 的函数. 公式 (5.4)的结果依据哈希证明系统的性质, 也就是说对于任意公钥 pk 及 $x \in X \backslash L$, 都有 $\mathsf{H}_\infty(\mathsf{H}_{sk}(x)|(pk, x)) \geq \log(1/\epsilon_1) = \nu$. 因为有损过滤器工作在单射模式下, 所以 $\tilde{\mathsf{H}}_\infty(\mathsf{LF}_{ek,t}(\mathsf{H}_{sk}(x))|\mathsf{view}_{\mathcal{A}}) \geq \nu - \ell - \tau - \kappa$. 这说明在游戏 Game_4 中, 解密规则接受第一个非良生成密文的概率最多为 $2^{\ell+\tau+\kappa}/2^\nu$. 通过被拒绝解密, 敌手每次最多可以排除一个可能的 y 值, 所以第 i 个非良生成密文被接受的概率最多为 $2^{\ell+\tau+\kappa}/(2^\nu - i + 1)$. 因为 $\mathcal{A}$ 最多询问 $Q(\kappa)$ 次解密谕言机, 所以

$$\Pr[\mathsf{bad}_C | \overline{F}] \leq \frac{Q(\kappa) 2^{\ell+\tau+\kappa}}{2^\nu - Q(\kappa)}$$

因为 $\ell \leq \nu - \kappa - \tau - \omega(\log \kappa)$, 所以上面的概率是可忽略的. 因此, 公式 (5.2)成立.

根据公式 (5.1)和公式 (5.2), 可以直接得到引理 5.2的结论. 证毕! □

笔记 通过定理 5.4, 可以知道通用构造 5.6允许私钥泄漏的信息量最大为 $\ell = \log 1/\epsilon_1 - \kappa - \tau - \omega(\log \kappa)$, 与哈希证明系统的参数 ϵ_1 和一次损耗过滤器的参数 τ 密切相关. 若哈希证明系统"足够强", 即哈希证明系统作用在元素 $x \in X \backslash L$ 上的哈希值在空间 Π 上几乎是均匀分布的, 则有 $\epsilon_1 \approx 1/|\Pi|$. 在这种情况下, $\nu = \log(1/\epsilon_1) \approx \log |\Pi|$. 用 L 表示哈希证明系统的私钥长度. 一般地, 一个 universal$_1$ 哈希证明系统的哈希值空间大小 $\log |\Pi|$ 要小于甚至只有私钥长度的一半. 例如, 构造 4.9中的哈希证明系统, $|\Pi| = \log q$, $L = 2 \log q$. 当私钥长度 L 足够大且参数 τ 不变时, 私钥泄漏比率接近 $(\log |\Pi|)/L$. 因此, 设计性能良好的 HPS 和 OT-LF 对于提高方案的密钥泄漏比率至关重要.

下面以 DDH 问题为例, 简要介绍如何实现所需的哈希证明系统和一次有损过滤器.

基于 $L_{\mathbf{DDH}}$ 语言的并行哈希证明系统. 在构造 4.9基础上, 可以利用并行化技术, 设计一个私钥空间足够大且哈希值空间与私钥空间比率固定的强安全 universal$_1$-HPS. 在构造 4.9中, 哈希值空间为群元素集合 $\mathbb{G}$ 且 $\epsilon_1 = 1/\log q$, 其中 q 是循环群 $\mathbb{G}$ 的阶. 利用图 5.11的并行化方法, 选择 n 个构造 4.9中的哈希证明系统的公钥 pk_i 和私钥 sk_i 可以得到一个私钥空间为 $(\mathbb{Z}_q \times \mathbb{Z}_q)^n$, 哈希值空间为 $\mathbb{G}^n$ 且 $\epsilon_1 = 1/q^n$ 的哈希证明系统.

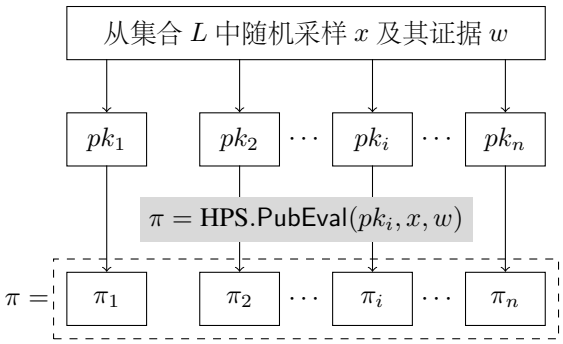


图 5.11: 并行哈希证明系统构造示意图

基于 DDH 问题的一次有损过滤器. 令 $(\mathbb{G}, q, g)$ 是一个有限循环群, $\mathsf{CH} : \{0,1\}^* \times R_{\mathsf{CH}} \to \mathbb{Z}_q$ 是一个变色龙哈希函数. 利用同态矩阵加密和变色龙哈希函数的思想, 可以设计一个 $(\mathbb{Z}_q^n, \log q)$-一次有损过滤器, 其标签空间与变色龙哈希函数的定义域相同, 即 $T = \{0,1\}^* \times R_{\mathsf{CH}}$. 设计的基本思路是构造一个如图 5.12所示的公钥矩阵 E, 其中 $r_i, s_i \xleftarrow{R} \mathbb{Z}_q$, b^* 可以看作是一个由变色龙哈希函数 $b^* = \mathsf{CH}(t_a^*, t_c^*)$ 计算而来的嵌入公钥矩阵 E 中的有损标签.

$$\mathbf{E} = \begin{pmatrix} g^{r_1 s_1} \cdot g^{-b^*} & g^{r_1 s_2} & \cdots & g^{r_1 s_n} \\ g^{r_2 s_1} & g^{r_2 s_2} \cdot g^{-b^*} & \cdots & g^{r_2 s_n} \\ \vdots & \vdots & \ddots & \vdots \\ g^{r_n s_1} & g^{r_n s_2} & \cdots & g^{r_n s_n} \cdot g^{-b^*} \end{pmatrix}$$

图 5.12: 一次有损过滤器的公钥矩阵

对于 OT-LF 定义域上的任意元素 $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{Z}_q^n$ 和任意标签 $b \in \mathbb{Z}_q$, OT-LF 的运算方式为 $y = \mathbf{x} \cdot (\mathbf{E} \otimes g^{b\mathbf{I}})$. 其中 $\mathbf{I}$ 表示 $n \times n$ 阶单位矩阵, 运算符 $\otimes$ 表示矩阵对应位置元素两两相乘. 对于矩阵 $\mathbf{E} = (E_{i,j}) \in \mathbb{G}^{n \times n}$, $\mathbf{x} \cdot \mathbf{E}$ 的运算方式为

$$\mathbf{x} \cdot \mathbf{E} = \left(\Pi_{i=1}^n E_{i,1}^{x_i}, \Pi_{i=1}^n E_{i,2}^{x_i}, \ldots, \Pi_{i=1}^n E_{i,n}^{x_i} \right)$$

由此可见, 当 $b = b^*$ 时, OT-LF 的值由 $g^{\sum_{i=1}^n r_i x_i}$ 完全确定, 此时 OT-LF 工作在有损模式下. 对于有损标签, OT-LF 的值泄漏 $\vec{x}$ 的信息量不超过 $\log q$ 比特. 当 $b \neq b^*$ 时, x_i 由 $g^{\sum_{i=1}^n r_i x_i} \cdot g^{(b-b^*)x_i}$ 完全确定, 此时 OT-LF 工作在单射模式下. 此外, 对于任意 $t_a \in \{0,1\}^*$, 可以利用变色龙哈希函数的陷门及 (t_a^*, t_c^*) 计算出另一个有损标签 (t_a, t_c) 使得 $\mathsf{CH}(t_a, t_c) = \mathsf{CH}(t_a^*, t_c^*)$.

笔记 当群空间固定时, 由于并行哈希证明系统的私钥长度为 $2n \log q$ 比特, 而 OT-LF 在有损模式下的像空间大小仅为 $\log q$, 所以当 n 增大时, 基于上述 HPS 和 OT-LF 的实例化方案允许的私钥泄漏量可以接近 $\log q^n$, 因此, 实例化的 LR-CCA 安全 PKE 方案的密钥泄漏比率可以达到 $1/2 - o(1)$. 类似地, 基于 DCR 问题也可构造具有相同密钥泄漏比率的 LR-CCA 安全 PKE 方案. 然而, 要达到最优密钥泄漏比率 $1 - o(1)$, 则需要一些特殊结构的困难问题, 如加强的子群不可区分问题, 来设计哈希证明系统及一次有损过滤器. 读者可以参考文献 [71] 了解具体的构造方法.

5.1.3.2 基于规则有损函数的 LR-CCA 安全 KEM

2018 年, 陈等 [313, 72] 提出规则有损函数的概念并用于设计抗泄漏密码学原语, 包括抗泄漏单向函数、抗泄漏消息认证码、抗泄漏密钥封装方案等. 下面, 介绍规则有损函数的形式化定义、构造及其应用.

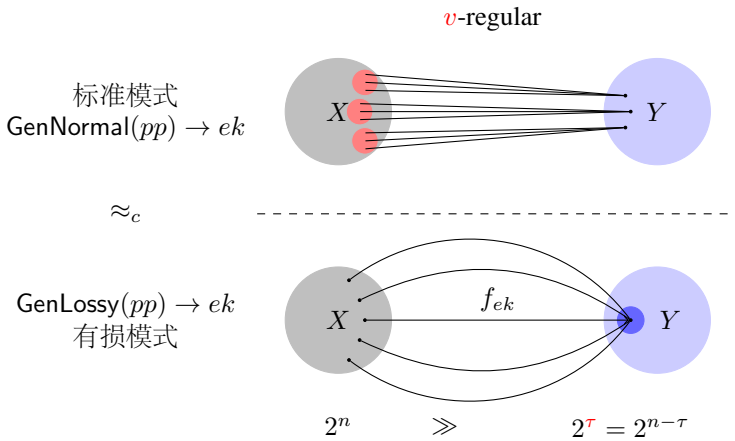


图 5.13: 规则有损函数示意图

规则有损函数. 规则有损函数是一种弱化的有损陷门函数和一次有损过滤器, 既不需要求逆陷门, 也不需要单射

模式. 与单射模式相对应的是标准模式, 也就是规则有损模式. 图 5.13给出了规则有损函数的示意图. 在 ν-标准模式中, 每个像都有 2^{ν} 相同大小的原像空间. 相应地, 像空间也缩减为 $2^{n-\nu}$. 在 τ-有损模式中, 像空间大小仅为 2^{τ}, 并且 $2^{n} \gg 2^{\tau}$. 下面介绍规则有损函数 RLF 的形式化定义.

定义 5.4 (规则有损函数)

假设定义域为 $2^{n(\kappa)}$, 其中 $n(\kappa) = \mathsf{poly}(\kappa)$. 定义 $2^{\nu(\kappa)} \leq 2^{n(\kappa)}$ 表示非单射集合大小, $2^{\tau(\kappa)} \leq 2^{n(\kappa)}$ 表示像空间大小. 一个 (ν, τ)-RLF 由以下 4 个 PPT 算法组成并满足以下性质:

- $\mathsf{Setup}(1^{\kappa})$: 以安全参数 1^{κ} 为输入, 输出公开参数 pp, 其中 pp 包含求值公钥空间 EK, 定义域 X 和值域 Y 的描述.
- $\mathsf{GenNormal}(pp)$: 以公开参数 pp 为输入, 输出求值公钥 ek. $f_{ek}(\cdot) : X \to Y$ 是一个 ν-规则函数, 即像空间上的每个元素有 2^{ν} 个原像与之对应.
- $\mathsf{GenLossy}(pp)$: 以公开参数 pp 为输入, 输出求值公钥 ek. $f_{ek}(\cdot) : X \to Y$ 是一个有损函数, 像空间最大为 2^{τ}, 损耗定义为 $n - \tau$.
- $\mathsf{Eval}(ek, x)$: 以求值公钥 ek 和原像 $x \in X$ 为输入, 输出 $y \leftarrow f_{ek}(x)$.
- 模式不可区分性: 对于任意公开参数 $pp \leftarrow \mathsf{Setup}(1^{\kappa})$, $\mathsf{GenNormal}(pp)$ 和 $\mathsf{GenLossy}(pp)$ 输出的求值公钥在计算意义下不可区分.

♣

笔记 规则有损函数可以看作是有损函数的一般化形式. 当 $\nu = 1$ 时, 规则有损函数即是有损函数. "规则有损 (regular lossy)" 这一概念在 [315, 316] 等文献中也有介绍. 但是与本书中的概念区别较大. 前者要求有损模式是规则有损的, 而本书要求的是在标准模式中是 (近似) 规则有损的. 对于一个近似规则有损函数, 在标准模式下, 函数值的熵几乎保存了原像的熵, 可以很容易得到如下的引理:

引理 5.3

假设 $f : D \to R$ 是一个 ν-到-1 规则函数, X 是定义域 D 上的一个随机变量, 则有:

$$\mathsf{H}_{\infty}(f(X)) \geq \mathsf{H}_{\infty}(X) - \log \nu$$

♡

全除一规则有损函数. 类似于全除一有损函数 (ABO-LF), 规则有损函数也可以推广为全除一规则有损函数 (ABO-RLF), 其形式化定义如下:

定义 5.5 (全除一规则有损函数)

在定义中, 参数 n, ν, τ 的含义同规则有损函数, B 表示分支空间. 一个 (ν, τ)-ABO-RLF 包含以下 3 个 PPT 算法并满足以下性质:

- $\mathsf{Setup}(1^{\kappa})$: 以安全参数 1^{κ} 为输入, 输出公开参数 pp, 其中 pp 包含对求值密钥空间 EK, 分支空间 B, 定义域 X 和值域 Y 的描述.
- $\mathsf{KeyGen}(pp, b^*)$: 以公开参数 pp 和分支 $b^* \in B$ 为输入, 输出求值密钥 ek. 对于任意 $b \neq b^*$, $f_{ek,b}(\cdot)$ 是一个从 X 到 Y 的 ν-规则函数, 而 $f_{ek,b^*}(\cdot)$ 是一个从 X 到 Y 的有损函数, 其像空间大小最多为 2^{τ}.
- $\mathsf{Eval}(ek, b, x)$: 以求值密钥 ek, 分支 $b \in B$ 和 $x \in X$ 为输入, 输出 $y \leftarrow f_{ek,b}(x)$.
- 隐藏有损分支性质: 对于任意 $b_0^*, b_1^* \in B \times B$, $\mathsf{KeyGen}(pp, b_0^*)$ 输出的求值密钥 ek_0 和 $\mathsf{KeyGen}(pp, b_1^*)$ 输出的求值密钥 ek_1 在计算上是不可区分的.

♣

图 5.14给出了全除一有损函数的示意图. 由图示可知, 每个求值函数都会额外输入一个分支 b, 并且有损分支 b^* 隐藏于求值密钥 ek 中. 当 $b = b^*$ 时, 函数才处于有损模式, 其他情况都是规则模式.

一次规则有损过滤器. 在 ABO-RLF 中, 求值公钥 ek 完全确定了有损分支 b^* 的值. 因此, 在归约证明中, 有损分支需要提前选取, 并不适应于自适应攻击的敌手环境. 类似 OT-LF, 可以将 RLF 推广到一次规则有损过滤器 (OT-RLF). OT-RLF 的形式化定义如下:

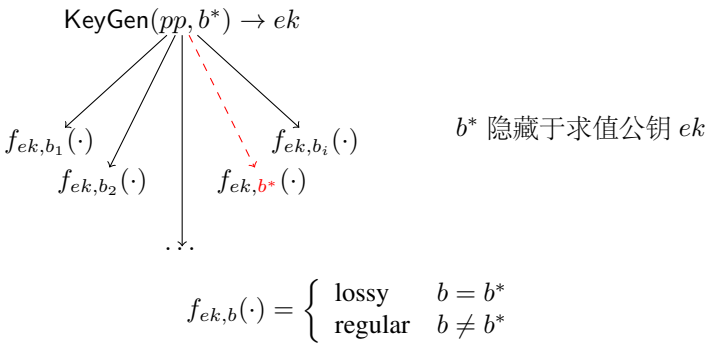


图 5.14: 全除一规则有损函数示意图

定义 5.6 (一次规则有损过滤器)

一个 (ν, τ)-OT-RLF 包含以下 4 个 PPT 算法并满足以下性质:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出公开参数 pp, 其中 pp 包含求值密钥空间 EK, 分支集合 $B = B_c \times B_a$ (其中, B_c 是核心分支集合, B_a 是辅助输入分支集合), 定义域 X 和值域 Y 的描述.
- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入, 输出求值密钥 ek 和陷门 td. 分支集合 B 包含两个不相交的子集: 规则分支集合 B_{normal} 和有损分支集合 B_{lossy}. 对于规则分支集合中的任意分支 $b \in B_{\text{normal}}$, $f_{ek,b}(\cdot)$ 确定了一个从 X 到 Y 的 ν-规则函数. 对于有损分支集合中的任意分支 $b \in B_{\text{lossy}}$, $f_{ek,b}(\cdot)$ 确定了一个从 X 到 Y 且像空间最大为 2^τ 的有损函数.
- $\mathsf{SampLossy}(td, b_a)$: 以陷门 td 和辅助分支 b_a 为输入, 输出核心分支 b_c, 使得 $b = (b_c, b_a)$ 是集合 B_{lossy} 中的一个有损分支.
- $\mathsf{Eval}(ek, b, x)$: 以求值密钥 ek, 分支 $b \in B$ 和元素 $x \in X$ 为输入, 输出 $y \leftarrow f_{ek,b}(x)$.
- 不可区分性: 对于任意辅助分支 $b_a \in B_a$, 由算法生成的核心分支 $b_c \leftarrow \mathsf{SampLossy}(td, b_a)$ 与随机选取的核心分支 $b_c \xleftarrow{R} B_c$ 在计算意义下是不可区分的.
- 隐没性: 对于任意 PPT 敌手, 在给定一个有损分支的条件下, 再生成一个新的有损分支是困难的.

笔记 如图 5.15所示, 规则有损函数的几个相关概念之间存在一定的联系. ABO-RLF 是 RLF 的推广. 实际上, 若存在一族 (ν, τ)-ABO-RLF 函数, 只需要把 ABO-RLF 的分支标签作为 RLFs 的求值密钥参数, 即可以得到一族 (ν, τ)-RLF 函数. 相反, 类似 LTF 与 ABO-LTF 的转化关系 [31], 若存在一族 (ν, τ)-RLF 函数, 则必然存在一族分支集合为 $B = \{0,1\}$ 的 (ν, τ)-ABO-RLF 函数. 进一步, 可以推广到一族分支集合为 $B = \{0,1\}^d$ 的 $(\nu, d\tau)$-ABO-RLF 函数. 此外, 根据文献 [71], 一族 (ν, τ)-ABO-RLF 函数结合一个变色龙哈希函数, 当分支集合与哈希值空间相匹配时, 可以构造一族 (ν, τ)-OT-RLF 函数.

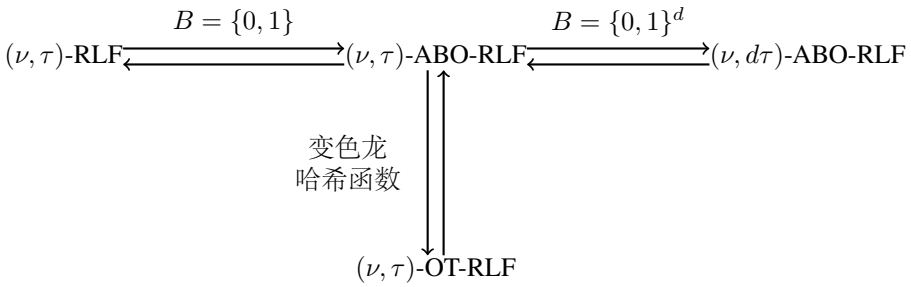


图 5.15: 规则有损函数相关概念之间的关系

ABO-RLF 的构造

首先介绍基于 DDH 问题和 DCR 问题的 ABO-RLFs 函数族的两种具体构造. 在介绍基于 DDH 的 ABO-RLF 构造之前, 先回顾群上伪随机的隐藏矩阵生成算法 GenConceal. 该算法的输入是两个正整数 n 和 m (其中 $n \geq m$), 输出是一个群元素构成的秩为 1 的 $n \times m$ 矩阵 $\mathbb{G}^{n\times m}$ 且与随机选取的 $n \times m$ 矩阵不可区分. 该算法的具体执行过程如下:

- 随机选择两个向量 $\mathbf{r} = (r_1, \ldots, r_n) \xleftarrow{R}{}^{n}$ 和 $\mathbf{s} = (s_1, \ldots, s_m) \xleftarrow{R} \mathbb{Z}_q^m$.
- 令 $\mathbf{V} = \mathbf{r} \otimes \mathbf{s} = \mathbf{r}^t\mathbf{s} \in \mathbb{Z}_q^{n\times m}$ 表示 $\mathbf{r}$ 与 $\mathbf{s}$ 的外积.
- 输出 $\mathbf{C} = g^{\mathbf{V}} \in \mathbb{G}^{n\times m}$ 作为隐藏矩阵.

根据文献 [31], 以下引理成立:

引理 5.4

令 $n, m = \mathrm{poly}(\kappa)$. 如果 DDH 假设成立, 则矩阵 $\mathbf{C} = g^{\mathbf{V}} \leftarrow \mathsf{GenConceal}(n, m)$ 在空间 $\mathbb{G}^{n\times m}$ 上是伪随机的.

♡

构造 5.7 (基于 DDH 的 ABO-RLF 构造)

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 运行 $(\mathbb{G}, q, g) \leftarrow \mathsf{GenGroup}(\kappa)$, 输出公开参数 $pp = (\mathbb{G}, q, g)$ 和分支空间 $B = \mathbb{Z}_p$.
- $\mathsf{KeyGen}(pp, b^*)$: 以公开参数 pp 和分支 $b^* \in \mathbb{Z}_q$ 为输入, 调用算法 $\mathsf{GenConceal}(n, m)$ 生成矩阵 $\mathbf{C} = g^{\mathbf{V}} \in \mathbb{G}^{n\times m}$. 输出求值密钥 $ek = g^{\mathbf{Y}} = g^{\mathbf{V}-b^*\mathbf{I}'}$, 其中 $\mathbf{I}' \in \mathbb{Z}_q^{n\times m}$, 即第 i 列向量是标准的单位向量 $\mathbf{e}_i \in \mathbb{Z}_q^n$, 其中 $i \leq n$, 而余下的列向量为零向量.
- $\mathsf{Eval}(ek, b, \mathbf{x})$: 以求值密钥 $ek = g^{\mathbf{Y}}$, 分支 $b \in \mathbb{Z}_q$ 和元素 $\mathbf{x} \in \mathbb{Z}_q^n$ 为输入, 输出 $\mathbf{y} = g^{\mathbf{x}(\mathbf{Y}+b\mathbf{I}')} = g^{\mathbf{x}(\mathbf{V}+(b-b^*)\mathbf{I}')} \in \mathbb{G}^m$.

♣

引理 5.5

如果 DDH 假设成立, 则构造 5.7 是一族 $(q^{n-m}, \log q)$-ABO-RLF 函数, 其中 $n > 1$.

♡

证明 对于任意 $b \neq b^*$, $(\mathbf{V}, b)$ 确定了一个 q^{n-m}-到-1 函数, 这是因为矩阵 $(\mathbf{Y} + b\mathbf{I}')$ 的秩是 m, 对于每个 $y \in \mathbb{G}^m$, 其解空间大小为 q^{n-m}. 当 $b = b^*$ 时, 每个输出结果 $\mathbf{y}$ 的形式是 $g^{r'\mathbf{s}}$, 其中 $r' = \mathbf{x}\mathbf{r}^t \in \mathbb{Z}_q$. 因为 $\mathbf{s}$ 由函数索引 $\mathbf{V}$ 确定, 所以由 $(\mathbf{V}, b^*)$ 确定的每个函数最多有 q 个不同的输出结果. 因此, 损耗为 $(n-1)\log q$.

在 DDH 假设下, 通过归约可以证明隐藏有损分支性质: 对于任意分支 $b^* \in \mathbb{Z}_q$, $\mathsf{Gen}(pp, b^*)$ 输出的求值密钥矩阵与 $\mathbb{G}^{n\times m}$ 上的随机矩阵在计算意义下是不可区分的.

笔记 在构造中, 参数 n 用于控制定义域的大小, 而参数 m 用于调节 ABO 分支的规则性. 当 $m = n$ 时, ABO 分支是单射的, 故上述构造方案变成了标准的 ABO-LF.

在基于 DDH 的 ABO-LTF 构造中 [31], 定义域限制在 $\{0,1\}^n$ 且 m 必须大于 n 才能保证函数可求逆. 在上述构造中, 由于 ABO-RLF 不需要求逆的性质, 在不改变求值密钥矩阵大小的情况下, 定义域可以由 $\{0,1\}^n$ 扩展到 $\mathbb{Z}_q^n$. 特别地, ABO-RLF 不需要单射性质, 所以矩阵参数 m 可以小于 n. 在基于矩阵的构造中, 求值密钥大小和计算开销的复杂性取决于参数 n 和 m. 因此, 与基于 DDH 的 ABO-LTF 相比, ABO-RLF 允许的输入空间更大, 计算效率更高.

利用文献 [317] 中的方法, 可以将上述基于 DDH 的构造方案扩展到基于 eDDH 问题的构造, 而 eDDH 是一类问题, 包含了 DDH、QR 和 DCR 等问题. 所以, 上述构造也蕴含了基于 DCR 的 ABO-RLF. 尽管如此, 直接基于 DCR 问题, 可以构造出更高效的 ABO-RLF 方案. 具体如下:

构造 5.8 (基于 DCR 的 ABO-RLF 构造)

- Setup(1^κ): 运行 $N \leftarrow \mathsf{GenModulus}(1^\kappa)$ 生成 RSA 模 N, 随机选择 $z \in \mathbb{Z}_N$ 并计算 $g = z^{2N} \mod N^2$, 输出公开参数 $pp = (N, y)$, 并令分支空间为 $B = \mathbb{Z}_N$.
- KeyGen(pp, b^*): 以公开参数 pp 和有损分支 $b^* \in \mathbb{Z}_N$ 为输入, 随机选择 $r \in \mathbb{Z}_N$, 计算并输出求值密钥 $ek = g^r(1+N)^{-b^*}$.
- Eval(ek, b, x): 以求值密钥 ek, 分支 $b \in \mathbb{Z}_N$ 和元素 $x \in \{0, 1, \cdots, \lfloor N^2/4 \rfloor\}$ 为输入, 输出 $y = [ek/(1+N)^b]^x = g^{rx}(1+N)^{(b-b^*)x} \in \mathbb{Z}_{N^2}$.

♣

引理 5.6

如果 DCR 假设成立, 则构造 5.8是一族 $(1, \phi(N)/4)$-ABO-RLF 函数.

♡

证明 对于任意 $b \neq b^*$, $f_{ek,b}$ 是一个单射函数, 这是因为 g 以压倒性的概率是 $2N$ 次剩余群的一个生成元. 令 ϕ 表示欧拉函数. 则 g 的阶至少为 $\phi(N)/4$, $g^r(1+N)^{b-b^*}$ 的阶至少为 $N\phi(N)/4$. 当 $b = b^*$ 时, 每个输出结果 g^{rx} 是一个 $2N$ 次剩余元素. 因此, 所有像元素至多有 $\phi(N)/4$ 个, 损耗至少为 $\log N$.

隐藏有损分支性质源于基于 DCR 假设的 Paillier 加密方案 [281] 安全性: 算法 $\mathsf{KeyGen}(pp, b^*)$ 的输出结果实际上是 Paillier 方案选择随机数 r 加密消息 b^* 的一个密文. 因此, 对于任意 $b_0^*, b_1^* \in \mathbb{Z}_N$, $\mathsf{KeyGen}(pp, b_0^*)$ 的输出结果和 $\mathsf{KeyGen}(pp, b_1^*)$ 的输出结果在计算意义下是不可区分的. 由于规则有损参数 $\nu = 1$, 所以构造 5.8实际上是一个 ABO-LF 函数族.

下面介绍 ABO-RLF 的通用构造方法.

2012 年, Wee [318] 提出利用对偶哈希证明系统构造有损陷门函数. 假设 $(\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$ 是对偶哈希证明系统的公开参数, 其中 $\mathsf{H} : X \times SK \to \Pi$. 基于对偶哈希证明系统的有损陷门函数 f 的构造方式如公式 (5.5)所示.

$$f_x(sk) = \alpha(sk) || \mathsf{H}_{sk}(x) \tag{5.5}$$

在上述构造中, 哈希证明系统的私钥 $sk \in SK$ 充当了有损陷门函数的定义域, 而子集成员 $x \in X$ 充当了函数的求值密钥. 当 $x \in X \setminus L$ 时, 根据对偶哈希证明系统的可逆性质, f_x 是单射函数且可逆; 根据对偶哈希证明系统的仿射性质, 当 $x \in L$ 时, 函数 f_x 是有损函数. 基于子集成员判定问题, 这两种模式在计算意义下是不可区分的.

利用上述构造方式, 可以基于任意的哈希证明系统构造规则有损函数. 由于规则有损函数比有损陷门函数的性质要弱, 仅需要哈希证明系统的投射性质即可, 并不需要哈希证明系统其他额外的性质, 比如平滑性、一致性、可逆性等. 令 $(\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$ 是哈希证明系统的一个公开参数. 对于任意 $x \in X \setminus L$, 假设 $f_x(sk) = \alpha(sk) || \mathsf{H}_{sk}(x)$ 是一个从定义域 SK 到值域 Π 的 ν-到-1 函数, 则有以下引理:

引理 5.7

如果 SMP 假设成立, 公式 (5.5)是一族 $(\nu, \log |\mathrm{Img}(\alpha)|)$-RLF.

♡

证明 标准模式的正确性源于 $f_x(\cdot)$ 是一个 ν-到-1 函数. 有损模式的损耗性源于哈希证明系统的投射性质. 对于任意 $x \in L$, $\mathrm{Img}(f_x) = \mathrm{Img}(\alpha)$. 两种模式的不可区分性可以归约到 SMP 的困难性.

至此, 利用哈希证明系统构造全除一规则有损函数可以通过以下两步实现: (1) 利用哈希证明系统构造一族规则有损函数; (2) 将具有二元分支空间 $\{0, 1\}$ 的规则有损函数放大到分支空间为 $\{0, 1\}^d$ 的全除一规则有损函数. 然而, 第二步扩大分支空间的效率并不高, 需要进行 d 次独立地规则有损函数计算, 且损耗量也会降低. 在上述构造中, 哈希证明系统所基于的子集成员判定问题没有任何代数结构限制. 如果子集成员判定问题具有一定的代数结构, 则可以构造更加高效的 ABO-RLF 函数. 下面, 首先介绍具有代数结构的子集成员判定问题, 即 ASMP 问题.

定义 5.7 (ASMP 问题)

ASMP 问题是一种特殊的子集成员判定问题 (X, L), 满足以下几个性质:

1. X 是一个有限阿贝尔群, L 是 X 的一个子群.
2. 商群 $H = X/L$ 是一个阶为 $p = |X|/|L|$ 的循环群.

基于上述性质, 可以推出以下两个实用的结论:

- 令 $\bar{a} = aL$, 其中 $a \in X \backslash L$ 是 H 的一个生成元, 则陪集 $(aL, 2aL, \ldots, (p-1)aL, paL = L)$ 构成了 X 的一个划分.
- 对于任意 $x \in L$, $ia + x \in X \backslash L$, 其中 $1 \leq i < p$.

♣

ASMP 问题的困难性同 SMP 问题一样, 要求集合 L 和 $X \backslash L$ 上的元素分布在计算意义下是不可区分的. L 的密度定义为 $\rho = |L|/|X|$. 当 ρ 是可忽略时, $U_L \approx_c U_{X\backslash L}$ 等价于 $U_L \approx_c U_X$. 此时, $U_{X\backslash L}$ 和 U_X 统计距离接近. 当 ρ 已知时, U_X 可由 $U_L, U_{X\backslash L}$ 和 ρ 重构出来, 所以 $U_L \approx_c U_{X\backslash L}$ 蕴含了 $U_L \approx_c U_X$.

笔记 ASMP 问题具有一般性, 可由 DDH、d-LIN、QR 和 DCR 等问题构造. 此外, ASMP 问题可以看作是满足性质 2 的加强子群成员判定问题. 在实际应用中, 性质 2 可以放宽至 H 包含一个循环子群. 子群不可区分问题 (SIP 问题) 是 Brakerski 和 Goldwasser [319] 于 2010 年提出的. SIP 问题定义在一个阿贝尔群 X 和子群 L 上. 此外, SIP 要求 X 同构于两个群的直积, 即 $X \simeq L \times M$ 且 $\gcd(\mathrm{ord}(L), \mathrm{ord}(M)) = 1$. 2014 年, Qin 和 Liu [71] 提出加强的子群不可区分问题 (RSIP), 进一步要求 M 也是一个循环子群. 与 RSIP 问题相比, ASMP 问题仅要求商群 X/L 是循环的. 因此, ASMP 问题要强于 RSIP 问题, 按理也比 SIP 问题强, 因为 SIP 问题无法由 DDH 和 d-LIN 等问题构造. 由此可见, 代数子集成员判定假设要弱于 RSIP 和 SIP 等假设.

下面介绍基于 ASMP 问题的 ABO-RLF 构造. 该构造假设存在 ASMP 问题上的一个哈希证明系统.

构造 5.9

假设 HPS 是 ASMP 问题上的一个哈希证明系统, 则 ABO-RLF 的具体构造如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{HPS.Setup}(1^\kappa)$ 生成 HPS 的公开参数 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$, 选择商群 H 的一个随机生成元 aL, 输出公开参数 $\hat{pp} = (pp, a)$.
- $\mathsf{KeyGen}(\hat{pp}, b^*)$: 以公开参数 $\hat{pp} = (pp, a)$ 和一个有损分支 $b^* \in \mathbb{Z}_q$ 为输入, 运行 $x \leftarrow \mathsf{SampYes}(r)$ 抽取 L 中的一个随机元素, 计算求值密钥 $ek = -b^*a + x \in X$.
- $\mathsf{Eval}(ek, b, sk)$: 以求值密钥 $ek = -b^*a + x$, 分支 b 和 sk 为输入, 计算并输出 $\alpha(sk)||\mathsf{H}_{sk}(ek + ba)$. 该算法定义了 $f_{ek,b}(sk) := \alpha(sk)||\mathsf{H}_{sk}(ek + ba)$.

♣

定理 5.5

假设 $X = \{0,1\}^n$. 对于任意 $x \notin L$, 函数 $f_x(sk) = \alpha(sk)||\mathsf{H}_{sk}(x)$ 是一个 ν-规则函数. 则构造 5.9是一族基于 ASMP 问题的 $(\nu, \log|\mathrm{Img}\alpha|)$-ABO-RLF 函数.

♡

证明 根据 ASMP 问题的性质, 当 $b \neq b^*$ 时, $ek + ba = x + (b - b^*)a \notin L$. 此时, $f_{ek,b}(\cdot)$ 是一个 ν-规则函数. 当 $b = b^*$ 时, $ek + ba = x + (b - b^*)a = x \in L$. 此时, $f_{ek,b}(\cdot)$ 是一个有损函数. 在安全性方面, 隐藏有损分支性质源于代数子集成员判定问题的困难性, 即对于任意 $b_0^*, b_1^* \in \mathbb{Z}_q$, 当 $u \xleftarrow{R} X$ 时, 则有 $(-b_0^*a + x) \approx_c (-b_0^*a + u) \equiv u \equiv (-b_1^*a + u) \approx_c (-b_1^*a + x)$. 定理得证! □

规则有损函数的应用

在泄漏密码学领域, 规则有损函数具有重要的应用. 陈等 [313] 指出规则有损函数可以用于构造抗泄漏单向函数 (LR-OWF)、抗泄漏消息认证码 (LR-MAC)、抗泄漏安全的密钥封装方案 (LR-KEM) 等高级密码学原语. 下面对这些应用分别作一简要介绍.

抗泄漏单向函数. 类似有损陷门函数蕴含一个单向陷门函数, 规则有损陷门函数则蕴含一个抗泄漏单向函数.

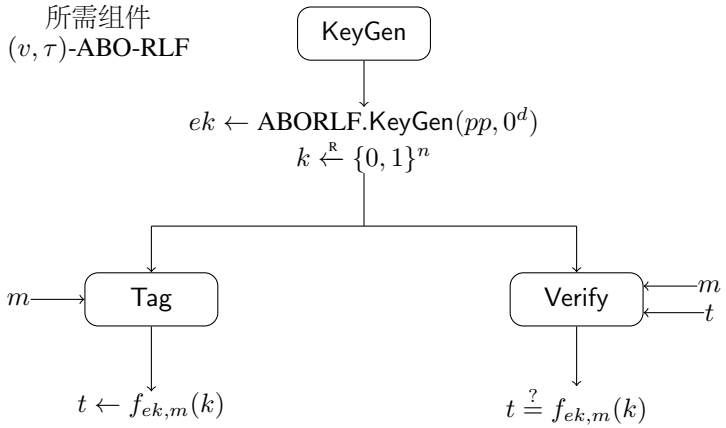


图 5.16: 抗泄漏消息认证码构造示意图

定理 5.6

假设 RLF = (Setup, GenNormal, GenLossy, Eval) 是一族定义在 $\{0,1\}^n$ 上的规则有损函数, 其有损模式的像空间大小最多为 2^τ. 则 (Setup, GenNormal, Eval) 是一个 ℓ-抗泄漏单向函数, 其中 $\ell \leq n - \tau - \omega(\log \kappa)$.

证明定理 5.6的基本思路是将规则有损函数的标准模式转化为不可区分的有损模式, 在有损模式下挑战原像 x^* 泄漏的信息量较少, 对于敌手来说剩余最小熵依然较大, 从而保证敌手正确猜测挑战原像 x^* 的概率是可忽略的. 下面介绍该定理的详细证明.

证明 通过游戏序列的方式组织证明. 令 S_i 表示事件 "$\mathcal{A}$ 在游戏 i 中成功".

Game_0: 该游戏是标准的抗泄漏单向函数游戏, 其中挑战者 $\mathcal{CH}$ 与敌手 $\mathcal{A}$ 按以下方式交互:

1. 初始化: $\mathcal{CH}$ 生成 RLF 的系统参数 $pp \leftarrow \mathsf{RLF.Setup}(1^\kappa)$ 及标准模式求值密钥 $ek \leftarrow \mathsf{GenNormal}(pp)$, 随机选择 $x^* \xleftarrow{R} \{0,1\}^n$ 并计算 $y^* \leftarrow f_{ek}(x^*)$, 然后将 (ek, y^*) 发送给敌手 $\mathcal{A}$ 作为挑战信息.
2. 泄漏询问: $\mathcal{A}$ 可以自适应地进行泄漏询问. 对于任意泄漏询问 $\langle g \rangle$, $\mathcal{CH}$ 返回 $g(x^*)$.
3. 求逆: $\mathcal{A}$ 输出 x, 如果 $x = x^*$ 则 $\mathcal{A}$ 成功.

根据定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr[S_0]$$

Game_1: 该游戏与 Game_0 唯一不同之处在于第 1 步:

1. 初始化: $\mathcal{CH}$ 生成有损模式求值密钥$ek \leftarrow \mathsf{GenLossy}(pp)$.

根据两种模式的不可区分性, 则有:

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{negl}(\kappa)$$

下面重点分析 Game_1 中的概率 $\Pr[S_1]$. 假设 ek 是任意一个由 $\mathsf{GenLossy}(pp)$ 生成的求值密钥. 对于任意敌手, 成功猜测 x^* 的概率完全取决于 x^* 的平均最小熵. 特别地, 在已知 $y^* \leftarrow f_{ek}(x^*)$ 和 x^* 的部分泄漏信息条件下, x^* 的平均最小熵为 $2^{-\tilde{\mathsf{H}}_\infty(x^*|(y^*,\mathsf{leak}))}$. 由于 $f_{ek}(\cdot)$ 的输出结果最多有 2^τ 种可能值, 泄漏量最多为 2^ℓ, 则有:

$$\tilde{\mathsf{H}}_\infty(x^*|(f_{ek}(x^*), \mathsf{leak})) \geq \mathsf{H}_\infty(x^*) - \tau - \ell = n - \tau - \ell$$

由于 $n - \tau - \ell \geq \omega(\log \kappa)$, 则 $\mathcal{A}(ek, y^*, \mathsf{leak})$ 输出 x^* 的概率是可忽略的. 该结论对于由 GenLossy 随机生成的求值密钥 ek 也成立. 这就证明了 $\Pr[S_1] = \mathsf{negl}(\kappa)$, 从而 $\Pr[S_0]$ 也是可忽略的. 定理得证! □

抗泄漏消息认证码. 消息认证码是一种重要的密码学原语, 在认证协议、公钥加密方案设计等方面具有重要的应用. 下面介绍如何利用 ABO-RLF 或者 OT-RLF 构造抗泄漏消息认证码. 利用 ABO-RLF 构造 MAC 的主要思路将 ABO-RLF 的定义域输入看作 MAC 的密钥, 而将输入的分支看作消息, 其输出作为 MAC 的标签, 如图 5.16所示.

构造 5.10 (基于 ABO-RLF 的 LR-MAC 构造)

构造所需组件是:

- 一个全除一规则有损函数 $\mathsf{ABORLF} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Eval})$.

构造 LR-MAC 如下:

- $\mathsf{Setup}(\kappa)$: 运行 $\mathsf{ABORLF.Setup}(1^\kappa)$ 生成 RLF 的公开参数 $pp = (EK, B, X, Y)$, 其中 $|X| = 2^n$ 和 $B = \{0,1\}^d$, 运行 $\mathsf{ABORLF.KeyGen}(pp, 0^d)$ 生成 ABO-RLF 的求值密钥 ek. 输出 MAC 的公开参数 $\hat{pp} = (pp, ek)$. 密钥空间、消息空间和认证码空间分别定义为 $K = X, M = B$ 和 $T = Y$.
- $\mathsf{KeyGen}(\hat{pp})$: 随机选择 $k \xleftarrow{R} X$ 作为 MAC 的密钥.
- $\mathsf{Tag}(k, m)$: 以密钥 k 和消息 m 为输入, 计算 $t \leftarrow f_{ek,m}(k)$, 输出消息 m 的认证码 (m, t).
- $\mathsf{Verify}(k, m, t)$: 如果 $t = f_{ek,m}(k)$ 则输出 1, 否则输出 0.

♣

定理 5.7

如果 ABORLF 是一个 (ν, τ)-全除一规则有损函数族, 则构造 5.10是一个 ℓ-抗泄漏消息认证码, 其中 $\ell \leq n - \tau - \log \nu - \omega(\log \kappa)$.

♡

证明定理 5.7的基本思路是利用 ABO-RLF 隐藏有损分支的性质, 将挑战消息 m^* 作为有损分支, 从而保证挑战消息的认证码 t^* 仅泄漏少量的认证密钥 k. 对于敌手伪造的消息认证码 (m, t), 由于 $m \neq m^*$, 此时 RLFs 工作在标准模式. 对于敌手来说, 认证密钥 k 依然具有较高的剩余最小熵, 从而使得敌手能够正确计算消息 m 对应的认证码 t 的概率是可忽略的. 下面给出该定理的详细证明过程.

证明 通过游戏序列的方式组织证明. 令 S_i 表示事件 "$\mathcal{A}$ 在游戏 i 中攻击成功".

Game_0: 该游戏是消息认证码的抗泄漏一次强不可伪造安全性游戏. 挑战者 $\mathcal{CH}$ 通过以下方式与敌手 $\mathcal{A}$ 交互完成游戏.

1. 承诺和初始化: 敌手 $\mathcal{A}$ 在看到公开参数前, 声明一个目标消息 m^*. $\mathcal{CH}$ 运行 $pp \leftarrow \mathsf{ABORLF.Setup}(1^\kappa)$ 和 $ek \leftarrow \mathsf{ABORLF.KeyGen}(pp, 0^d)$. $\mathcal{CH}$ 选择 $k \xleftarrow{R} X$, 计算 $t^* \leftarrow f_{ek,m^*}(k)$, 再将 $\hat{pp} = (pp, ek)$ 和 t^* 发送给 $\mathcal{A}$.
2. 询问: $\mathcal{A}$ 可以自适应地进行泄漏询问. 对于任意泄漏查询 $\langle g \rangle$, 只要泄漏量不超过 ℓ, $\mathcal{CH}$ 返回 $g(k)$ 给 $\mathcal{A}$.
3. 伪造: $\mathcal{A}$ 输出 (m, t). 如果 $m \neq m^*$ 且 $t = f_{ek,m}(k)$, 则 $\mathcal{A}$ 成功.

根据定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr[S_0]$$

Game_1: 该游戏与 Game_0 的唯一区别是 $\mathcal{CH}$ 通过运行$\mathsf{ABORLF.KeyGen}(pp, m^*)$替代 $\mathsf{ABORLF.KeyGen}(pp, 0^d)$ 生成 ek. 根据 ABO-RLF 隐藏有损分支的性质,Game_0 与 Game_1 在计算上不可区分. 因此有:

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{Adv}_{\mathcal{B}}(\kappa)$$

其中 $\mathcal{B}$ 是一个攻击 ABO-RLF 隐藏有损分支性质的敌手.

下面分析事件 S_1 的概率. 由于消息认证码是唯一的, 事件 S_1 实际上等价于 $\mathcal{A}$ 输出 (m, t), 其中 $m \neq m^*$ 且 $f_{ek,m}(k) = t$. 则消息认证码 t 的条件熵由 $\mathcal{A}$ 的视图 $\mathsf{view} = (pp, ek, m, leak, m^*, t^*)$ 决定. 具体为

$$\begin{aligned} \tilde{\mathsf{H}}_\infty(t|\mathsf{view}) &= \tilde{\mathsf{H}}_\infty(t|ek, m, leak, t^*) && (5.6) \\ &\geq \tilde{\mathsf{H}}_\infty(t|ek, m) - \ell - \tau && (5.7) \\ &\geq n - \log \nu - \ell - \tau && (5.8) \end{aligned}$$

其中, 公式 (5.6)依据事实 $t = f_{ek,m}(k)$ 由 ek、m 和 k 确定, 且 k 与 m^* 和 pp 独立. 公式 (5.7)依据引理 2.2和泄漏量的上界为 ℓ 比特以及挑战认证码 t^* 最多有 2^τ 个可能的值. 对于任意 $m \neq m^*$, $f_{ek,m}(\cdot)$ 是一个 ν-到-1 函数. 根据引理 5.3, 则有 $\mathsf{H}_\infty(f_{ek,m}(k)) \geq \mathsf{H}_\infty(k) - \log \nu$, 从而公式 (5.8)成立.

根据参数选择方式, 则有 $n - \log \nu - \ell - \tau \geq \omega(\log \kappa)$. 因此,

$$\Pr[S_1] \leq \frac{1}{2^{n - \log \nu - \ell - \tau}} \leq \mathsf{negl}(\kappa)$$

综上, 定理 5.7得证! □.

抗泄漏密钥封装方案. 类似 Qin 和 Liu 在 2013 年亚密会上提出的抗泄漏公钥加密方案的设计模式, 利用 ABO-RLF 设计 LR-CCA 安全 KEM 方案也额外需要一个哈希证明系统作为封装密钥的工具. 设计思路如图 5.17所示. 简单地说, 先利用一个哈希证明系统封装一个 (随机) 哈希证明 π, 再利用 ABO-RLF 计算 π 的知识证明 t, 类似于 Cramer-Shoup 方案的指定验证者零知识证明. 这里的 t 也可以看作是密文参数的一个消息认证码. 然而, 我们不能将工具 ABO-RLF 替换为普通的消息认证码. 这是因为 π 还要用于提取随机值作为封装密钥, 为了实现方案的可证明安全性, 知识证明 t 不能泄漏 π 太多的信息, 这就要求挑战密文中 ABO-RLF 函数是有损的, 而敌手查询的解密密文中 ABO-RLF 几乎是单射的. 与 Qin-Liu 方案的设计模式相比, 该模式具有的优势是 ABO-RLF 也可以利用哈希证明系统构造, 从而整个密钥封装方案可以基于哈希证明系统实现.

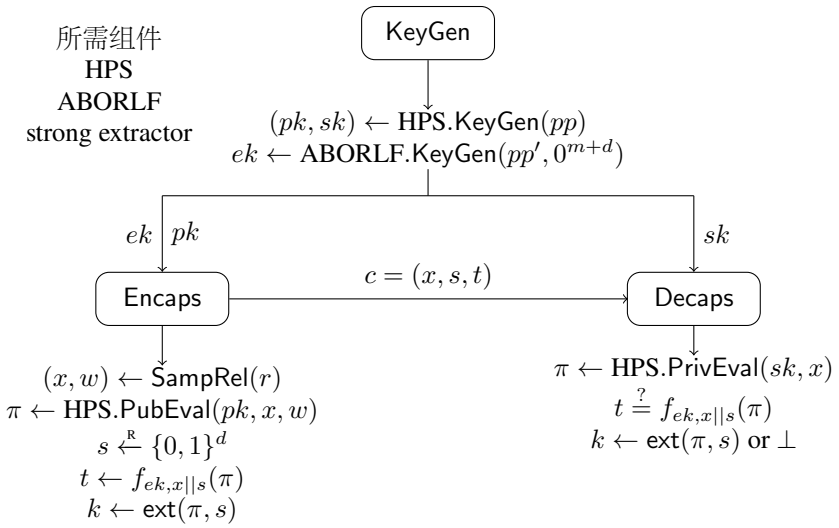


图 5.17: 抗泄漏密钥封装方案构造示意图

构造 5.11 (基于 ABO-RLF 的 LR-CCA 安全 KEM)

构造所需组件是:

- 一个 universal$_1$ 哈希证明系统 $\mathsf{HPS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PubEval}, \mathsf{PrivEval})$.
- 一个全除一规则有损函数 $\mathsf{ABORLF} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Eval})$.
- 一个平均意义强随机性提取器 $\mathsf{ext} : \Pi \times \{0,1\}^d \to \{0,1\}^\kappa$.

构造 LR-CCA KEM 如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{HPS.Setup}(1^\kappa)$ 生成 HPS 的公开参数 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$; 运行 $\mathsf{ABORLF.Setup}(1^\kappa)$ 生成 ABO-RLF 的公开参数 $pp' = (EK, B = X \times \{0,1\}^d, \Pi, T)$; 选择一个平均意义 $(n - \tau - \ell, \epsilon_2)$-强随机性提取器 $\mathsf{ext} : \Pi \times \{0,1\}^d \to \{0,1\}^\kappa$, 其中 $n = \log 1/\epsilon_1$; 输出公开参数 $\hat{pp} = (pp, pp', \mathsf{ext})$.
- $\mathsf{KeyGen}(\hat{pp})$: 首先, 拆分公开参数为 $\hat{pp} = (pp, pp', \mathsf{ext})$; 然后, 运行 $(pk, sk) \leftarrow \mathsf{HPS.KeyGen}(pp)$ 和 $ek \leftarrow \mathsf{ABORLF.Gen}(pp', 0^{m+d})$. 输出公钥 $\hat{pk} = (pk, ek)$ 和私钥 $\hat{sk} = sk$.
- $\mathsf{Encaps}(\hat{pk}; r)$: 以公钥 $\hat{pk} = (pk, ek)$ 为输入. 首先, 随机采样 $(x, w) \leftarrow \mathsf{SampRel}(r)$, 计算 $\pi \leftarrow \mathsf{HPS.PubEval}(pk, x, w)$; 然后, 选取随机种子 $s \xleftarrow{R} \{0,1\}^d$, 计算 $t \leftarrow f_{ek,x||s}(\pi)$; 输出密文 $c = (x, s, t)$ 和封装密钥 $k \leftarrow \mathsf{ext}(\pi, s)$.
- $\mathsf{Decaps}(\hat{sk}, c)$: 以私钥 $\hat{sk} = sk$ 和密文 $c = (x, s, t)$ 为输入. 首先, 计算 $\pi \leftarrow \mathsf{HPS.PrivEval}(sk, x)$; 然后, 判断 $t = f_{ek,x||s}(\pi)$ 是否成立. 如果成立, 则输出 $k \leftarrow \mathsf{ext}(\pi, s)$; 否则, 输出 $\perp$.

笔记 在构造 5.11中, 密文 (x, s) 可以看作是一个 Naor-Segev 模式的 LR-CPA 安全 KEM 方案的密文, 而 t 可以看作是构造 5.10中对消息 (x, s) 的一个消息认证码.

定理 5.8

如果 SMP 假设成立, HPS 是一个 ϵ_1-universal$_1$ 哈希证明系统, ABORLF 是一个 (ν,τ)-全除一规则有损函数, ext 是一个平均意义 $(n-\tau-\ell,\epsilon_2)$ 强随机性提取器, 则构造 5.11 中的 KEM 是 ℓ-LR-CCA 安全的, 其中 $\ell \leq n-\tau-\kappa-\log\nu-\omega(\log\kappa)$.

证明思路: 基于通用构造模式 "HPS+ABO-RLF" 的 KEM 方案和基于通用构造模式 "HPS+OT-LF" 的 PKE 方案二者的 LR-CCA 安全性的证明思路几乎是一样的. 两种构造模式中, 第二个密码组件 ABO-RLF 和 OT-LF 都起到对良生成密文有效性的一个验证作用, 同时利用它们有损模式的性质, 保证了挑战密文不会泄漏密钥太多的信息或者仅泄漏固定量的信息. 而在敌手查询的密文中, 它们都工作在标准模式下, 几乎保留了原像 (即元素 x 的哈希值 π) 的所有信息熵. 由于非良生成密文中哈希证明 π 具有较高的信息熵, 所以敌手对非良生成密文进行解密查询时, 仅有可忽略的概率能够通过密文有效性的检验. 也就是说, 解密谕言机对于敌手来说几乎是没有价值的. 因此, 在定理 5.8的证明过程中, 游戏 Game_0 到 Game_4 主要目的是将挑战密文 (x^*,s^*,t^*) 变成非良生成密文, 即元素 $x^* \in L$ 替换元素 $x^* \in X \setminus L$. 这需要用到 HPS 公开计算和私有计算的一致性以及 SMP 问题的困难性. 当 $x^* \in L$ 时, 挑战密文中的哈希函数 $\pi^* = \mathsf{H}_{sk}(x^*)$ 是一个关于私钥 sk 的有损函数, 除了公钥 pk 可能泄漏 sk 的部分信息外, π^* 不会泄漏 sk 的额外信息. 但是, 当 $x^* \in X \setminus L$ 时, $\pi^* = \mathsf{H}_{sk}(x^*)$ 可以看作是关于私钥 sk 的一个单射函数. 从信息论角度看, 此时 π^* 会泄漏私钥的全部信息. 为了避免出现这种情况, 在游戏 Game_2 中, 利用 ABO-RLF 隐藏有损分支的性质, 使得挑战密文的 ABO-RLF 工作在有损模式下, 从而保证认证元素 t^* 仅泄漏少量 π^* 的信息, 继而保证了泄漏 sk 的信息量也较少. 至此, 敌手从挑战密文中能够获取私钥 sk 的信息量是有限的, 也就是说 HPS 的私钥依然具有较高的最小熵. 在此条件下, 游戏 Game_5 主要目的是为了说明解密查询对于敌手来说是没有帮助的. 从某种意义上来说, 对于一个良生成密文 (x,s,t), 其中 $x \in L$, 敌手可以利用 x 的凭证 w 及 HPS 的公开计算模式进行解密. 而对于非良生成密文, 其中 $x \in X \setminus L$, 由于敌手能够伪造出正确的认证码 t 的概率是可忽略的, 因此, 当解密查询中 $x \notin L$ 时, 挑战者可以直接返回 "$\perp$". 由于挑战密文中 π^* 依然保持了较高的最小熵, 所以利用强随机性提取器的性质, 在 Game_6 中可以随机选择 k_0^*, 从而使得挑战密文中 k_0^* 和 k_1^* 的分布是完全一样的, 敌手无任何优势进行区分.

笔记 既然 "HPS+ABO-RLF" 和 "HPS+OT-LF" 两种构造模式的证明思路几乎是一样, 那么为什么 "HPS+ABO-RLF" 只用于构造 KEM 方案而不是 PKE 方案呢? 这是因为 ABO-RLF 中的有损分支只能事前确定. 在 KEM 方案中, 挑战者可以事先选择 (x^*,s^*) 作为有损分支. 若是 PKE 方案, 由于分支标签中还要包含敌手在挑战阶段选择的挑战消息, 因此挑战者无法事先确定有损分支. 这就需要用到类似变色龙哈希函数的性质, 可以在分支的部分信息确定的情况下再计算出有损分支.

证明 在证明中, 将满足 $x \in L$ 的密文 (x,s,t) 称之为良生成密文, 而满足 $t = f_{ek,x||s}(\pi)$ 的密文称之为有效密文. 显而, 一个有效密文不一定是良生成的.

下面通过游戏的方式组织证明. 起始游戏定义为 Game_0, 在该游戏中挑战者 $\mathcal{CH}$ 执行标准的 LR-CCA 安全性游戏, 即 k_0^* 是一个真实密钥而 k_1^* 是一个随机密钥, 而在终止游戏中, k_0^* 和 k_1^* 都是随机选取的. 令 S_i 表示事件 "$\mathcal{A}$ 在游戏 Game_i 中成功".

Game_0: 标准的 LR-CCA 安全性游戏. $\mathcal{CH}$ 按以下方式与 $\mathcal{A}$ 交互通信:

1. 初始化: $\mathcal{CH}$ 运行 $pp \leftarrow \text{HPS.Setup}(1^\kappa)$ 和 $pp' \leftarrow \text{ABORLF.Setup}(1^\kappa)$ 分别生成 HPS 和 ABORLF 的公开参数; 运行 $(pk,sk) \leftarrow \text{HPS.KeyGen}(pp)$ 和 $ek \leftarrow \text{ABORLF.Gen}(pp',0^{m+d})$ 分别生成 HPS 的公私钥 (pk,sk) 和 ABO-RLF 的求值密钥 ek; 选择一个平均强随机性提取器 $\mathsf{ext}: \Pi \times \{0,1\}^d \to \{0,1\}^\kappa$. 令 $\hat{sk} = sk$ 为密钥封装方案的私钥并将公开参数 $\hat{pp} = (pp,pp',\mathsf{ext})$ 和公钥 $\hat{pk} = (pk,ek)$ 发送给 $\mathcal{A}$.
2. 阶段 1 询问: $\mathcal{A}$ 可以自适应地进行密钥泄漏查询. 对于任意泄漏询问 $\langle g \rangle$, 只要泄漏总量小于 ℓ, 则 $\mathcal{CH}$ 返回 $g(sk)$.
3. 挑战: $\mathcal{CH}$ 按以下方式处理:
   (a). 随机选取 $\beta \xleftarrow{R} \{0,1\}$, $s^* \xleftarrow{R} \{0,1\}^d$, $(x^*,w^*) \leftarrow \text{HPS.SampR}(pp)$;
   (b). 通过 $\text{HPS.PubEval}(pk,x^*,w^*)$ 计算哈希值 $\pi^* \leftarrow \mathsf{H}_{sk}(x^*)$;

(c). 计算函数值 $t^* \leftarrow f_{ek,x^*||s^*}(\pi^*)$ 和封装密钥 $k_0^* \leftarrow \mathsf{ext}(\pi^*, s^*)$;

(d). 随机选取 $k_1^* \xleftarrow{R} \{0,1\}^\kappa$;

(e). 将挑战密文 $c^* = (x^*, s^*, t^*)$ 和 k_β^* 发送给敌手 $\mathcal{A}$.

4. 阶段 2 询问: $\mathcal{A}$ 可以自适应地进行解封装查询. 对于任意解封装询问 $c = (x, s, t)$, 当 $c \neq c^*$ 时, $\mathcal{CH}$ 返回 $\mathsf{KEM.Decaps}(\hat{sk}, c)$ 给 $\mathcal{A}$. 具体地, 利用 $\mathsf{HPS.PrivEval}(sk, x)$ 计算 $\pi \leftarrow \mathsf{H}_{sk}(x)$. 如果 $t = f_{ek,x||s}(\pi)$, 输出 $k \leftarrow \mathsf{ext}(\pi, s)$. 否则, 输出 $\perp$. 如果询问挑战密文 c^* 的解封装, 依据游戏规则, 挑战者直接拒绝回答.
5. 猜测: 最终, $\mathcal{A}$ 输出 β 的猜测结果 β'. 如果 $\beta' = \beta$, 则 $\mathcal{A}$ 成功.

根据定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game_1: 该游戏与 Game_0 的唯一不同之处是 $\mathcal{CH}$在初始化阶段选择 (x^*, w^*) 和 s^*. 该变化仅是概念上的不同. 因此:

$$\Pr[S_0] = \Pr[S_1]$$

Game_2: 该游戏与 Game_1 的不同之处是 $\mathcal{CH}$ 在生成 ABO-RLF 的求值密钥 $ek \leftarrow \mathsf{ABORLF.KeyGen}(pp_2, \cdot)$ 时, 将分支参数0^{m+d} 替换为 $x^*||s^*$. 根据 ABO-RLF 隐藏有损分支的性质, 则有:

$$|\Pr[S_1] - \Pr[S_2]| \leq \mathsf{Adv}_{\mathcal{B}_1}(\kappa)$$

其中 $\mathcal{B}_1$ 是攻击 ABO-RLF 隐藏有损分支性质的敌手.

Game_3: 该游戏与 Game_2 的不同之处是在挑战阶段$\mathcal{CH}$ 通过私有计算 $\mathsf{HPS.PrivEval}(sk, x^*)$ 替代公开计算 $\mathsf{HPS.PubEval}(pk, x^*, w$ 算哈希值 $\pi^* \leftarrow \mathsf{H}_{sk}(x^*)$. 依据 HPS 的正确性, 可得:

$$\Pr[S_2] = \Pr[S_3]$$

Game_4: 该游戏与 Game_3 的不同之处是 $\mathcal{CH}$ 通过SampNo 替代 SampRel 来采样 x^*. 该变化可以直接归约到 SMP 问题的困难性上, 即:

$$|\Pr[S_3] - \Pr[S_4]| \leq \mathsf{Adv}_{\mathcal{B}_2}(\kappa)$$

其中 $\mathcal{B}_2$ 是攻击子集成员判定问题的敌手.

Game_5: 该游戏与 Game_4 的不同之处是如果密文 $c = (x, s, t)$ 中 $x \notin L$, 则 $\mathcal{CH}$ 直接决绝回答解封装询问.

令 E 表示事件"在 Game_5 中 $\mathcal{A}$ 询问了一个非良生成但有效的解封装密文, 即, $f_{ek,x||s}(\pi) = t$, 其中 $\pi = \mathsf{H}_{sk}(x)$ 和 $x \notin L \wedge (x, s, t) \neq (x^*, s^*, t^*)$. 显然, 如果事件 E 不发生, 则 Game_4 和 Game_5 是等价的. 根据差异引理 2.1, 则有:

$$|\Pr[S_4] - \Pr[S_5]| \leq \Pr[E]$$

Game_6: 该游戏与 Game_5 不同之处是 $\mathcal{CH}$ 随机选择 $k_0^* \xleftarrow{R} \{0,1\}^\kappa$ 替代 $k_0^* \leftarrow \mathsf{ext}(\pi^*, s^*)$. 显而易见, $\mathcal{A}$ 在 Game_6 中的视图与 $\beta \in \{0,1\}$ 独立无关. 因此,

$$\Pr[S_6] = 1/2$$

下面只需要证明概率 $\Pr[E]$ 是可忽略的以及敌手在 Game_5 和 Game_6 之间的视图差别是可忽略的.

引理 5.8

概率 $\Pr[E]$ 关于安全参数 κ 是可忽略的. ♡

证明 令 E_i 表示事件 "$\mathcal{A}$ 的第 i 次解封装询问 $c = (x, s, t)$ 是非良生成但有效的密文". 根据 E 的定义, 则有 $E = \cup_{1 \leq i \leq Q_d} E_i$. 接下来, 分析 $\Pr[E_i]$ 的上界. 令 view 表示敌手在提交第一个解封装询问之前的视图. 显然, $\mathsf{view} = (pk, ek, \mathsf{leak}, x^*, s^*, t^*, k_\beta^*)$.

事件 E_1 的概率与哈希值 $\mathsf{H}_{sk}(x)$ 有密切关系. 下面的推导给出了该值的平均最小熵:

$$\begin{aligned}\tilde{\mathsf{H}}_\infty(\mathsf{H}_{sk}(x)|\mathsf{view},x) &= \tilde{\mathsf{H}}_\infty(\mathsf{H}_{sk}(x)|pk,x,\mathsf{leak},t^*,k^*_\beta) &(5.9)\\ &\geq \tilde{\mathsf{H}}_\infty(\mathsf{H}_{sk}(x)|pk,x)-\ell-\tau-\kappa &(5.10)\\ &= n-\ell-\tau-\kappa &(5.11)\end{aligned}$$

在上述推导过程中, 公式 (5.9)依据事实 $\mathsf{H}_{sk}(x)$ 由私钥 sk 和元素 x 完全确定, 而 sk 与 AOB-RLF 的求值密钥 ek, 挑战元素 x^* 和随机种子 s^* 独立无关. 公式 (5.10)依据引理 2.2和泄漏量上界为 ℓ 比特, t^* (或 k^*_β) 最多有 2^τ(或 2^κ) 个可能取值. 公式 (5.11)依据 HPS 的 ϵ_1-universal$_1$ 性质. 一个有效的密文需满足 $x||s\neq x^*||s^*$. 由于密文的第三部分元素完全确定了前两部分的元素值, 而分支满足 $x||s\neq x^*||s^*$ 的求值密钥 ek 确定了一个 ν-规则有损函数. 因此, 认证码 $t=f_{ek,x||s}(\mathsf{H}_{sk}(x))$ 依然保持了 $\mathsf{H}_{sk}(x)$ 的大部分平均最小熵. 结合引理 5.3和公式 (5.11), 故有 $\tilde{\mathsf{H}}_\infty(t|\mathsf{view}',x)\geq n-\ell-\tau-\kappa-\log\nu$. 这就证明了 $\Pr[E_1]\leq 2^{\ell+\tau+\kappa+\log\nu}/2^n$. 由于敌手每次通过非良生成密文的解封装查询最多排除 ν 个哈希值 $\mathsf{H}_{sk}(x)$, 所以 $\Pr[E_i]\leq 2^{\ell+\tau+\kappa+\log\nu}/(2^n-i\nu)$. 利用联合界性质, 则有:

$$\Pr[E]\leq \Sigma_{i=1}^{Q(\kappa)}\Pr[E_i]\leq \frac{Q(\kappa)2^{\ell+\tau+k+\log\nu}}{2^n-Q(\kappa)\nu}\leq \frac{Q_d}{2^{n-\ell-\tau-\kappa-\log\nu}-Q(\kappa)}$$

由于 $n-\tau-\ell-\kappa-\log\nu\geq\omega(\log\kappa)$, 所以该上界关于安全参数 κ 是可忽略的.

综上, 引理 5.8得证!

引理 5.9

敌手在 Game$_5$ 和 Game$_6$ 之间的视图在统计上不可区分. ♡

证明 从敌手视角看, 上述两个游戏不可区分的主要原因在于挑战密文中哈希证明系统封装的哈希值 $\mathsf{H}_{sk}(x^*)$ 依然具有较高的条件熵, 从而提取器提取的随机比特串的分布同均匀分布统计距离可忽略. 下面将 Game$_5$ 中的元素 k^*_β 记作 $k^*_{5,\beta}$, 而将 Game$_6$ 中的记作 $k^*_{6,\beta}$, 将密钥泄漏信息记作 leak. 令 $\mathsf{view}'=(pk,ek,\mathsf{leak},x^*,s^*,t^*)$. 则有:

$$\begin{aligned}\tilde{\mathsf{H}}_\infty(\mathsf{H}_{sk}(x^*)|\mathsf{view}') &= \tilde{\mathsf{H}}_\infty(\mathsf{H}_{sk}(x^*)|pk,x^*,\mathsf{leak},t^*) &(5.12)\\ &\geq \tilde{\mathsf{H}}_\infty(\mathsf{H}_{sk}(x^*)|pk,x^*)-\ell-\tau &(5.13)\\ &= n-\ell-\tau &(5.14)\end{aligned}$$

在上述推导过程中, 公式 (5.12)依据事实 $\mathsf{H}_{sk}(x^*)$ 与 ek 和 s^* 独立. 公式 (5.13)依据事实泄漏量上界为 ℓ 和 t^* 最多有 2^τ 个可能取值. 公式 (5.14)依据 HPS 的 ϵ_1-universal$_1$ 性质.

由于 $k^*_{5,0}\leftarrow\mathsf{ext}(\mathsf{H}_{sk}(x^*),s^*)$, $k^*_{6,0}\xleftarrow{\mathrm{R}}K$, ext 是一个平均意义 $(n-\tau-\ell,\kappa,\epsilon_2)$-强随机性提取器, 则有

$$\Delta[(\mathsf{view}',k^*_{5,0}),(\mathsf{view}',k^*_{6,0})]\leq\epsilon_2$$

根据 $k^*_{5,\beta}$ 和 $k^*_{6,\beta}$ 的定义, 有

$$\Delta[(\mathsf{view}',k^*_{5,\beta}),(\mathsf{view}',k^*_{6,\beta})]\leq\epsilon_2/2$$

值得注意到是, 在 Game$_5$ 和 Game$_6$ 中, 对于非良生成密文 (即 $x\notin L$) 的解封装询问, 挑战者直接返回 $\perp$, 而对于所有良生成密文 ($x\in L$) 的解封装询问, 依据 H 的仿射性质, 其结果完全由 (pk,ek) 确定. 由此可知, Game$_5$ 中的所有解封装询问结果完全由 $(\mathsf{view}',k^*_{5,\beta})$ 的一个函数 h 确定, 而 Game$_6$ 中的所有解封装询问完全由同一函数的函数值 $h(\mathsf{view}',k^*_{6,\beta})$ 确定. 敌手在 Game$_5$ 中的视图记作 $\mathsf{view}_5=(\mathsf{view}',k^*_{5,\beta},h(\mathsf{view}',k^*_{5,\beta}))$, 在 Game$_6$ 中的视图记作 $view_6=(\mathsf{view}',k^*_{6,\beta},h(\mathsf{view}',k^*_{6,\beta}))$. 则有 $\Delta[\mathsf{view}_5,\mathsf{view}_6]\leq\epsilon_2/2$. 引理 5.9得证!

综上, 定理 5.8得证! □

5.2 抗篡改安全

千磨万击还坚劲, 任尔东西南北风.

— 清 · 郑燮《竹石》

侧信道攻击不仅可能获取密码算法在实现过程中的部分内部状态信息, 还可能通过错误注入等方式改变密码算法实现的内部状态. 当敌手篡改密码算法的密钥并观察在篡改密钥下的密码算法输出结果时, 这类侧信道攻击即是密钥篡改攻击. 被篡改的密钥可以是签名方案的签名密钥也可以是加密方案的解密密钥. 密钥篡改攻击也称相关密钥攻击 (related-key attack, RKA), 最早由 Biham [320] 和 Knudsen [321] 提出的. 2003 年, Bellare 和 Kohno [322] 给出它的形式化定义. 设计抵抗密钥篡改攻击的密码算法的目标之一是能够抵抗范围更广的密钥篡改函数. 早期设计的密码算法仅能抵抗简单的线性密钥篡改攻击, 例如 Bellare 和 Cash [323] 提出的基于 DDH 假设的抗相关密钥攻击的伪随机函数 (RKA-PRFs). 2011 年, Bellare 等 [89] 给出如何从 RKA-PRFs 和其它非 RKA 安全的密码算法来实现 RKA 安全的密码算法, 包括公钥加密、对称加密、签名和加密. 同年, Applebaum 等 [324] 提出基于 LPN 和 LWE 假设的抗线性密钥篡改语义安全对称加密方案. 2012 年, Wee [318] 提出利用特殊性质的自适应单向陷门函数构造抗线性篡改的 RKA-CCA 安全公钥加密方案, 并给出基于因子分解、DBDH 和 LWE 等困难问题的具体实现.

5.2.1 抗篡改安全模型

一个密码系统通常由系统参数、算法 (程序实现的代码) 和密钥 (公钥/私钥) 三部分组成. 公钥/私钥是最有可能受到 RKA 攻击的, 而系统参数和算法假定是不受攻击的. 这是因为, 系统参数并不包含用户的密钥信息, 与用户是独立的. 它可以在用户密钥选取之前确定并且可以嵌入到算法的实现代码中. 令 $\Phi = \{\phi : SK \to SK\}$ 是一个从密钥空间 SK 到自身的变换函数族. 一个公钥加密方案 PKE 的 RKA-CCA 安全模型的定义如下:

RKA-CCA 安全性. 定义公钥加密方案 PKE 的 RKA-CCA 敌手 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left| \Pr \left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ (m_0, m_1, state) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{rka}}}(pp, pk), s.t. |m_0| = |m_1|; \\ \beta \xleftarrow{R} \{0,1\}; \\ c^* \leftarrow \mathsf{Encrypt}(pk, m_\beta); \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{rka}}}(pp, pk, state, c^*); \end{array} \right] - \frac{1}{2} \right|$$

上述定义中, 敌手 $\mathcal{A}$ 在接收到挑战密文前后两阶段都可以访问密钥篡改谕言机同时获得在篡改密钥下的解密结果. 具体地, 谕言机 $\mathcal{O}_{\mathsf{rka}}$ 的输入为一对篡改函数和密文 (ϕ, c), 其中 $\phi \in \Phi$, 输出为 $\mathsf{Decrypt}(\phi(sk), c)$. 在第 2 阶段询问中, 敌手不能进行满足条件 $(\phi(sk), c) = (sk, c^*)$ 的询问, 否则敌手直接获取了挑战密文的解密结果, 使安全模型失去了实际意义. 如果对于任意的 PPT 敌手 $\mathcal{A}$, 优势函数 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 是可忽略的, 则称公钥加密方案 PKE 是 Φ-RKA-CCA 安全的.

笔记 在 RKA 攻击中, Φ 称为密钥篡改函数族. 如果对于所有密钥 $sk \in \mathcal{SK}$ 及所有不同的篡改函数 $\phi, \phi' \in \Phi$ 都有 $\phi(sk) \neq \phi'(sk)$, 则密钥篡改函数族 Φ 称为 "claw-free" 的. 在已有的 RKA 安全加密或其他密码方案中, 大部分方案仅能抵抗这类篡改函数. "claw-free" 篡改函数是一种特殊的函数, 在实际中, 绝大部分篡改攻击函数都是非 "claw-free" 的. 从前面的定义可以看出 RKA-CCA 与 IND-CCA 安全性之间有着密切的联系. 在两种模型中, 敌手都可以访问解密服务. 不同之处在于 RKA 敌手还可以访问篡改密钥下的解密服务. 此外, 只要 $\phi(sk) \neq sk$, 敌手是可以访问挑战密文的解密服务的. 这也是 RKA-CCA 安全性比 IND-CCA 安全性更难实现的原因之一. 如果密钥篡改函数仅包含恒等函数 1_ϕ, 则 $\{1_\phi\}$-RKA-CCA 等价于 IND-CCA.

5.2.2 RKA-CCA 安全 PKE 的通用构造方法

5.2.2.1 基于自适应单向陷门关系的 RKA-CCA 安全 PKE

自适应单向陷门关系 (ATDR) 在构造 IND-CCA 安全公钥加密方面具有强大的优势. 而 RKA-CCA 安全的 PKE 本身也是 IND-CCA 安全的, 那么一个自然的问题是自适应单向陷门关系能否用于构造 RKA-CCA 安全的公钥加密方案, 需要满足哪些特殊的性质, 篡改函数集的形式又如何呢? 2012 年, Wee [318] 给出了这些问题的答案, 提出一种基于自适应单向陷门关系的 RKA-CCA 安全公钥加密方案的通用构造. 带标签自适应单向陷门关系在随机采样和求逆算法中会额外输入一个标签, 而该标签与自适应单向陷门关系的陷门无关. 一个带标签自适应单向陷门关系 ATDR $=(\mathsf{Setup},\mathsf{KeyGen},\mathsf{Sample},\mathsf{TdInv})$ 需要满足以下两个额外的性质:

- 密钥同态性 (Φ-key homomorphism): 对于任意 $\phi\in\Phi$ 和任意的陷门 td, 标签 tag, 关系值 y, 存在一个 PPT 算法 T, 使得:
$$\mathsf{TdInv}(\phi(td),tag,y)=\mathsf{TdInv}(td,tag,T(pp,\phi,tag,y)).$$
- 指纹识别性 (Φ-fingerprinting): 类似于指纹认证, 对于一个固定的关系值 (指纹) y^*, 对陷门的任何篡改, 通过求逆算法都可以被检测出来. 具体地, 定义敌手 $\mathcal{A}$ 的优势函数如下:
$$\mathsf{Adv}_{\mathcal{A}}(\kappa)=\Pr\left[\begin{array}{c|l}\mathsf{TdInv}(\phi(td),tag^*,y^*)\neq\perp & pp\leftarrow\mathsf{Setup}(1^\kappa);\\ \wedge\phi\in\Phi\wedge\phi(td)\neq td & (ek,td)\leftarrow\mathsf{KeyGen}(pp);\\ & (s^*,y^*)\xleftarrow{R}\mathsf{Sample}(ek,tag^*);\\ & \phi\leftarrow\mathcal{A}(pp,ek,td,y^*);\end{array}\right]$$

对于任意 PPT 敌手 $\mathcal{A}$, 如果优势函数 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 都是可忽略的, 则称 ATDR 是 Φ-fingerprinting 的.

笔记 上面这两个额外的性质为 RKA-CCA 安全性的证明提供了一种简洁的解决办法. 首先, 密钥同态性实际上提供了一种通过原始求逆谕言机 $\mathsf{TdInv}(td,tag,\cdot)$ 来回答在篡改密钥 $\phi(td)$ 下的求逆询问. 指纹识别性实际上保证了敌手不能查询挑战关系值在原始陷门下的求逆询问. 当用 ATDR 构造 IND-CCA 安全的公钥加密方案时, 这两种额外的性质可以直接用于 RKA-CCA 安全性中, 从而使得构造 IND-CCA 安全的公钥加密方案也是 RKA-CCA 安全的.

在 Φ-fingerprinting 性质中, 敌手知晓陷门 td. 这一事实在后面的证明中至关重要, 等价于挑战者知道自适应单向陷门关系的陷门, 从而可以正确应答解密询问.

下面介绍如何基于 ATDR 构造一个 RKA-CCA 安全的公钥加密方案. 该构造还需要一个强不可伪造一次签名方案 OTS $=(\mathsf{Setup},\mathsf{KeyGen},\mathsf{Sign},\mathsf{Verify})$. 其中, $\mathsf{Setup}(1^\kappa)$ 输出签名方案的公开参数 pp; $\mathsf{KeyGen}(pp)$ 输出签名公钥 vk 和私钥 sk; $\mathsf{Sign}(sk,m)$ 输出消息 m 的签名 σ; $\mathsf{Verify}(vk,m,\sigma)$ 输出 1 当且仅当 σ 是 m 的一个合法签名. 一次签名方案的敌手 $\mathcal{A}$ 的优势函数定义如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa)=\Pr\left[\begin{array}{c|l}\mathsf{Verify}(vk,m',\sigma')=1 & pp\leftarrow\mathsf{Setup}(1^\kappa);\\ \wedge(m',\sigma')\neq(m,\sigma) & (vk,sk)\leftarrow\mathsf{KeyGen}(pp);\\ & m\leftarrow\mathcal{A}(vk);\\ & \sigma\xleftarrow{R}\mathsf{Sign}(sk,m);\\ & (m',\sigma')\leftarrow\mathcal{A}(\sigma);\end{array}\right]$$

对于任意 PPT 敌手, 如果优势函数 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 是可忽略的, 则称一次签名方案是强不可伪造的.

构造 5.12 (基于 ATDR 的 RKA-CCA 安全 PKE)

构造所需的组件是:

- 自适应单向陷门关系 ATDR $=(\mathsf{Setup},\mathsf{KeyGen},\mathsf{Sample},\mathsf{TdInv})$.
- 强不可伪造一次签名方案 OTS $=(\mathsf{Setup},\mathsf{KeyGen},\mathsf{Sign},\mathsf{Verify})$.
- 伪随机函数 $\mathsf{G}:X\to\{0,1\}^l$.

构造 PKE 如下:

- Setup(1^κ): 运行 ATDR.Setup(1^κ) 和 OTS.Setup(1^κ), 分别输出 ATDR 的系统参数 pp_1 和 pp_2. 选择一个伪随机函数 $\mathsf{G}: X \to \{0,1\}^l$, 其中 X 为 ATDR 的原像空间. 输出公钥加密方案的公开参数 $pp = (pp_1, pp_2, \mathsf{G})$. 其中 $\{0,1\}^l$ 作为明文空间.
- KeyGen(pp): 运行 $(ek, td) \leftarrow$ ATDR.KeyGen(pp), 输出公钥 $pk := ek$ 和私钥 $dk := td$.
- Encrypt(pk, m): 以公钥 $pk := ek$ 和明文 $M \in \{0,1\}^l$ 为输入, 执行如下步骤:
  1. 运行 $(vk, sk) \leftarrow$ OTS.KeyGen(pp_2) 生成一次签名的公钥和私钥;
  2. 运行 $(x, y) \leftarrow$ ATDR.Sample(ek, vk) 生成一个随机采样 (x, y);
  3. 计算 $\psi = \mathsf{G}(x) \oplus m$;
  4. 运行 $\sigma \leftarrow$ OTS.Sign($sk, y||\psi$);
  5. 输出密文 $c = (vk, \sigma, y, \psi)$.
- Decrypt(dk, c): 以私钥 $dk := td$ 和密文 $c = (vk, \sigma, y, \psi)$ 为输入, 执行如下步骤:
  1. 验证 OTS.Verify($vk, y||\psi, \sigma$) $= 1$. 若不成立, 则返回 $\perp$, 否则执行后续步骤;
  2. 计算 $x \leftarrow$ ATDR.TdInv(td, vk, y). 若 $x = \perp$, 则返回 $\perp$, 否则执行后续步骤;
  3. 计算 $m' = \mathsf{G}(x) \oplus \psi$ 并返回明文 m'.

♣

正确性. 构造 5.12的正确性可由自适应单向陷门关系的正确性直接推导出来. 下面主要介绍方案的 RKA-CCA 安全性的证明.

定理 5.9

如果 ATDR 是一族自适应单向陷门关系, 且满足 Φ-密钥同态性和 Φ-指纹识别性, OTS 是一个强不可伪造一次签名方案, 那么构造 5.12中的 PKE 是 Φ-RKA-CCA 安全的.

♡

证明 令 S_i 表示敌手在 Game$_i$ 中的成功事件. 以游戏序列的方式组织证明如下:

Game$_0$: 该游戏是标准的 RKA-CCA 游戏, 挑战者 $\mathcal{CH}$ 和敌手 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 运行 Setup(1^κ) 生成公开参数 pp, 同时运行 KeyGen(pp) 生成公私钥对 (pk, sk). $\mathcal{CH}$ 将 (pp, pk) 发送给 $\mathcal{A}$.
- 询问: 对于敌手的任意询问 (ϕ, c), $\mathcal{CH}$ 首先判断 $\phi \in \Phi$ 是否成立. 如果成立, 则返回 Decrypt($\phi(sk), c$) 的解密结果; 否则, 返回 $\perp$.
- 挑战: $\mathcal{A}$ 选择 $m_0, m_1 \in \mathbb{G}$ 并发送给 $\mathcal{CH}$. $\mathcal{CH}$ 选择随机比特 $\beta \in \{0,1\}$, 作如下计算:
  1. 运行 $(vk^*, sk^*) \leftarrow$ OTS.KeyGen(pp_2) 生成一次签名的公钥和私钥;
  2. 运行 $(x^*, y^*) \leftarrow$ ATDR.Sample(ek, vk^*) 生成一个随机采样 (x^*, y^*);
  3. 计算 $\psi^* = \mathsf{G}(x^*) \oplus m_\beta$;
  4. 运行 $\sigma^* \leftarrow$ OTS.Sign($sk^*, y^*||\psi^*$);
  5. 输出密文 $c^* = (vk^*, \sigma^*, y^*, \psi^*)$ 并发送给 $\mathcal{A}$.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. $\mathcal{A}$ 成功当且仅当 $\beta' = \beta$.

根据定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game$_1$: 该游戏与 Game$_0$ 的唯一不同在于拒绝解密查询的条件. 对于解密查询 (ϕ, c), 其中 $c = (vk, \sigma, y, \psi)$, 若$vk = vk^*$, 则 $\mathcal{CH}$ 直接拒绝提供解密服务并返回 $\perp$. 若 $vk \neq vk^*$, 则 $\mathcal{CH}$ 提供的解密谕言机与 Game$_0$ 完全一样. 下面分四种情况讨论敌手在两个连续游戏中的视图之间的区别.

- 情形 1: $vk \neq vk^*$. 在这种情况下, 游戏 Game$_0$ 与 Game$_1$ 中的解密谕言机是完全一样的.
- 情形 2: $vk = vk^*$ 且 $(\sigma, y||\psi) = (\sigma^*, y^*||\psi^*)$ 且 $\phi(dk) = dk$. 该情况实际上等价于 $(\phi(dk), c) = (sk, c^*)$. 根据 RKA-CCA 安全模型的定义, 这种情况在两个游戏中都是不允许进行解密查询的.

- 情形 3: $vk = vk^*$ 且 $(\sigma, y||\psi) \neq (\sigma^*, y^*||\psi^*)$. 根据一次签名的强不可伪造性, 可以直接证明 $(y||\psi, \sigma)$ 通过验证的概率是可忽略的. 因此, 对于任意攻击一次签名强不可伪造性的敌手 $\mathcal{B}_1$, 则有:

$$\Pr\left[\mathsf{OTS.Verify}(vk, y||\psi, \sigma) = 1\right] \leq \mathsf{Adv}_{\mathcal{B}_1}(\kappa)$$

- 情形 4: $vk = vk^*$ 且 $(\sigma, y||\psi) = (\sigma^*, y^*||\psi^*)$ 且 $\phi(sk) \neq sk$. 根据 ATDR 的 Φ-指纹识别性, 解密服务在计算 $x \leftarrow \mathsf{ATDR.TdInv}(td, vk, y)$ 时, $x \neq \perp$ 的概率是可忽略的. 若 $x = \perp$, 则解密谕言机直接返回 $\perp$. 此时, 两个游戏中解密谕言机返回的结果是一样的. 因此, 对于任意攻击自适应单向陷门关系 Φ-指纹识别性的敌手 $\mathcal{B}_2$, 则有:

$$\Pr\left[\mathsf{ATDR.TdInv}(\phi(dk), vk^*, y) \neq \perp\right] \leq \mathsf{Adv}_{\mathcal{B}_2}(\kappa)$$

由上述分析可知, 敌手在两个游戏中的视图区别是可忽略的, 故有:

$$|\Pr[S_0] - \Pr[S_1]| \leq \mathsf{Adv}_{\mathcal{B}_1}(\kappa) + \mathsf{Adv}_{\mathcal{B}_2}(\kappa)$$

Game_2: 该游戏与 Game_1 的唯一不同在于挑战者利用 ATDR 的自适应性来回答解密询问. 对于敌手提交的解密查询 (ϕ, c), 其中 $c = (vk, \sigma, y, \psi)$, 解密谕言机的工作方式如下:

- 若 $vk = vk^*$ 或者 $\mathsf{OTS.Verify}(vk, y||\psi, \sigma) = 0$, 返回 $\perp$;
- 计算 $x := \mathsf{ATDR.TdInv}(td, vk, T(pp, \phi, vk, y))$. 如果 $x = \perp$, 返回 $\perp$;
- 否则, 计算并返回 $m' = \mathsf{G}(x) \oplus \psi$.

根据 Φ-密钥同态性, 则有 $\mathsf{ATDR.TdInv}(td, vk, T(pp, \phi, vk, y)) = \mathsf{ATDR.TdInv}(\phi(td), vk, y)$. 也就是说, 挑战者在 Game_2 中模拟的解密谕言机和 Game_1 中的是完全一致的, 故有:

$$\Pr[S_1] = \Pr[S_2]$$

Game_3: 该游戏与 Game_2 的唯一不同在于挑战密文中 ψ^* 的计算方式. $\mathcal{CH}$随机选择 $K \xleftarrow{R} \{0,1\}^l$, 计算 $\psi^* = K \oplus m_\beta$. 根据 ATDR 的自适应单向性及函数 G 输出结果的伪随机性, 可以证明敌手在这两个连续游戏中的视图区别不超过 $\mathsf{Adv}_{\mathcal{B}_3}(\kappa)$. 因此, 对于任意攻击自适应单向陷门关系的单向性的敌手 $\mathcal{B}_3$, 则有:

$$|\Pr[S_2] - \Pr[S_3]| \leq \mathsf{Adv}_{\mathcal{B}_3}(\kappa)$$

在最后一个游戏中, 由于 ψ^* 的分布与挑战比特 β 完全独立不相关, 从而敌手在游戏中的优势为零, 即:

$$Pr[S_3] = \frac{1}{2}$$

综上, 定理 5.9得证. □

笔记 一次签名方案的强不可伪造安全性与自适应单向陷门关系的指纹识别性相结合, 完美地避免了挑战密文中的标签 vk^* 被敌手重用. 再结合密钥同态性, 又完美地将解密谕言机转化为 ATDR 的自适应求逆谕言机. 最后, 再根据 ATDR 的单向性及函数 G 的伪随机性, 将挑战密文 ψ^* 变得完全随机, 从而不会泄漏挑战比特 β 的任何信息.

5.2.2.2 基于不可延展函数的 RKA-CCA PKE 构造

上世纪 90 年代, Gödel Prize 得主 Dwork 和 Naor 引入的不可延展性是密码学中单向性和伪随机性之外的另一重要性质, 该性质精准刻画了密码组件输入/输出之间的独立性. 已有的研究工作考察了加密、承诺、零知识证明、编码、程序混淆的不可延展性, 然而一直未涉及密码学乃至计算机科学中最基本的函数. 函数的不可延展性与单向性之间存在怎样的关联以及如何构造高效的不可延展函数均是未解的公开问题.

2022 年, 陈等 [325] 在 PKC 2016 工作 [93] 的基础上进一步完善了不可延展函数的性质及构造, 成功解决了上述问题. 在理论层面, 首次绘制出函数不可延展性与单向性之间的清晰图景, 通过巧妙结合方程求解技巧和变换集代数性质, 建立起不可延展函数与单向函数之间的关联, 并分别在标准模型和随机谕言机模型中给出了通用构造, 解决了 Boldyreva 和 Kiltz 等著名密码学家提出的公开问题. 在应用层面, 不仅直接蕴含了密码谜题的高效设计, 还深度揭示了不可延展函数在抗篡改安全中的强力应用: (1) 证明了对于代数诱导的变换集, 抗非平凡拷贝攻击属于密码方案的内蕴性质, 从而直接提升了一大批密码方案的抗相关密钥攻击安全性; (2) 构造出了迄今为止效率和安全均最优的认证密钥导出函数, 提供了将传统安全提升为抗篡改安全的关键技术工具.

下面介绍不可延展函数的相关概念及性质.

定义 5.8 (有效计算函数)

一族有效计算函数 F 包含以下 3 个 PPT 算法 $(\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Verify})$:

- $\mathsf{KeyGen}(1^\kappa)$: 输入安全参数 1^κ, 输出函数索引 s. 每个函数索引 s 定义了一个函数 $f_s : X_s \to Y_s$. 该函数可以是确定性的也可以是随机性的.
- $\mathsf{Eval}(s, x)$: 输入函数索引 s 和元素 $x \in X_s$, 输出函数的像值 $y \leftarrow f_s(x)$. 令 $\mathrm{supp}(f_s(x))$ 是随机变量 $f_s(x)$ 的支撑集. 如果 f_s 是确定的, 则 $\mathrm{supp}(f_s(x))$ 缩减为包含唯一像值 $f_s(x)$ 的集合.
- $\mathsf{Verify}(s, x, y)$: 输入函数索引 s 和 $(x, y) \in X_s \times Y_s$, 当 $y \in \mathrm{supp}(f_s(x))$ 时, 输出 "1", 否则, 输出 "0". 对于确定性函数, 可以直接通过重新计算 x 的像值来验证.

♣

笔记 上述定义统一了确定函数和随机函数的概念. 对于任意两个不同的原像 $x_1, x_2 \in X_s$, 如果 $\mathrm{supp}(f(x_1))$ 和 $\mathrm{supp}(f(x_2))$ 没有交集, 则 f 是单射函数.

有效计算函数的单向性 (one-wayness) 和不可延展性 (non-malleability) 的定义分别如下:

定义 5.9 (单向性和自适应单向性)

定义 F 的单向性敌手的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[x \in f_s^{-1}(y^*) : \begin{array}{l} s \leftarrow \mathsf{KeyGen}(1^\kappa); \\ x^* \xleftarrow{R} X_s, y^* \leftarrow f_s(x^*); \\ x \leftarrow \mathcal{A}(f_s, y^*); \end{array}\right].$$

对于任意 PPT 敌手 $\mathcal{A}$, 如果优势函数 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 关于安全参数 κ 是可忽略的, 则 F 是单向的. 在允许 $\mathcal{A}$ 访问除了 y^* 之外的任意 y 的求逆谕言机 $\mathcal{O}_{\mathsf{inv}}$ 情况下, 如果 F 仍然是单向的, 则称 F 是自适应单向的.

♣

定义 5.10 (不可延展性和自适应不可延展性)

令 Φ 是一个定义域 X 上的变换函数集. 定义 F 的不可延展性敌手的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[\begin{array}{c} \phi \in \Phi \wedge \mathsf{Verify}(s, \phi(x^*), y) = 1 \\ \wedge (\phi, y) \neq (\mathsf{id}, y^*) \end{array} : \begin{array}{l} s \leftarrow \mathsf{Gen}(1^\kappa); \\ x^* \xleftarrow{R} X_s, y^* \leftarrow f_s(x^*); \\ (\phi, y) \leftarrow \mathcal{A}(f_s, y^*); \end{array}\right].$$

对于任意 PPT 敌手 $\mathcal{A}$, 如果优势函数 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 关于安全参数 κ 是可忽略的, 则 F 是 Φ-不可延展的. 在允许 $\mathcal{A}$ 访问除了 y^* 之外的任意 y 的求逆谕言机 $\mathcal{O}_{\mathsf{inv}}$ 情况下, 如果 F 仍然是不可延展的, 则称 F 是自适应 Φ-不可延展的.

♣

笔记 一般来说, 一族函数的定义域和值域范围依赖函数索引. 为简便起见, 本书假设对于所有函数索引 s, 定义域和值域都是不变的. 从而将 X_s, Y_s 和 f_s 分别简写为 X, Y 和 f. 在不可延展函数中, 存在一些不可能转换函数类, 如恒等变换 id 和常量变换 ϕ_c, 无法实现不可延展性. 敌手可以输出 (id, y^*) 或 $(\phi_c, f(c))$ 从而赢得游戏. 因此, 给出一个可能实现不可延展性的变换函数集 Π 是非常重要的.

定义 5.11 (一般变换函数集)

定义满足下面两个性质的变换函数集 $\Phi_{\mathrm{brs}}^{\mathrm{srs}}$:

- 有界根集合 (bounded root space): 令 $r(\kappa)$ 是安全参数 κ 的一个变量, $R_\phi = \{x \in X : \phi(x) = 0\}$. 如果 $|R_\phi| \leq r(\kappa)$, 则 ϕ 最多有 $r(\kappa)$ 个根. 如果对于每个 $\phi \in \Phi$ 和 $\phi_c \in \mathsf{cf}$, 变换函数 $\phi' = \phi - \phi_c$ 和 $\phi'' = \phi - \mathsf{id}$ 都最多有 $r(\kappa)$ 个根, 那么称变换函数集 Φ 有 $r(\kappa)$-有界根集.
- 可采样根集合 (sampleable root space): 如果存在一个 PPT 算法 SampRS, 输入 ϕ, 均匀随机地输出集合 R_ϕ 中的一个元素, 则称变换函数 ϕ 有一个可采样根集. 如果对于每个 $\phi \in \Phi$ 和 $\phi_c \in \mathsf{cf}$, 复合函数 $\phi' = \phi - \phi_c$ 和 $\phi'' = \phi - \mathsf{id}$ 都有可采样根集, 那么称变换函数集 Φ 有可采样根集.

♣

笔记 变换函数集 $\Phi^{\text{srs}}_{\text{brs}}$ 非常强大，几乎包含了所有除去恒等变换 id 和常量变换 cf 的代数诱导变换函数集，如线性变换集 $\Phi^{\text{lin}} = \{\phi_a : \phi_a(x) = a + x\}_{a\in\mathbb{G}}$，仿射变换集 $\Phi^{\text{aff}} = \{\phi_{a,b} : \phi_{a,b}(x) = ax + b\}_{a,b\in\mathbb{R}}$ 和多项式变换集 $\Phi^{\text{poly}(d)} = \{\phi_q : \phi_q(x) = q(x)\}_{q\in\mathbb{F}_d(x)}$，其中 $\mathbb{G}$ 是一个群，$\mathbb{R}$ 是一个环，$\mathbb{F}_d(x)$ 是有限域 $\mathbb{F}$ 上次数不超过 d 的多项式集.

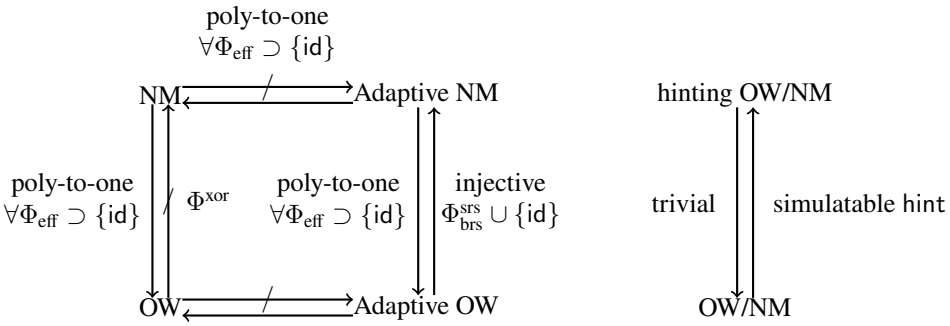


图 5.18: 不可延展函数与单向函数之间的关系

图 5.18概括地给出了函数的不可延展性与单向性之间的关系. 在一定条件下，一个函数满足不可延展性，则一定是单向的，反之却不一定成立，见引理 5.10和引理 5.11. 在一定条件下，自适应不可延展性与自适应单向性是等价的，见引理 5.12和引理 5.13. 而自适应安全性强于非自适应安全性似乎是显然的，见引理 5.14. 下面介绍每个引理的具体内容.

引理 5.10

令 F 是一族多对一函数. 对于任意可实现的变换集 Φ，则: Φ-不可延展性 $\Rightarrow$ 单向性.

证明 假设 $\mathcal{A}$ 是一个以不可忽略概率攻破 F 单向性的敌手，则可以构造另一个算法 $\mathcal{B}$ 以不可忽略的概率攻破 F 的不可延展性. 算法 $\mathcal{B}$ 模拟敌手 $\mathcal{A}$ 在单向性游戏中的挑战者的过程如下:

- 初始化及挑战: 给定一个函数 $f \leftarrow F.\mathsf{KeyGen}(1^\kappa)$ 和一个像值 $y^* \leftarrow f(x^*)$，其中 $x^* \xleftarrow{R} X$，$\mathcal{B}$ 将 (f, y^*) 发送给敌手 $\mathcal{A}$.
- 攻击: 当敌手 $\mathcal{A}$ 输出一个解 x 时，$\mathcal{B}$ 随机选择一个变换函数 $\phi \in \Phi\backslash\mathsf{id}$，返回 $(\phi, f(\phi(x)))$ 作为对 $\mathcal{F}$ 不可延展性的攻击结果.

由于 F 是多对一的，在 $\mathcal{A}$ 攻击成功的条件下，即 $x \in f^{-1}(y^*)$，则有 $\Pr[x = x^*|y^*] \geq 1/\mathsf{poly}(\kappa)$，其中 $x^* \xleftarrow{R} X$. 这是因为，最多有多项式个数的 x 满足 $f(x) = y^*$，并且每个 x 在敌手 $\mathcal{A}$ 的角度看都是一样的. 因此，如果 $\mathcal{A}$ 能够以不可忽略的概率攻破 $\mathcal{F}$ 的单向性，那么算法 $\mathcal{B}$ 同样能够以不可忽略的概率攻破 $\mathcal{F}$ 的不可延展性. □

笔记 引理 5.10的结论是非常直接的. 如果一个函数都不满足单向性，那么，在知道它的原像的情况下，可以直接计算出与该原像相关的任意原像的像，从而攻破函数的不可延展性.

引理 5.10反向结论并不成立，也就是说一个函数满足单向性但并不一定满足不可延展性.

引理 5.11

单向性 $\not\Rightarrow$ Φ^{xor}-不可延展性.

证明

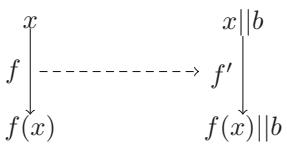


图 5.19: 可延展函数示例

引理 5.11可通过反证法证明: 如图 5.19所示, 通过一个单向函数 f 构造另一个单向函数 f', 使得 f' 仍然满足单向性, 但是不满足异或变换集下的不可延展性. 引理得证! □

笔记 不可延展性的反例还有 Φ-同态的单向函数. 这是因为, 对于任意 $\phi \in \Phi$ 和任意 $x \in X$, 有 $f(\phi(x)) = \phi(f(x))$. 显而易见, f 是单向的, 但不是 Φ-不可延展的.

引理 5.12

对于 F 上任意可实现的变换集 Φ, 自适应 Φ-不可延展性 $\Rightarrow$ 自适应单向性.

证明 引理 5.12的证明可以直接通过引理 5.10的结论推出. □

引理 5.13

当 F 是单射函数时, 对于 $\Phi = \Phi_{\text{brs}}^{\text{srs}} \cup \mathsf{id}$, $(q+1)$-自适应单向性 $\Rightarrow$ q-自适应 Φ-不可延展性.

证明 假设 $\mathcal{A}$ 是一个攻击 F 自适应不可延展性的敌手. 下面构造一个攻击 F 自适应单向性的敌手 $\mathcal{B}$. $\mathcal{B}$ 按以下方式模拟 $\mathcal{A}$ 在自适应不可延展游戏中的挑战者:

- 初始化及挑战: 给定 $f \leftarrow F.\mathsf{KeyGen}(1^\kappa)$ 和一个像值 $y^* \leftarrow f(x^*)$, 其中 $x^* \xleftarrow{R} X$, $\mathcal{B}$ 将 (f, y^*) 作为挑战信息发送给敌手 $\mathcal{A}$.
- 攻击: 当 $\mathcal{A}$ 询问求逆谕言机时, $\mathcal{B}$ 将询问直接发送给自己的挑战者并将返回的结果发送给 $\mathcal{A}$. 当 $\mathcal{A}$ 输出一个攻击结果 $(\phi, y) \neq (\mathsf{id}, y^*)$ 时, $\mathcal{B}$ 按下面的方式进行处理:
  - 情形 1: $\phi = \mathsf{id} \wedge y \neq y^*$. $\mathcal{B}$ 询问求逆谕言机 $\mathcal{O}_{\mathsf{inv}}$, 获取 y 的逆 x, 再将 x 输出作为 $\mathcal{B}$ 的求解结果.
  - 情形 2: $\phi \in \Phi_{\text{brs}}^{\text{srs}} \wedge y \neq y^*$. $\mathcal{B}$ 询问求逆谕言机 $\mathcal{O}_{\mathsf{inv}}$, 获得 y 的逆 x, 再运行 $\mathsf{SampRS}(\phi')$ 输出 $\phi'(\alpha) = 0$ 的一个随机解, 其中 $\phi'(\alpha) = \phi(\alpha) - x$.
  - 情形 3: $\phi \in \Phi_{\text{brs}}^{\text{srs}} \wedge y = y^*$. $\mathcal{B}$ 运行 $\mathsf{SampRS}(\phi'')$ 输出 $\phi''(\alpha) = 0$ 的一个随机解, 其中 $\phi''(\alpha) = \phi(\alpha) - \alpha$.

下面分析 $\mathcal{B}$ 的策略的正确性. 在 $\mathcal{A}$ 攻击成功的条件下, 则有 $\mathsf{Verify}(f, \phi(x^*), y) = 1$. 利用 $\mathcal{F}$ 的单射性质, 对于情形 1, 有 $\mathsf{id}(x^*) = x^* = x$; 对于情形 2, 则有 $\phi(x^*) = x$, 即 x^* 是 $\phi'(\alpha) = 0$ 的一个解; 对于情形 3, 则有 $\phi(x^*) = x^*$, 即 x^* 是 $\phi''(\alpha) = 0$ 的一个解. 将这三种情形结合起来, 利用变换集 $\Phi_{\text{brs}}^{\text{srs}}$ 的 BRS&SRS 性质, $\mathcal{B}$ 最多通过 $(q+1)$ 次求逆询问输出正确解 x^* 的概率为 $1/\mathsf{poly}(\kappa)$. 因此, 如果 $\mathcal{A}$ 攻破 q-自适应不可延展性的概率是不可忽略的, 那么 $\mathcal{B}$ 攻破 $(q+1)$-自适应单向性的概率也是不可忽略的. 引理 5.13得证. □

引理 5.14

如果 F 是一族多对一函数, 对于任意可实现的变换集 $\Phi \supset \{\mathsf{id}\}$, Φ-不可延展性 $\not\Rightarrow$ 自适应 Φ-不可延展性.

证明 引理 5.14可通过反例来证明. 令 $F = (\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Verify}, \mathsf{TdInv})$ 是一族带陷门的 Φ-不可延展函数. 如图 5.20所示, 则可以从 F 构造另一族函数 F', 使得 F' 仍然是 Φ-不可延展的, 但不是自适应 Φ-不可延展的.

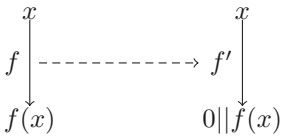


图 5.20: 自适应可延展函数示例

对于任意 $f \in F$, 假设 $f : \{0,1\}^n \to \{0,1\}^m$, 则 $f' \in F'$ 的定义如下:

$$\begin{array}{rccc} f': & \{0,1\}^n & \to & \{0,1\}^{m+1} \\ & x & \to & 0||f(x) \end{array}$$

对于任意 $y' = b||y \in \{0,1\}^{m+1}$, $\mathcal{F}'$ 求逆函数的定义如下:

$$f'^{-1}(y') = \begin{cases} f^{-1}(y) & \text{if } b = 0 \\ td & \text{if } b = 1 \end{cases}$$

上面实际上定义了一个“危险的”求逆算法，对于非法原像 $b||y$ (其中 $b=1$)，求逆算法将直接输出 f 的陷门. 因此，在自适应不可延展游戏中，敌手可以通过这类“危险的”询问获取求逆陷门从而攻破 f' 的不可延展性. 而在不可延展性游戏里，由于没有提供求逆谕言机，f' 依然保持有不可延展性. 综上，引理 5.14得证. □

不可延展函数的构造

下面介绍如何利用自适应单向陷门函数和全除一有损陷门函数分别构造确定性不可延展函数和随机化不可延展函数. 对于确定性不可延展函数，可以通过自适应单向陷门函数直接构造. 在下面的结论中，$\mathcal{H}$-hinting 的含义是函数除了输出 $y=f(x)$，还会输出 x 的 hardcore，即 $h(x)$，其中 $h \xleftarrow{R} \mathcal{H}$. 对于任意变换集 $\Phi \subseteq \Phi_{\text{brs}}^{\text{srs}} \cup \mathsf{id}$，自适应 Φ-不可延展性蕴含了 $\mathcal{H}$-hinting 自适应 Φ-不可延展性，见引理 5.15.

引理 5.15 (计算可模拟情形)

如果 F 是一族单射函数，$\mathcal{H}: X \to K$ 是 F 的 hardcore 函数，那么，对于任意变换集 $\Phi \subseteq \Phi_{\text{brs}}^{\text{srs}} \cup \mathsf{id}$，自适应 Φ-不可延展性蕴含 $\mathcal{H}$-hinting 自适应 Φ-不可延展性.

♡

定理 5.10

如果 F 是一族单射的自适应单向陷门函数，$\mathcal{H}$ 是 F 的 hardcore 函数，那么，F 是自适应 $\mathcal{H}$-hinting Φ-不可延展的，其中 $\Phi = \Phi_{\text{brs}}^{\text{srs}} \cup \{\mathsf{id}\}$.

♡

证明 根据引理 5.13，F 是自适应 Φ-不可延展的. 根据引理 5.15，F 也是自适应 $\mathcal{H}$-hinting 不可延展的. 定理得证! □

随机化的不可延展函数的构造需要用到两个密码工具：全除一有损函数 [71](可以看作是不带陷门的全除一有损单向陷门函数) 和一次签名方案. 假设 $\mathsf{ABOLF} = (\mathsf{KeyGen}, \mathsf{Eval})$ 是一个 (X, Z, τ)-全除一有损函数，记 $g_{s,vk}(x) = \mathsf{Eval}(s, vk, x)$，分支空间为 $B = \{0,1\}^d$. $\mathsf{OTS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ 是一个强不可伪造一次签名方案，其中验证密钥空间满足 $VK \subseteq B$，签名空间为 Σ，$Y = B \times Z \times \Sigma$. 令 $n = \log|X|$，$\tau = \log|Z|$. 下面构造一个从 X 到 Y 的不可延展函数.

构造 5.13 (基于 ABO-LF 的随机化 NMF)

构造所需的组件是:

- 一个全除一有损函数 $\mathsf{ABOLF} = (\mathsf{Gen}, \mathsf{Eval})$.
- 一个一次签名方案 $\mathsf{OTS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$.

构造随机化的 NMF 如下:

- $\mathsf{KeyGen}(1^\kappa)$: 输入安全参数 1^κ，输出 $s \leftarrow \mathsf{ABOLF.KeyGen}(1^\kappa, 0^d)$.
- $\mathsf{Eval}(s, x)$: 输入函数索引 s 和原像 $x \in X$，执行以下步骤:
  1. 生成一对一次签名密钥 $(vk, sk) \leftarrow \mathsf{OTS.KeyGen}(1^\kappa)$;
  2. 计算 $z \leftarrow g_{s,vk}(x)$，$\sigma \leftarrow \mathsf{OTS.Sign}(sk, z)$;
  3. 输出 $y = (vk, z, \sigma)$.
- $\mathsf{Verify}(s, x, y)$: 输入 s，x 和 $y = (vk, z, \sigma)$，如果 $z = g_{s,vk}(x) \wedge \mathsf{OTS.Verify}(vk, z, \sigma) = 1$，输出“1”，否则输出“0”.

♣

定理 5.11

令 $\mathcal{H}: X \to K = \{0,1\}^\ell$ 是 F 的一族 hardcore 函数. 则 F 是一族 $\mathcal{H}$-hinting Φ-不可延展函数，其中 $\Phi = \Phi^{\mathsf{poly}(d)} \backslash \mathsf{cf}$，$\log d \leq n - \tau - \ell - \omega(\log\kappa)$.

♡

证明 令 S_i 表示敌手在 $\mathcal{A}$ 在 Game_i 中成功的事件. 以游戏序列的方式组织证明如下:

Game_0: 该游戏是标准的 hinting 不可延展性游戏. 挑战者 $\mathcal{CH}$ 与敌手 $\mathcal{A}$ 按如下方式执行游戏.

1. 初始化: $\mathcal{CH}$ 通过运行 $s \leftarrow \mathsf{ABOLF.KeyGen}(1^\kappa, 0^d)$ 生成 F 的一个随机索引 s, 并选择 $h \xleftarrow{R} \mathcal{H}$, 将 (s, h) 发送给 $\mathcal{A}$.
2. 挑战: $\mathcal{CH}$ 选择 $x^* \xleftarrow{R} X$ 和 $(vk^*, sk^*) \leftarrow \mathsf{OTS.KeyGen}(1^\kappa)$, 计算 $z^* \leftarrow g_{s,vk^*}(x^*)$, $\sigma^* \leftarrow \mathsf{OTS.Sign}(sk^*, z^*)$, 将 $(y^* = (vk^*, z^*, \sigma^*), h(x^*))$ 发送给 $\mathcal{A}$, 其中 y^* 是函数的像值, $h(x^*)$ 是 hint 函数值.
3. 攻击: $\mathcal{A}$ 输出一对元素 $(\phi, y = (vk, z, \sigma))$. 如果 $z = g_{s,vk}(\phi(x^*))$ 并且 $\mathsf{OTS.Verify}(vk, z, \sigma) = 1$, 则 $\mathcal{A}$ 成功.

根据定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}} = \Pr[S_0]$$

Game_1: 该游戏与 Game_0 的唯一区别是 $\mathcal{A}$ 的攻击结果 (ϕ, y^*) 成功的条件定义为 $\phi(x^*) = x^* \wedge \phi \in \Phi \backslash \{\mathsf{id}\}$. 由于 $g_{s,vk^*}(\cdot)$ 是一个单射函数, z^* 的值完全决定了它的原像, 所以敌手成功的定义仅是一种概念上的变化. 故有:

$$\Pr[S_0] = \Pr[S_1]$$

Game_2: 该游戏与 Game_1 的区别在于 $\mathcal{CH}$ 在初始化阶段生成 (vk^*, sk^*) 并将 s 的生成方式由 $\mathsf{ABOLF.KeyGen}(1^\kappa, 0^d)$ 替换为 $\mathsf{ABOLF.KeyGen}(1^\kappa, vk^*)$. 根据 ABOLF 隐藏分支的性质, 敌手在两个游戏中的视图是不可以区分的, 故有:

$$|\Pr[S_1] - \Pr[S_2]| \leq \mathsf{Adv}_{\mathcal{B}_1}(\kappa)$$

其中 $\mathcal{B}_1$ 攻击 ABO-LF 隐藏有损分支性质的 PPT 敌手.

下面分析事件 S_2 发生的概率. 在 Game_2 中, 当下列条件之一成立, 则 $\mathcal{A}$ 的攻击结果 (ϕ, x) 成功

- E_1: $y = y^*$ 且 $\phi(x^*) = x^* \wedge \phi \in \Phi \backslash \mathsf{id}$.
- E_2: $vk \neq vk^*$ 且 $\mathsf{OTS.Verify}(vk, z, \sigma) = 1 \wedge g_{s,vk}(\phi(x^*)) = z \wedge \phi \in \Phi$.
- E_3: $vk = vk^* \wedge (z, \sigma) \neq (z^*, \sigma^*)$ 且 $\mathsf{OTS.Verify}(vk^*, z, \sigma) = 1 \wedge g_{s,vk^*}(\phi(x^*)) = z \wedge \phi \in \Phi$.

显然, $S_2 = E_1 \vee E_2 \vee E_3$. 下面分析概率 $\Pr[E_i]$ 的上界, 其中 $1 \leq i \leq 3$. 值得注意的是 $\mathcal{A}$ 在输出 (ϕ, y) 前的视图信息是 $\mathsf{view} = (s, h, y^* = (vk^*, z^*, \sigma^*), h(x^*))$. 则有:

$$\begin{aligned} \tilde{\mathsf{H}}_\infty(x^*|\mathsf{view}) &= \tilde{\mathsf{H}}_\infty(x^*|(z^*, \sigma^*, h(x^*))) && (5.15) \\ &= \tilde{\mathsf{H}}_\infty(x^*|(z^*, h(x^*))) && (5.16) \\ &\geq \mathsf{H}_\infty(x^*) - \tau - \ell = n - \tau - \ell && (5.17) \end{aligned}$$

在上述推导过程中, 公式 (5.15)源于 s, h 和 vk^* 与 x^* 独立这一事实. 公式 (5.16)源于 σ^* 是从 sk^* 和 z^* 导出且 sk^* 与 x^* 独立这一事实. 在 Game_2 中, $g_{s,vk^*}(\cdot)$ 是一个有损函数, 其像空间尺寸最大为 2^τ. 公式 (5.17)依据引理 2.2和 $(z^*, h(x^*))$ 最多有 $2^{\tau+\ell}$ 种可能取值这一事实.

由于 $\tilde{\mathsf{H}}_\infty(x^*|\mathsf{view}) \geq n - \tau - \ell$ 以及变换函数 $\phi \in \Phi \backslash \mathsf{id}$ 的抗碰撞性质, 故有 $\Pr[E_1] \leq 1/2^{n-\tau-\ell-\log d}$.

根据 $\phi \in \Phi$ 的输出高熵性质, 则有 $\tilde{\mathsf{H}}_\infty(\phi(x^*)|\mathsf{view}) \geq n - \tau - \ell - \log d$. 对于所有 $vk \neq vk^*$, $g_{s,vk}(\cdot)$ 是一个单射函数, 故 $z = g_{s,vk}(\phi(x^*))$ 的平均最小熵与 $\phi(x^*)$ 一样. 故有 $\Pr[E_2] \leq 1/2^{n-\tau-\ell-\log d}$.

由于选择的参数 d 满足 $\omega(\kappa) \leq n - \tau - \ell - \log d$, 所以 $\Pr[E_1]$ 和 $\Pr[E_2]$ 关于安全参数 κ 都是可忽略的. 根据一次签名的强不可伪造性, 则有 $\Pr[E_3] \leq \mathsf{Adv}_{\mathcal{B}_2}(\kappa)$, 其中 $\mathcal{B}_2$ 是攻击一次签名强不可伪造性的 PPT 敌手.

通过上述分析, 可得 $\Pr[S_2]$ 关于安全参数 κ 是可忽略的.

综上, 定理 5.11得证! □.

注记 5.1

F 的一族 hardcore 函数可以是一个定义在 X 到 $\{0,1\}^\ell$ 的一族一致哈希函数. ♠

不可延展函数的应用

2015 年, 秦等 [91] 将不可延展密钥密钥派生函数 [92] 推广为连续不可延展密钥派生函数 (continuous non-malleable non-malleable key derivation function, CNM-KDF), 并提出一种利用 CNM-KDFs 构造抗相关密钥攻击的密码原语, 如公钥加密、数字签名、身份加密等的通用模式. 2022 年, 陈等 [325] 进一步简化了此概念的名称, 称

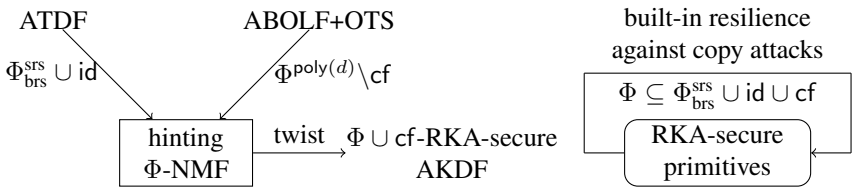


图 5.21: 不可延展函数的构造及应用.

之为抗相关密钥攻击的可认证密钥派生函数 (authenticated non-malleable key derivation function, AKDF), 使其名称更能展现出它的性质, 并且增强了 CNM-KDF 的安全性. 以 AKDF 为跳板, 可以设计出性能良好、变换函数集范围广的多种抗相关密钥攻击的密码原语. 图 5.21展示了不可延展函数的构造及其在 RKA 安全密码原语方面的应用.

下面介绍 AKDFs 的形式化定义及其基于不可延展函数的通用构造.

定义 5.12 (可认证密钥派生函数)

一个可认证密钥派生函数 AKDF 包含以下 3 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 输入安全参数 1^κ, 输出系统参数 pp, 并定义了原始密钥空间 X, 认证标签空间 T 和派生密钥空间 K.
- $\mathsf{Sample}(pp)$: 输入公开参数 pp, 选择一个原始密钥 $x \xleftarrow{R} X$, 计算它的认证标签 $t \in T$, 输出 (x, t).
- $\mathsf{Derive}(x, t)$: 输入原始密钥 $x \in X$ 和标签 $t \in T$, 输出一个派生密钥 $k \in K$ 或者一个拒绝符号 $\perp$, 表示 t 不是 x 的合法标签.

♣

相关密钥攻击安全性. 令 Φ 是一个定义在原始密钥空间 X 上的一个变换函数集. 假设 Φ 包含单位变换 id. 定义 AKDFs 敌手的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left| \Pr\left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (x^*, t^*) \leftarrow \mathsf{Sample}(pp); \\ k_0^* \leftarrow \mathsf{Derive}(x^*, t^*), k_1^* \xleftarrow{R} K; \\ \beta \xleftarrow{R} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}^{\Phi}_{\mathsf{derive}}}(pp, t^*, k_\beta^*); \end{array} \right] - \frac{1}{2} \right| .$$

对于任意 PPT 敌手 $\mathcal{A}$, 如果优势函数 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 关于安全参数 κ 是可忽略的, 那么称 AKDF 是 Φ-RKA-安全的. 其中, $\mathcal{O}^{\Phi}_{\mathsf{derive}}$ 是相关密钥派生谕言机, 输入 $\langle \phi, t \rangle \neq \langle \mathsf{id}, t^* \rangle$, 返回 $\mathsf{Derive}(\phi(x^*), t)$. 在实验中, $\mathcal{A}$ 可以自适应地询问谕言机 $\mathcal{O}^{\Phi}_{\mathsf{derive}}$. 但是, 敌手不能进行形如 $\langle \mathsf{id}, t^* \rangle$ 的非法询问, 否则敌手必定成功. 根据 $\phi \in \mathsf{cf}$ 还是 $\phi \in \Phi$, 合法询问 $\langle \phi, t \rangle \neq \langle \mathsf{id}, t^* \rangle$ 可以进一步分为常量询问和非常量询问.

笔记 可认证密钥派生函数与传统的密钥派生函数主要区别在于多了一个认证标签 t. t 的主要作用是保障原始密钥 x^* 的完整性, 使得 x^* 的篡改结果 $\phi(x^*)$ 能够被有效检测出来, 从而拒绝输出 $\phi(x^*)$ 的派生密钥, 即 $k \leftarrow \mathsf{Derive}(\phi(x^*), t)$.

在证明 AKDF 方案的 RKA 安全性时, 首要任务是在不知道原始密钥 x^* 的情况下如何回答敌手的相关密钥派生询问. 一种简单粗暴的方式是与其设法回答敌手的相关密钥派生询问, 不如直接拒绝回答敌手所有的 RKA 询问. 即使敌手在看到挑战信息 (x^*, t^*) 的情况下, 也无法生成一个合法的询问 $\langle \phi, t \rangle$, 使得 t 是 $\phi(x^*)$ 的合法认证标签. 因此, 在回答敌手的相关密钥派生询问时, 挑战者 (模拟者) 直接返回 $\perp$ 即可. 下面介绍如何利用不可延展函数巧妙地构造抗相关密钥攻击的可认证密钥派生函数.

构造 5.14 (基于 NMF 的 AKDF 构造)

令 Φ 是一个包含恒等变换 id 的变换集. 构造所需组件是:

- 一个 hardcore 函数 $\mathcal{H} : X \to K$.

- 一个 $\mathcal{H}$-hinting Φ-不可延展函数族 $F = (\mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Verify})$.

构造 AKDF 如下:

- $\mathsf{Setup}(1^\kappa)$: 输入安全参数 1^κ, 运行 $f \leftarrow F.\mathsf{KeyGen}(1^\kappa)$, 选择一个 hardcore 函数 $h \xleftarrow{R} \mathcal{H}$, 输出公开参数 $pp = (f, h)$.
- $\mathsf{Sample}(pp)$: 输入公开参数 $pp = (f, h)$, 随机选择一个原始密钥 $x \xleftarrow{R} X$, 计算 $t \leftarrow f(x)$, 输出 (x, t).
- $\mathsf{Derive}(x, t)$: 输入原始密钥 x 和标签 t, 如果 $\mathcal{F}.\mathsf{Verify}(f, x, t) = 1$ 成立, 则输出 $k \leftarrow h(x)$, 否则输出 $\perp$.

定理 5.12

如果 F 是 $\mathcal{H}$-hinting Φ-不可延展的, 那么构造 5.14是一族 Φ'-RKA 安全的 AKDF, 其中 $\Phi' = \Phi \cup \mathsf{cf}$.

证明 令 S_i 表示事件 "敌手在 Game_i 中成功". 以游戏序列的方式组织证明如下:

Game_0: 该游戏是 AKDF 的标准 RKA 安全性游戏. 挑战者 $\mathcal{CH}$ 与敌手 $\mathcal{A}$ 按如下方式执行游戏:

1. 初始化: $\mathcal{CH}$ 生成函数索引 $f \leftarrow F.\mathsf{KeyGen}(1^\kappa)$ 并选择一个相应的 hardcore 函数 $h \xleftarrow{R} \mathcal{H}$, 然后将 $pp = (f, h)$ 发送给 $\mathcal{A}$.
2. 挑战: $\mathcal{CH}$ 随机采样一个原始密钥 $x^* \xleftarrow{R} X$, 计算 $t^* \leftarrow f(x^*)$, $k_0^* \leftarrow h(x^*)$, 选择 $k_1^* \xleftarrow{R} K$, $\beta \xleftarrow{R} \{0, 1\}$, 然后将 (pp, t^*, k_β^*) 作为挑战信息发送给 $\mathcal{A}$.
3. 相关密钥派生询问: 当收到合法询问 $\langle \phi, t \rangle \neq \langle \mathsf{id}, t^* \rangle$ 时, 如果 $\mathcal{F}.\mathsf{Verify}(f, \phi(x^*), t) = 1$ 成立, $\mathcal{CH}$ 返回 $h(\phi(x^*))$, 否则返回 $\perp$. 特别地, 对于常量查询, $\mathcal{CH}$ 不需要利用 x^* 就可以进行回答.
4. 猜测: $\mathcal{A}$ 输出一个猜测比特 β', 如果 $\beta' = \beta$ 成立, 则 $\mathcal{A}$ 成功.

根据 RKA 安全性的定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game_1(拒绝所有非常量询问): 该游戏与 Game_0 的唯一不同之处在于处理非常量询问的方式. 对于所有非常量询问 $\langle \phi, t \rangle$, 其中 $\phi \in \Phi$, $\mathcal{CH}$ 直接返回 $\perp$. 令 E 为事件 "存在 $\mathcal{A}$ 的非常量询问 $\langle \phi, t \rangle$ 满足条件 $F.\mathsf{Verify}(f, \phi(x^*), t) = 1$". 根据 Game_0 和 Game_1 的定义, 如果事件 E 发生, 在 Game_1 中, $\mathcal{CH}$ 直接返回 $\perp$, 而在 Game_0 中, $\mathcal{CH}$ 返回 $\phi(x^*)$. 显而易见, 对于任意 PPT 敌手 $\mathcal{A}$, 当事件 E 未发生时, $\mathcal{A}$ 在 Game_0 和 Game_1 中的视图是完全一样的. 根据 Difference Lemma, 则有:

$$|\Pr[S_0] - \Pr[S_1]| \leq \Pr[E]$$

根据引理 5.16, 当 F 满足 $\mathcal{H}$-hinting Φ-不可延展性时, 事件 E 发生的概率是可忽略的.

引理 5.16

$\Pr[E] \leq \mathsf{poly}(\kappa) \cdot \mathsf{Adv}_{\mathcal{B}_1}(\kappa)$, 其中 $\mathcal{B}_1$ 是攻击 F $\mathcal{H}$-hinting Φ-不可延展性的敌手.

Game_2 ($k_0^* \xleftarrow{R} K_1$): 该游戏与 Game_1 的唯一不同之处在于 $\mathcal{CH}$ 随机选择 $k_0^* \xleftarrow{R} K$, 而不是通过 $k_0^* \leftarrow h(x^*)$ 计算而来. 显然, 如果存在一个敌手 $\mathcal{A}$ 在 Game_1 和 Game_2 中的视图存在差异 $\epsilon(\kappa)$, 则可以构造一个归约算法以至少 $\epsilon(\kappa)/2$ 的优势攻破 hardcore 函数的伪随机性. 故有:

$$|\Pr[S_1] - \Pr[S_2]| \leq 2\mathsf{Adv}_{\mathcal{B}_1}(\kappa)$$

其中 $\mathcal{B}_2$ 是攻击 hardcore 函数伪随机性的敌手.

在 Game_2 中, k_β 的值与 β 完全独立. 故, $\Pr[S_2] = 1/2$.

综上, 定理 5.12得证! □

引理 5.16的证明如下:

证明 令 $\mathcal{B}_1$ 是一个攻击 F 的 $\mathcal{H}$-hinting Φ-不可延展性的敌手. 给定挑战信息 $(f, h, y^*, h(x^*))$, 其中 $f \leftarrow F.\mathsf{KeyGen}(1^\kappa)$, $h \xleftarrow{R} \mathcal{H}$, $y^* \leftarrow f(x^*)$ 并且 $x^* \xleftarrow{R} X$, $\mathcal{B}_1$ 按以下方式模拟 $\mathcal{A}$ 在 Game_1 中的挑战者: 令 $pp = (f, h)$, $t^* = y^*$,

$k_0^* \leftarrow h(x^*)$, 选择 $k_1^* \xleftarrow{R} K$, $\beta \xleftarrow{R} \{0,1\}$, 然后将 (pp, t^*, k_β^*) 发送给 $\mathcal{A}$. 尽管 $\mathcal{B}_1$ 并不知道 x^* 的值, 然而, 在 Game_1 中对于所有非常量查询, $\mathcal{B}$ 直接返回 $\perp$, 所以这并不影响 $\mathcal{B}_1$ 模拟回答 $\mathcal{A}$ 的询问. 由此可知, $\mathcal{B}_1$ 能完美地模拟 $\mathcal{A}$ 在 Game_1 中的视图环境. 令 L 为 $\mathcal{A}$ 的所有非常量询问列表. 由于 $\mathcal{A}$ 是一个 PPT 敌手, 所以 $|L| \leq \mathsf{poly}(\kappa)$. 最后, $\mathcal{B}_1$ 从列表 L 中随机选择一组元素 $\langle \phi, t \rangle$ 作为攻击 F $\mathcal{H}$-hinting Φ-不可延展性的结果. 当事件 E 发生时, $\mathcal{B}_1$ 成功的概率至少是 $1/\mathsf{poly}(\kappa)$. 因此, $\mathcal{B}$ 成功的优势至少为 $\Pr[E]/\mathsf{poly}(\kappa)$. 如果 $\Pr[E]$ 是不可忽略的, 那么 $\mathcal{B}_1$ 的优势也是不可忽略的, 与 $\mathcal{F}$ 的不可延展性矛盾. 特别地, 由于 $\mathcal{B}_1$ 攻击 F 不可延展性成功的概率不超过 $\mathsf{Adv}_{\mathcal{B}_1}(\kappa)$, 所以 $\Pr[E]/\mathsf{poly}(\kappa) \leq \mathsf{Adv}_{\mathcal{B}_1}(\kappa)$. 引理得证! □

5.3 消息依赖密钥安全

水能载舟, 亦能覆舟.

—《荀子 · 哀公》

在传统的公钥加密方案中, 加密的消息一般是根据明文空间的某一概率分布选择的, 而与加密方案的私钥无关. 然而在硬盘加密、匿名证书系统等应用环境下, 加密的消息与私钥有关甚至是私钥本身, 如 $f(sk)$, 这里的 f 是一个从密钥空间到消息空间的函数. 因此, 传统的安全模型并不能完全满足这类应用的需求. 实际上, 早在 1984 年 Goldwasser 和 Micali [236] 提出概率加密方案时, 已经指出当加密的消息与密钥相关时, 无法保证方案的语义安全性. 针对这一问题, Black 等 [53] 提出消息依赖密钥安全性 (key-dependent message security, KDM-security) 的概念. Camenisch 和 Lysyanskaya [52] 针对多用户环境提出的循环加密安全性 (circular security) 也可以看作是一种特殊的 KDM 安全性. 通俗地讲, 即使敌手获得一些与私钥相关的消息的密文, KDM 安全性仍然能够保障方案的语义安全性. KDM 安全性不仅能够解决实际应用中面临的安全问题, 而且还可以用于设计 CCA 安全的公钥加密方案和单向陷门函数 [326].

针对不同的应用环境, KDM 安全性可由不同形式的私钥函数族刻画. 一般情况下, 简单的仿射函数即可满足需求. 然而, 即使在这种情况下, 设计 KDM 安全的公钥加密方案也是相当困难的. 2008 年, Boneh 等 [51] 利用私钥的密文公开可计算性的思想, 设计了第一个标准模型下基于 DDH 假设的循环加密方案. 后来, 学者们基于类似思想提出了不同计算假设下的 KDM 安全的公钥加密方案, 如 Applebaum 等基于 LWE 假设的方案 [54] 和 Brakerski 等 [319] 基于 QR 和 DCR 的方案. 尽管这些方案的 KDM 安全性仅针对简单的仿射函数族, 通过扩大 KDM 函数族的技术 [327, 328, 57, 55], 可以解决这一问题. 2016 年, Wee [56] 将这些方案的设计思想统一为同态哈希证明系统技术. 尽管这些方案具有 KDM 安全性, 但是仅能抵抗选择明文攻击. 对于选择密文攻击, 由于私钥的密文公开可计算性与解密服务之间是相矛盾的, 因此设计抗选择密文攻击的 KDM 安全加密方案更具有挑战性. 一种方式是利用从 CPA 到 CCA 转化的 Naor-Yung“双密钥加密范式” [58]. 另一种方式是寻找特殊的密码工具实现 KDM-CCA 安全性, 如有损代数过滤器 [314]、辅助输入安全的认证加密 [59] 等, 或者针对特殊形式的 KDM 函数族设计 KDM-CCA 安全的公钥加密方案, 如秦等 [329] 证明了经典的 IND-CCA 安全的 Cramer-Shoup 方案 [285] 在加密不同用户私钥之差时满足 KDM-CCA 安全性. 此外, KDM 安全性在拓展的属性加密、身份加密等密码原语中也有着重要意义, 学者们也提出了不同密码原语的 KDM 安全性方案构造 [330, 331, 332].

本节内容主要介绍消息依赖密钥的安全性模型、基于同态哈希证明系统的 KDM-CPA 安全 PKE 的通用构造方法和 KDM 安全性从 PKE 到 IBE 的转化方法.

5.3.1 消息依赖密钥安全模型

在消息依赖密钥安全模型中, 存在一个与密钥相关的函数集合 $\mathcal{F}$ 将 (一组) 密钥映射到消息空间. 与密钥泄漏安全模型不同, 消息依赖密钥加密泄漏的不是该密钥的函数值而是它的密文, 如 $\mathsf{Encrypt}(pk, f(sk))$.

KDM-CCA 安全性. 对于任意 $n \in \mathbb{N}$, 令 $\mathcal{F} = \{f : SK^n \to M\}$ 是一个从 n 维密钥空间到消息空间的 KDM 函数族. 定义公钥加密方案 PKE 的 KDM-CCA 敌手 $\mathcal{A}$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left| \Pr \left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk_i, sk_i) \leftarrow \mathsf{KeyGen}(pp), \forall i \in [n]; \\ \text{Set } \mathsf{CL} = \emptyset, \vec{pk} = (pk_1, \ldots, pk_n), \vec{sk} = (sk_1, \ldots, sk_n); \\ \beta \xleftarrow{R} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}^{\beta}_{\mathsf{encrypt}}, \mathcal{O}_{\mathsf{decrypt}}}(pp, \vec{pk}); \end{array} \right] - \frac{1}{2} \right|$$

其中, 加密谕言机和解密谕言机的定义如下:

- 加密谕言机 $\mathcal{O}^{\beta}_{\mathsf{encrypt}}$: 输入 (i, f), 其中 $i \in [n]$, $f \in \mathcal{F}$, 如果 $\beta = 0$, 返回 $c = \mathsf{Encrypt}(pk_i, f(\vec{sk}))$; 如果 $\beta = 1$, 返回 $c = \mathsf{Encrypt}(pk_i, 0^{|M|})$. 最后, 将 (i, c) 添加至密文列表 CL 中.
- 解密谕言机 $\mathcal{O}_{\mathsf{decrypt}}$: 输入 (i, c), 其中 $i \in [n]$. 如果 $(i, c) \in \mathsf{CL}$, 返回 $\perp$; 否则, 返回 $\mathsf{Decrypt}(sk_i, c)$.

上述定义中, 如果对于任意的 PPT 敌手 $\mathcal{A}$, 优势函数 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 是可忽略的, 则称公钥加密方案 PKE 是 $\mathcal{F}$-KDM-CCA 安全的. 如果不允许敌手访问解密谕言机, 则则称公钥加密方案 PKE 是 $\mathcal{F}$-KDM-CPA 安全的.

笔记 KDM-CCA 安全模型说明了敌手在解密服务的帮助下, 也无法区分一组密文加密的是私钥相关的函数值还是某一固定消息, 例如 $0^{|M|}$. 不同类型的函数族 $\mathcal{F}$ 对于是实现 KDM 安全性的难度是不同的. 若 $\mathcal{F}$ 是常数函数族 $\{f_m : \vec{sk} \to m\}_{m\in M}$, 则 KDM-CPA 安全性等价于传统的语义安全性 (IND-CPA). 而 KDM-CCA 安全性即是传统的 IND-CCA 安全性. 若 $\mathcal{F}$ 是选择函数族 $\{f_i : \vec{sk} \to sk_i\}$, 此时的 KDM 安全性也称之为循环加密安全性. 消息依赖密钥加密也可以看作是一种特殊的密钥泄漏函数. Brakerski 等设计的 KDM-CPA 安全的 PKE 方案同时满足 LR-CPA 安全性, 也说明二者之间存在一定的联系.

通过上述定义可以看出, KDM 安全性蕴含语义安全性, 反之未必成立. 事实上, 不是所有语义安全的加密方案都是 KDM 安全的 [333].

构造 5.15 (KDM-PKE 反例构造)

假设 $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ 是任意一个语义安全的 PKE 方案. 在该方案的基础上构造一个新的加密方案 $\mathsf{PKE}' = (\mathsf{Setup}', \mathsf{KeyGen}', \mathsf{Encrypt}', \mathsf{Decrypt}')$, 其中 Setup' 和 KeyGen' 与原方案一样, $\mathsf{Encrypt}'$ 和 $\mathsf{Decrypt}'$ 的定义如下:

$$\mathsf{Encrypt}'(m) = \begin{cases} \mathsf{Encrypt}(m)||0 & \text{if } m \neq sk \\ \mathsf{Encrypt}(m)||1 & \text{if } m = sk \end{cases} \qquad \mathsf{Decrypt}'(c||b) = \begin{cases} \mathsf{Decrypt}(c) & \text{if } b = 0 \\ sk & \text{if } b = 1 \end{cases}.$$

♣

在语义安全性模型中, 消息是从明文空间中公开选取的, 被加密的消息等于私钥 sk 的概率是可忽略的, 所以密文的形式以压倒性的概率是 $\mathsf{Encrypt}(m)||0$. 由此可得, PKE' 仍然是语义安全的. 在消息依赖密钥加密模型中, 根据挑战比特 β 的不同, 消息可能等于或不等于私钥 sk, 并且两种情况下的密文形式是可以直接区分的. 由此可得, PKE' 不是 KDM 安全的.

5.3.2 KDM-CPA 安全 PKE 的通用构造方法

本节介绍一种基于同态哈希证明系统的 KDM-CPA 安全公钥加密方案. 同态哈希证明系统是一种特殊的哈希证明系统, 除了具有私有可计算、公开可计算、平滑性等性质外, 还需要具有同态性.

定义 5.13 (同态哈希证明系统)

令 $\mathsf{HPS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PrivEval}, \mathsf{PubEval})$ 是一个哈希证明系统. 运行 $\mathsf{Setup}(1^\kappa)$ 输出一组公开参数 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$. 运行 $\mathsf{KeyGen}(pp)$ 将输出一对密钥 (pk, sk), 其中 $sk \xleftarrow{R} SK$, $pk = \alpha(sk)$ 也称为投射密钥. 如果 HPS 满足如下性质:

- 私有可计算性: 对于任意 $x \in X$, 存在算法 $\mathsf{PrivEval}(sk, x)$, 输出 $\pi = \mathsf{H}_{sk}(x)$.
- 公开可计算性: 对于任意 $x \in L$ 以及相应的 w, 存在算法 $\mathsf{PubEval}(pk, x, w)$, 输出 $\pi = \mathsf{H}_{sk}(x)$.
- 平滑性: 在输入 $x \xleftarrow{R} X$ 时, $\mathsf{H}_{sk}(x)$ 与 Π 上的均匀分布统计接近, 即:

$$(pk, \mathsf{H}_{sk}(x)) \approx_s (pk, \pi)$$

  其中 $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$, $\pi \xleftarrow{R} \Pi$.
- 同态性: 对于所有 $sk \in SK$ 和所有 $x_0, x_1 \in X$, 则有 $\mathsf{H}_{sk}(x_0) \cdot \mathsf{H}_{sk}(x_1) = \mathsf{H}_{sk}(x_0 \cdot x_1)$.

则称 HPS 是一个同态哈希证明系统.

♣

构造 5.16 (基于同态哈希证明系统的 KDM-CPA 安全 PKE)

构造所需的组件是:

- 一个同态哈希证明系统 $\mathsf{HPS} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{PrivEval}, \mathsf{PubEval})$.
- 一个从消息空间 M 到哈希值空间 Π 的可公开计算且可逆的映射 $\phi : M \to \Pi$.

构造 KDM-CPA PKE 如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp \leftarrow \mathsf{HPS.Setup}(1^\kappa)$, 输出系统参数 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$.
- $\mathsf{KeyGen}(pp)$: 运行 $(pk, sk) \leftarrow \mathsf{HPS.KeyGen}(pp)$, 输出公钥 pk 和私钥 sk, 其中 $sk \xleftarrow{R} SK$, $pk = \alpha(sk)$.
- $\mathsf{Encrypt}(pk, m)$: 以公钥 pk 和明文 $m \in M$ 为输入, 执行如下步骤:
  1. 运行 $(x, w) \leftarrow \mathsf{SampRel}(r)$ 生成随机实例 $x \in L$ 及相应的证据 w, 其中 r 为采样时使用的随机数;
  2. 通过 $\mathsf{HPS.PubEval}(pk, x, w)$ 计算实例 x 的哈希证明 $\pi = \mathsf{H}_{sk}(x)$;
  3. 计算 $\psi = \pi \cdot \phi(m)$;
  4. 输出密文 $c = (x, \psi)$.
- $\mathsf{Decrypt}(sk, c)$: 以私钥 sk 和密文 $c = (x, \psi)$ 为输入, 计算 $m' = \phi^{-1}(\mathsf{H}_{sk}(x)^{-1} \cdot \psi)$ 并返回明文 m'. ♣

正确性. 方案的正确性可由哈希证明系统的正确性保证, 安全性由如下定理保证.

定理 5.13

如果 HPS 满足平滑性和同态性, 并且 $L \subseteq X$ 上的 SMP 困难问题成立, 那么构造 5.16 中的 PKE 是 $\mathcal{F}$-KDM-CPA 安全的, 其中 $\mathcal{F} = \{f_{e,k} : sk \to \phi^{-1}(\mathsf{H}_{sk}(e) \cdot k) \mid e \in X, k \in \Pi\}$. ♡

定理 5.13的证明思路主要是将密钥函数值 $f_{x,\pi}(sk)$ 的密文转化为函数参数 (x, π) 的密文, 由此使得 KDM 密文与私钥 sk 无关. 转化的技术是 HPS 的同态性. 即, 将 $f_{x,\pi}(sk)$ 的密文:

$$\mathsf{Encrypt}(pk, f_{e,k}(sk)) = (x, \mathsf{H}_{sk}(x) \cdot f_{e,k}(sk))$$

转化为

$$\mathsf{Encrypt}(pk, f_{e,k}(sk)) = (x \cdot e^{-1}, \mathsf{HPS.PubEval}(pk, x, w) \cdot k). \tag{5.18}$$

从而使得挑战者在不知道私钥 sk 的情况下, 也可以回答敌手的 KDM 加密询问. 下面给出详细的证明过程.

证明 令 S_i 表示敌手在 Game_i 中的成功事件. 以游戏序列的方式组织证明如下:

Game_0: 该游戏是标准的 KDM-CPA 游戏, 挑战者 $\mathcal{CH}$ 和敌手 $\mathcal{A}$ 交互如下:

- 初始化: $\mathcal{CH}$ 运行 $\mathsf{Setup}(1^\kappa)$ 生成公开参数 pp, 同时运行 $\mathsf{KeyGen}(pp)$ 生成公私钥对 (pk, sk). $\mathcal{CH}$ 将 (pp, pk) 发送给 $\mathcal{A}$.
- 挑战: $\mathcal{CH}$ 选择随机比特 $\beta \in \{0, 1\}$.
- 询问: 对于敌手的任意询问 $f_{e,k} \in \mathcal{F}$, $\mathcal{CH}$ 作如下计算:
  1. 如果 $\beta = 0$, $\mathcal{CH}$ 随机选择 $x \in L$ 及相应的证据 w, 计算密文 $C = (x, \psi)$, 其中:
  $$\psi = \mathsf{HPS.PubEval}(pk, x, w) \cdot \phi(f_{e,k}(sk)) = \mathsf{HPS.PubEval}(pk, x, w) \cdot \mathsf{H}_{sk}(e) \cdot k$$
  2. 如果 $\beta = 1$, $\mathcal{CH}$ 随机选择 $x \in L$ 及相应的证据 w, 计算密文 $C = (x, \psi)$, 其中:
  $$\psi = \mathsf{HPS.PubEval}(pk, x, w) \cdot \phi(0^{|M|})$$
  3. 最后, $\mathcal{CH}$ 将密文 $C = (x, \psi)$ 返回给敌手.
- 猜测: $\mathcal{A}$ 输出对 β 的猜测 β'. $\mathcal{A}$ 成功当且仅当 $\beta' = \beta$.

根据定义, 则有:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|$$

Game_1: 该游戏与 Game_0 的唯一不同在于 $\beta = 0$ 时加密谕言机的工作方式. 具体地, 对于敌手的任意加密询问 $f \in \mathcal{F}$, 当 $\beta = 0$ 时, $\mathcal{CH}$ 返回形如公式 5.18 中的密文.

假设 $\mathcal{A}$ 询问加密谕言机的次数最多为 Q 次, 则可以利用混合游戏的思想在 Game_0 和 Game_1 之间定义 $Q-1$ 个混合游戏 $\mathsf{Game}_{0,i}$, 其中 $i \in \{1, \ldots, Q-1\}$. 在 $\mathsf{Game}_{0,i}$ 中, 当 $\beta = 0$ 时, $\mathcal{A}$ 的前 i 个询问的密文是公式 5.18 中的形式, 而后 $Q-i$ 次询问的密文按正常方式加密得来. 显然, $\mathsf{Game}_{0,0} = \mathsf{Game}_0$, $\mathsf{Game}_{0,Q} = \mathsf{Game}_1$. 对于任意的 i,

敌手在两个连续游戏中的视图是不可区分的, 即 $\text{Game}_{0,i-1} \approx \text{Game}_{0,i}$. 特别地, 对于任意攻击子集成员判定问题困难性的敌手 $\mathcal{B}_1$, 则有 $|\Pr[S_{0,i-1}] - \Pr[S_{0,i}]| \leq 2\mathsf{Adv}_{\mathcal{B}_1}(\kappa)$. 这是因为

$$
\begin{aligned}
& \mathsf{Encrypt}(pk, f_{e,k}(sk)) & \\
= \ & (x, \mathsf{HPS.PubEval}(pk, x, w) \cdot \mathsf{H}_{sk}(e) \cdot k) & //(x,w) \leftarrow \mathsf{SampRel}(r) \\
= \ & (x, \mathsf{H}_{sk}(x) \cdot \mathsf{H}_{sk}(e) \cdot k) & //\text{投射性质} \\
\approx_c \ & (x, \mathsf{H}_{sk}(x) \cdot \mathsf{H}_{sk}(e) \cdot k) & //x \xleftarrow{\mathrm{R}} X, \text{SMP 问题} \\
= \ & (x, \mathsf{H}_{sk}(x \cdot e) \cdot k) & //x \xleftarrow{\mathrm{R}} X, \text{同态性质} \\
= \ & (x \cdot e^{-1}, \mathsf{H}_{sk}(x) \cdot k) & //x \xleftarrow{\mathrm{R}} L, \text{SMP 问题} \\
\approx_c \ & (x \cdot e^{-1}, \mathsf{H}_{sk}(x) \cdot k) & //(x,w) \leftarrow \mathsf{SampRel}(r) \\
= \ & (x \cdot e^{-1}, \mathsf{HPS.PubEval}(pk, x, w) \cdot k) & //(x,w) \leftarrow \mathsf{SampRel}(r), \text{投射性质}
\end{aligned}
$$

特别注意, 在上式的演进过程中, 密钥 sk 是完全公开的. 因此, 在前 i 次询问时, $\mathcal{CH}$ 可以用公钥和 KDM 函数 $f_{e,k}$ 计算密文 $(x \cdot e^{-1}, \mathsf{HPS.PubEval}(pk, x, w) \cdot k)$, 而对于后 $Q - i$ 次询问, $\mathcal{CH}$ 可以用私钥 sk 和 KDM 函数 $f_{e,k}$ 计算密文 $\mathsf{Encrypt}(pk, f_{e,k}(sk))$. 由此可知,

$$|\Pr[S_1] - \Pr[S_2]| \leq 2Q\mathsf{Adv}_{\mathcal{B}_1}(\kappa)$$

Game_2: 该游戏与 Game_1 的唯一不同在于 $\beta = 0$ 时加密谕言机的工作方式. 具体地, 对于敌手的任意加密询问 $f \in \mathcal{F}$, 当 $\beta = 0$ 时, $\mathcal{CH}$ 返回一个随机密文 (x, ψ), 其中 $x \xleftarrow{\mathrm{R}} X, \psi \xleftarrow{\mathrm{R}} \Pi$. 在 Game_1 中, 由于 KDM 密文可以由公钥 pk 和 KDM 函数的参数 (e, k) 公开计算, 因此, 只需要证明

$$(x \cdot e^{-1}, \mathsf{HPS.PubEval}(pk, x, w) \cdot k) \approx (x, \psi)$$

其中 $(x, w) \leftarrow \mathsf{SampRel}(r), x \xleftarrow{\mathrm{R}} X, \psi \xleftarrow{\mathrm{R}} \Pi$. 这是因为

$$
\begin{aligned}
& (x \cdot e^{-1}, \mathsf{HPS.PubEval}(pk, x, w) \cdot k) & //(x,w) \leftarrow \mathsf{SampRel}(r) \\
\approx_c \ & (x \cdot e^{-1}, \mathsf{H}_{sk}(x) \cdot k) & //(x,w) \leftarrow \mathsf{SampRel}(r), \text{投射性质} \\
\approx_c \ & (x \cdot e^{-1}, \mathsf{H}_{sk}(x) \cdot k) & //x \xleftarrow{\mathrm{R}} X, \text{SMP 问题} \\
\approx_s \ & (x \cdot e^{-1}, \pi \cdot k) & //x \xleftarrow{\mathrm{R}} X, \pi \xleftarrow{\mathrm{R}} \Pi, \text{平滑性} \\
= \ & (x, \psi) & //x \xleftarrow{\mathrm{R}} X, \psi \xleftarrow{\mathrm{R}} \Pi
\end{aligned}
$$

由此可知, 在游戏 Game_2 中, 当 $\beta = 0$, 加密谕言机返回的密文都是随机的, 与 KDM 函数值 $f_{e,k}(sk)$ 无关. 特别地

$$|\Pr[S_2] - \Pr[S_3]| \leq \mathsf{Adv}_{\mathcal{B}_1}(\kappa) + \mathsf{Adv}_{\mathcal{B}_2}(\kappa)$$

Game_3: 该游戏与 Game_2 的唯一不同在于 $\beta = 1$ 时加密谕言机的工作方式. 具体地, 对于敌手的任意加密询问 $f \in \mathcal{F}$, 当 $\beta = 1$ 时, $\mathcal{CH}$ 返回一个随机密文 (x, ψ), 其中 $x \xleftarrow{\mathrm{R}} X, \psi \xleftarrow{\mathrm{R}} \Pi$. 当 $\beta = 1$ 时, 加密谕言机返回的密文形式是 $\mathsf{Encrypt}(pk, 0^{|M|})$. 利用 HPS 的平滑性, 可以直接证明消息 $0^{|M|}$ 的密文与一个随机密文是不可区分的, 即

$$|\Pr[S_2] - \Pr[S_3]| \leq \mathsf{Adv}_{\mathcal{B}_2}(\kappa)$$

其中 $\mathcal{B}_2$ 是攻击同态哈希证明系统平滑性的敌手.

在 Game_3 中, 不管 $\beta = 0$ 还是 $\beta = 1$, 加密谕言机都返回一个随机密文, 与挑战比特 β 完全无关. 由此, 可得

$$\Pr[S_3] = 1/2$$

综上, 定理 5.13得证! □

下面介绍一种 L_{nDDH} 语言的同态哈希证明系统. L_{nDDH} 语言可以看作是 L_{DDH} 的扩展. 首先, 运行 $(\mathbb{G}, q, g) \leftarrow \mathsf{GenGroup}(1^\kappa)$, 其中 $\mathbb{G}$ 是一个阶为素数 q, g 是生成元的有限循环群, 且 DDH 问题在群 $\mathbb{G}$ 上是困难的. 令 n 为任意正整数, $X = \mathbb{G}^n, W = \mathbb{Z}_q$. 随机选择 $g_1, g_2, \ldots, g_n \xleftarrow{\mathrm{R}} \mathbb{G}$, 则群 $\mathbb{G}$ 上的 $\mathcal{NP}$ 语言定义如下:

$$L_{\text{nDDH}} = \{(x_1, x_2, \ldots, x_n) \in X : \exists w \in W \text{ s.t. } x_1 = g_1^w \wedge x_2 = g_2^w \wedge \ldots \wedge x_n = g_n^w)\}$$

可以验证, DDH 假设蕴含 $L_{\text{nDDH}} \subset X$ 上的 SMP 困难问题成立.

构造 5.17 (L_{nDDH} 语言的同态 HPS 构造)

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入，运行 $(\mathbb{G}, q, g) \leftarrow \mathsf{GenGroup}(1^\kappa)$，随机选择 n 个生成元 $g_1, g_2, \ldots, g_n \xleftarrow{R} \mathbb{G}^n$，输出系统参数 $pp = (\mathsf{H}, SK, PK, X, L, W, \Pi, \alpha)$. 其中，

$$PK = \mathbb{G}, SK = \{0,1\}^n, X = \mathbb{G}^n, L = L_{\text{nDDH}}, W = \mathbb{Z}_q, \Pi = \mathbb{G}$$

对于任意 $sk = (s_1, s_2, \ldots, s_n) \in SK$ 和 $(x_1, x_2, \ldots, x_n) \in X$, α 和 H 的定义如下:

$$\alpha(sk) = g_1^{s_1} \cdot g_2^{s_2} \cdots g_n^{s_n} \in \mathbb{G} \qquad \mathsf{H}_{sk}(x) = x_1^{s_1} \cdot x_2^{s_2} \cdots x_n^{s_n} \in \mathbb{G}$$

- $\mathsf{KeyGen}(pp)$: 以公开参数 pp 为输入，随机采样 $sk = (s_1, s_2, \ldots, s_n) \xleftarrow{R} \{0,1\}^n$，计算 $pk = \alpha(sk)$，输出 (pk, sk).
- $\mathsf{PrivEval}(sk, x)$: 以私钥 sk 和 $x = (x_1, x_2, \ldots, x_n) \in X$ 为输入，输出 $\pi = \mathsf{H}_{sk}(x)$.
- $\mathsf{PubEval}(pk, x, w)$: 以公钥 pk、$x \in L$ 以及相应的 w 为输入，输出 $\pi = pk^w$. 以下公式说明了公开求值算法的正确性:

$$pk^w = (g_1^{s_1} \cdot g_2^{s_2} \cdots g_n^{s_n})^w = x_1^{s_1} \cdot x_2^{s_2} \cdots x_n^{s_n} = \mathsf{H}_{sk}(x)$$

♣

引理 5.17

当 $n \geq 2\log q + 2\log(1/\epsilon)$ 时，构造 5.17在 DDH 假设下满足 ϵ-smooth 性质.

♡

证明 当 $n \geq 2\log q + 2\log(1/\epsilon)$ 时，根据 Leftover Hash Lemma，则 $\mathsf{H}_{sk}(x)$ 是一个平均意义 $(n - \log q, \epsilon)$-强随机性提取器. 则有

$$\Delta((pk, x, \mathsf{H}_{sk}(x)), (pk, x, \pi)) \leq \epsilon$$

其中 $x \xleftarrow{R} X, \pi \xleftarrow{R} \Pi$. 所以 H 是一个 ϵ-smooth 哈希证明系统. 证毕! □

引理 5.18

构造 5.17满足同态性.

♡

证明 对于任意两个元素 $x = (x_1, x_2, \ldots, x_n), y = (y_1, y_2, \ldots, y_n) \in X$, 由于

$$\mathsf{H}_{sk}(x) = x_1^{s_1} \cdot x_2^{s_2} \cdots x_n^{s_n} \qquad \mathsf{H}_{sk}(y) = y_1^{s_1} \cdot y_2^{s_2} \cdots y_n^{s_n}$$

所以

$$\mathsf{H}_{sk}(x) \cdot \mathsf{H}_{sk}(y) = (x_1 \cdot y_1)^{s_1} \cdot (x_2 \cdot y_2)^{s_2} \cdots (x_n \cdot y_n)^{s_n} = \mathsf{H}_{sk}(x \cdot y)$$

从而同态性质得证! □

综上可知，构造 5.17是一个满足同态性的哈希证明系统. 结合 KDM-CPA PKE 的通用构造 5.16，可以得到一个基于 DDH 问题的 KDM-CPA 安全的 PKE 方案. 该方案也是 Boneh 等 [51] 在 2008 年美密会上提出的首个标准模型下的 KDM-CPA 安全的 PKE 方案. 由于 Wee 等通过同态哈希证明系统的高度概括，使得方案的 KDM 安全性理解起来更加值观和容易.

KDM-CPA IBE 的构造. 类似公钥加密方案，身份加密方案的 KDM 安全模型除了向攻击者提供用户密钥查询谕言机外，也会向攻击者提供一组用户密钥的 KDM 函数值的加密谕言机.

定义 5.14 (IBE 的 KDM-CPA 安全性)

令 $\mathcal{F} \subset \{f : SK^{d \leq n} \to M\}$，其中 SK 是 IBE 方案的用户密钥空间，M 是消息空间，n 是参与 KDM 函数加密的最大用户数量. 令 $|m|$ 表示消息空间 M 中每个消息的长度. 定义身份加密方案 IBE 的 KDM-CPA 敌

手 $\mathcal{A}$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa)=\left|\Pr\left[\beta=\beta':\begin{array}{l}(mpk,msk)\leftarrow\mathsf{Setup}(1^{\kappa});\\ \beta\xleftarrow{R}\{0,1\};\\ \beta'\leftarrow\mathcal{A}^{\mathcal{O}_{\mathsf{ext}},\mathcal{O}^{\beta}_{\mathsf{encrypt}}}(mpk);\end{array}\right]-\frac{1}{2}\right|.$$

令 **id** 是一个初始化为空的目标身份集合. $\mathcal{A}$ 可以自适应地向集合 **id** 中添加身份, 并可以访问以下两个谕言机:

- 密钥提取谕言机 $\mathcal{O}_{\mathsf{ext}}$: 输入身份 $id\in I$, 返回用户密钥 $sk_{id}\leftarrow\mathsf{Extract}(msk,id)$. 这里假设对于相同的身份, $\mathcal{O}_{\mathsf{ext}}$ 返回相同的用户密钥 sk_{id}.
- 加密谕言机 $\mathcal{O}^{\beta}_{\mathsf{encrypt}}$: 输入 $i\in[n]$ 和 $f\in\mathcal{F}$, 如果 $\beta=0$, 返回 $\mathsf{Encrypt}(mpk,id_i^*,f(sk_1^*,\ldots,sk_d^*))$; 如果 $\beta=1$, 返回 $\mathsf{Encrypt}(mpk,id_i^*,0^{|m|})$. 其中, $(d\leq n)$, sk_i^* 是身份 id_i^* 的密钥.

如果对于任意的 PPT 敌手 $\mathcal{A}$, 优势函数 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 是可忽略的, 则称身份加密方案 IBE 是 $\mathcal{F}$-KDM-CPA 安全的. 如果目标身份集合 id 是敌手事先声明的, 则该方案是选择身份 $\mathcal{F}$-KDM-CPA 安全的. ♣

身份哈希证明系统 [75] 是哈希证明系统的一种推广. 类似同态哈希证明系统, 陈等 [334] 指出, 如果一个身份哈希证明系统满足同态性, 则利用身份哈希证明系统可以构造一个 $\mathcal{F}$-KDM-CPA 安全的身份加密方案, 其中 $\mathcal{F}$ 是一个从哈希证明系统私钥空间到哈希值空间的仿射变换函数.

此外, 陈等 [334] 还指出利用可穿孔伪随机函数和不可区分混淆器也可以将一个 KDM-CPA 安全的 PKE 方案转化为 KDM-CPA 安全的 IBE 方案. 其基本思路是将可穿孔伪随机函数的密钥作为 IBE 方案的主私钥, 并利用可穿孔伪随机函数将用户身份 id 映射为 PKE 密钥生成算法的随机数空间上的一个随机数, 继而利用该随机数生成 PKE 方案的公钥 pk 和私钥 sk, 并将 sk 作为用户私钥 sk_{id}. 在加密算法中, 对于相同的用户身份 i, 为了生成相同的 PKE 公钥 pk 用于加密消息, 同时不泄漏可穿孔伪随机函数的私钥, 利用不可区分混淆器生成同一个公钥 pk. KDM-CPA 安全 IBE 方案的这一构造方法本质上是一种从 PKE 到 IBE 的转化方法, 具体见构造 5.18.

构造 5.18

构造所需的组件是:

- 一个公钥加密方案 $\mathsf{PKE}=(\mathsf{Setup},\mathsf{KeyGen},\mathsf{Encrypt},\mathsf{Decrypt})$.
- 一个可穿孔伪随机函数 $\mathsf{PPRF}=(\mathsf{Setup},\mathsf{KeyGen},\mathsf{Puncture},\mathsf{Eval})$.
- 一个不可区分混淆 $i\mathcal{O}$.

构造 KDM-CPA IBE 如下:

- $\mathsf{Setup}(1^{\kappa})$: 运行 $pp_1\leftarrow\mathsf{PKE.Setup}(1^{\kappa})$ 和 $pp_2\leftarrow\mathsf{PPRF.Setup}(1^{\kappa})$, 输出系统参数 $pp=(pp_1,pp_2)$.
- $\mathsf{KeyGen}(pp)$: 运行 $k\leftarrow\mathsf{PPRF.KeyGen}(pp_2)$, 并构造图 5.22 中从身份 id 到公钥 pk 的混淆程序. 将混淆电路作为系统的主公钥 mpk, 并输出主公钥 mpk 和主私钥 $msk=k$.
- $\mathsf{Extract}(msk,id)$: 以主私钥 msk 和身份 id 为输入, 计算 $r\leftarrow\mathsf{PPRF.Eval}(msk,id)$ 和 $(pk,sk)\leftarrow\mathsf{PKE.KeyGen}(pp_1;r)$. 输出用户 id 的私钥 $sk_{id}=sk$.
- $\mathsf{Encrypt}(mpk,id,m)$: 以主公钥 mpk, 身份 id 和明文 $m\in M$ 为输入, 利用混淆电路计算身份 id 相应的公钥 $pk=i\mathcal{O}(id\text{-to-}pk\ \text{hash})$. 输出密文 $c\leftarrow\mathsf{PKE.Encrypt}(pk,m)$.
- $\mathsf{Decrypt}(sk_{id},c)$: 以私钥 sk_{id} 和密文 c 为输入, 计算 $m'=\mathsf{PKE.Decrypt}(sk_{id},c)$.

♣

根据文献 [334], 则有以下结论:

定理 5.14

如果 PKE 是 $\mathcal{F}$-KDM-CPA 安全的, PPRF 是选择伪随机的, $i\mathcal{O}$ 是安全的, 则构造 5.18 中的 IBE 是选择身份 $\mathcal{F}$-KDM-CPA 安全的. ♡

id-to-pk hash

Constants: PPRF key k

Input: id

1. Compute $r \leftarrow \mathsf{PPRF.Eval}(k, id)$, $(pk, sk) \leftarrow \mathsf{PKE.KeyGen}(r)$, and output pk.

图 **5.22:** 从 id 到 pk 的转化方式

第六章 公钥加密的功能性扩展

章前概述

内容提要

- 可搜索公钥加密
- 可托管公钥加密
- 代理重加密

本章开始介绍公钥密码学的第六部分-公钥加密的功能性扩展. 6.1节介绍了可搜索公钥加密、支持消息加密的可搜索公钥加密和抗关键词猜测攻击的公钥认证可搜索加密的基本概念、性质和 (通用) 构造方法, 6.2节介绍了可托管公钥加密的基本概念、性质和两种通用构造方法, 6.3节介绍了代理重加密的基本概念、性质和基于双线性映射的构造方法.

6.1 可搜索公钥加密

寻枝寻叶必知根, 无智便乃心昏.

— 宋 · 无名氏《西江月》

网络技术的发展使得个人以及企业的数据规模迅速膨胀, 海量的数据资源受限于硬件设备而不能妥善保存. 随着云存储技术的逐渐成熟, 许多大型互联网公司开始搭建大容量的云存储服务设施, 为企业及个人的数据存储提供支持. 越来越多的用户选择将本地数据上传至云存储服务器以便减轻本地数据的存储和管理开销. 由于云存储服务器的提供商并不是一个完全可信的实体, 黑客针对云存储服务器的攻击也层出不穷, 用户在接受云服务的同时面临数据泄密的风险 [335]. 因此, 企业及个人的隐私数据不能以明文形式存储在云服务器上, 用户在数据上传之前需要对本地数据进行加密处理, 确保云端数据即使在遭受恶意攻击或不可信云存储服务器主动泄密数据的情况下仍能维护其安全性. 传统的数据加密技术能保证数据的安全特性, 然而这种技术对于云上数据的保护阻碍数据检索的高效性. 云服务器将数据提供者的密文数据存储起来, 当数据使用者检索所需信息时, 只能将所有密文从云端下载、解密之后才能进行检索. 这种方式的效率极低且容易对网络资源造成巨大的浪费, 无法满足数据使用者检索隐私数据时的效率需求.

为了在保证隐私数据安全的同时解决数据检索的效率问题, 可搜索加密 (searchable encryption, SE) 这一概念应运而生. 用户在本地提取明文数据中的关键词信息构造关键词密文索引, 云服务器存储这些密文索引, 具备检索能力的用户再根据其所要检索的关键词信息生成检索令牌发送至云服务器, 云服务器通过其检索匹配算法对其所寻求的密文信息进行搜索并返回结果. 如此, 便可在不需要解密云端密文的情况下完成对需求数据的高效率检索. 根据加密密钥是否可公开, 可搜索加密可以划分为可搜索对称加密 [213] 和可搜索公钥加密 [185].

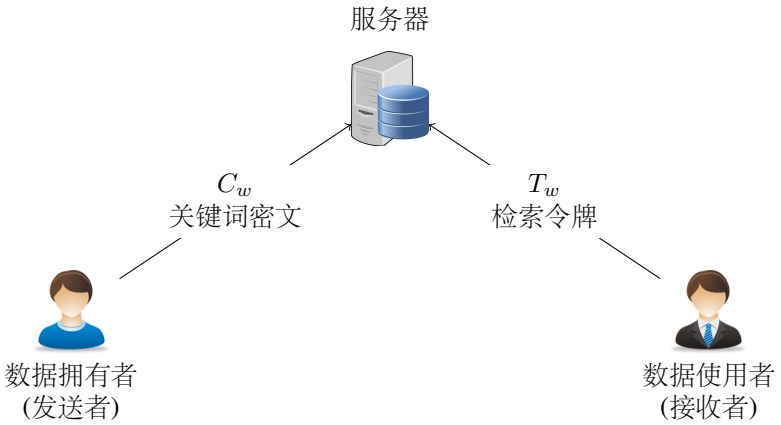


图 6.1: 可搜索公钥加密的应用模式

可搜索公钥加密技术的产生可以追溯到最初的加密邮件路由问题. 如图 6.1所示, Email 用户可以将邮件中包含的关键词信息提取出来, 并使用邮件接收者的公钥将其加密为关键词密文索引, 并与邮件内容的密文一起上传到服务器. 而邮件接收者可以利用自己的私钥生成一个关键词 w 的检索令牌 T_w 发送给 Email 服务器, 使得服务器能够返回所有包含关键词 w 的邮件, 而服务器不会得到密文关键词的其他信息. 由于该系统利用邮件接收者的公钥加密关键词, Boneh 等 [185] 将其称为可搜索公钥加密 (public-key encryption with keyword search, PEKS).

在索引建立方面, 一般采用比较流行的倒排索引结构, 每个关键词对应了多个包含该关键词的文档. 在传统倒排索引结构基础上, 只需要利用可搜索公钥加密算法对关键词列表进行加密, 而文档内容可采用其他方式进行加密保护, 如图 6.2所示. PEKS 方案的设计初衷是保护关键词索引信息的隐私, 存储服务器或恶意敌手无法从密态关键词索引中获取关键词的相关信息, 同时能够保证合法用户从密态关键词索引中检索出指定的关键词, 从而利用倒排索引结构获取所有包含该关键词的文档.

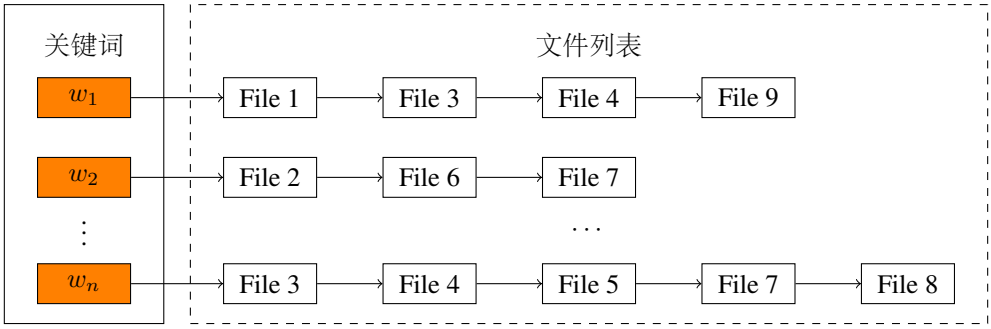


图 6.2: 倒排索引结构

除了对密态关键词索引进行检索外, 也有工作如 [216, 217, 219] 将 PKE 和 PEKS 结合不仅能够实现密态关键词索引的检索, 而且可以正确恢复出原始的关键词, 这类方案也称为 PKE+PEKS 方案. 标准的 PEKS 安全模型和 PKE+PEKS 安全模型仅考虑对关键词密文的安全性保护, 而无法保护检索令牌中的关键词的安全性, 使其容易遭受关键词猜测攻击 [336, 337]. 特别地, 在支持关键词解密操作的 PKE-PEKS 方案中, 部分方案的检索令牌甚至包含明文形式的检索关键词, 如 [338]. 近年来, 许多学者在 PEKS 方案的基础上研究如何保护检索令牌隐私和抵抗关键词猜测攻击的方法, 如 [339, 340, 341]. 其中一种较为流行的方法是公钥认证可搜索加密 (public-key authenticated encryption with keyword search, PAEKS). PAEKS 是在标准的 PEKS 基础上, 通过引入关键词加密者的私钥, 以防止非法用户生成合法密文的目的, 从而避免检索令牌遭受关键词猜测攻击. 本节将围绕 PEKS, PKE+PEKS 和 PAEKS 等概念展开介绍和讨论.

6.1.1 可搜索公钥加密的定义与安全性

定义 6.1 (可搜索公钥加密)

一个可搜索公钥加密方案 PEKS 由以下 5 个 PPT 算法组成:

- $\mathsf{Setup}(1^\kappa)$: 系统参数生成算法以安全参数 1^κ 为输入, 输出系统公开参数 pp, 其中 pp 包含了用户的公钥空间 PK、私钥空间 SK、关键词空间 W、密文空间 C 和检索令牌空间 T 的描述. 类似公钥加密方案, 该算法由可信第三方生成并公开, 系统中的所有用户共享, 所有算法均将 pp 作为输入的一部分.
- $\mathsf{KeyGen}(pp)$: 密钥生成算法以公开参数 pp 为输入, 输出一对公/私钥 (pk, sk), 其中 pk 公开, sk 保密.
- $\mathsf{Encrypt}(pk, w)$: 加密算法以公钥 $pk \in PK$ 和关键词 $w \in W$ 为输入, 输出关键词 w 的一个可搜索密文 $c_w \in C$.
- $\mathsf{TokenGen}(sk, w)$: 检索令牌生成算法以私钥 $sk \in SK$ 和关键词 $w \in W$ 为输入, 输出关键词 w 的一个检索令牌 t_w.
- $\mathsf{Test}(t_{w'}, c_w)$: 检索算法以关键词 w' 的检索令牌 $t_{w'}$ 和关键词 w 的密文 c_w 为输入, 如果 $w = w'$, 则输出 1; 否则, 输出 0.

♣

正确性. 该性质保证了 PEKS 密文的可检索功能, 即利用私钥可以生成关键词的检索令牌并检索出所有包含匹配关键词的密文. 正式地, 对于任意关键词 $w \in W$, 有:

$$\Pr[\mathsf{Test}(t_w, \mathsf{Encrypt}(pk, w)) = 1] \geq 1 - \mathsf{negl}(\kappa) \tag{6.1}$$

在公式 6.1中, Test 算法输出 1 的概率建立在系统参数 $pp \leftarrow \mathsf{Setup}(1^\kappa)$, 公/私钥对 $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$, 检索令牌 $t_w \leftarrow \mathsf{TokenGen}(sk, w)$ 和关键词密文 $c_w \leftarrow \mathsf{Encrypt}(pk, w)$ 的随机带上. 如果上述概率严格等于 1, 则称 PEKS 方案满足完美正确性.

一致性. 该性质保证了 PEKS 密文的检索错误率, 即检索令牌仅能与所有包含匹配关键词的密文通过检索算法.

也就是说, 对于任意关键词 $w, w' \in W$ 且 $w \neq w'$, 有:

$$\Pr[\mathsf{Test}(t_{w'}, \mathsf{Encrypt}(pk, w)) = 1] \leq \mathsf{negl}(\kappa) \tag{6.2}$$

与 PKE 方案不同, PEKS 方案不仅需要满足正确性, 还要满足一致性. Abdalla 等 [214] 研究了 PEKS 方案的完美一致性, 统计一致性和计算一致性. 一般地, 仅考虑计算一致性即可. 许多 PEKS 方案满足正确性的同时也满足一致性, 而忽略对 PEKS 方案一致性的分析. 下面介绍一致性的两种形式化定义: 弱一致性和强一致性.

弱一致性. 定义一个 PEKS 方案敌手 $\mathcal{A}$ 的弱一致性优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[\mathsf{Test}(t_{w'}, c_w) = 1 : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ (w, w') \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{tokengen}}}(pp, pk); \\ c_w \leftarrow \mathsf{Encrypt}(pk, w); \\ t_{w'} \leftarrow \mathsf{TokenGen}(sk, w'); \end{array}\right]$$

在上述定义中, $\mathcal{O}_{\mathsf{tokengen}}$ 表示检索令牌谕言机, 其在接收到关键词 w 的询问后, 输出 $\mathsf{TokenGen}(sk, w)$. 如果任意的 PPT 敌手 $\mathcal{A}$ 在上述定义中的优势函数是可忽略的, 则称 PEKS 方案是弱一致的.

强一致性. 定义一个 PEKS 方案敌手 $\mathcal{A}$ 的强一致性优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[\mathsf{Test}(t_{w'}, c_w) = 1 : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ (w, w', c_w) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{tokengen}}}(pp, pk); \\ t_{w'} \leftarrow \mathsf{TokenGen}(sk, w'); \end{array}\right]$$

在上述定义中, $\mathcal{O}_{\mathsf{tokengen}}$ 表示检索令牌谕言机. 如果任意的 PPT 敌手 $\mathcal{A}$ 在上述定义中的优势函数是可忽略的, 则称 PEKS 方案是强一致的.

注记 6.1

弱一致性和强一致性的区别主要在于匹配检索的密文是通过合法途径生成的还是敌手设法伪造的.

可搜索公钥加密的语义安全性是为了防止敌手 (恶意存储服务器) 从关键词密文 $\mathsf{PEKS}(pk, w)$ 中得到 w 的任何额外信息, 除非敌手获取了 w 的检索令牌. 此外, 敌手可以自适应地获取其它关键词 w' 的检索令牌 $t_{w'}$. 下面通过两个关键词密文的不可区分性来描述可搜索加密的语义安全性, 即自适应选择关键词攻击下的密文不可区分安全性, 简称 CI-CKA 安全性.

CI-CKA 安全性. 定义一个 PEKS 方案敌手 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left|\Pr\left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ (w_0, w_1, state) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{tokengen}}}(pp, pk); \\ \beta \xleftarrow{R} \{0, 1\}; \\ c^* \leftarrow \mathsf{Encrypt}(pk, w_\beta); \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{tokengen}}}(pp, pk, state, c^*); \end{array}\right] - \frac{1}{2}\right|$$

在上述定义中, $\mathcal{O}_{\mathsf{tokengen}}$ 表示检索令牌谕言机, 其在接收到关键词 w 的询问后, 输出 $\mathsf{TokenGen}(sk, w)$, 但是要求 $w \notin \{w_0, w_1\}$. 如果任意的 PPT 敌手 $\mathcal{A}$ 在上述定义中的优势函数是可忽略的, 则称 PEKS 方案是 CI-CKA 安全的.

注记 6.2

如果没有检索令牌询问, PEKS 方案的 CI-CKA 安全模型和 PKE 方案的 IND-CPA 安全模型是完全一样的.

PEKS 与 IBE 之间的关系. PEKS 和 IBE 两种密码原语之间有着天然的联系, 可以相互转化. 图 6.3给出了二者参数空间以及算法之间的匹配关系. Boneh 等指出构造一个安全的 PEKS 方案比构造一个 IBE 方案更困难, 这是因为任意一个 PEKS 方案蕴含了一个 IBE 方案, 见构造 6.1. 然而, 反之未必成立.

参数对应关系		算法对应关系	
PEKS.PK	IBE.MPK	PEKS.Setup	IBE.Setup
PEKS.SK	IBE.MSK	PEKS.KeyGen	IBE.KeyGen
PEKS.T	IBE.sk_{id}	PEKS.TokenGen	IBE.Extract
PEKS.W	IBE.ID	PEKS.Encrypt	IBE.Encrypt
PEKS.C	IBE.C	PEKS.Test	IBE.Decrypt

图 6.3: PEKS 与 IBE 之间的关系

构造 6.1 (从 PEKS 到 IBE 的转化)

假设 PEKS = (Setup, KeyGen, Encrypt, TokenGen, Test) 是一个 PEKS 方案, 下面构造一个消息空间为 $\{0,1\}$ 的身份加密方案 IBE = (Setup, KeyGen, Extract, Encrypt, Decrypt).

- Setup(1^κ): 运行 $pp \leftarrow$ PEKS.Setup(1^κ), 将 PEKS 的系统参数 pp 作为 IBE 的系统参数.
- KeyGen(pp): 运行 $(pk, sk) \leftarrow$ PEKS.KeyGen(pp), 将 PEKS 的用户公钥 pk 和私钥 sk 分别作为 IBE 的主公钥 mpk 和主私钥 msk.
- Extract(msk, id): 对于任意用户身份 $id \in \{0,1\}^*$, 运行 $t_b \leftarrow$ PEKS.TokenGen$(sk, id||b)$ 两次, 其中 $b = 0, 1$. 将检索令牌 t_0 和 t_1 作为用户 id 的私钥, 即 $sk_{id} = (t_0, t_1)$.
- Encrypt(mpk, id, m): 对于消息 $m \in \{0,1\}$, 运行 $c \leftarrow$ PEKS.Encrypt$(pk, id||m)$. 将 PEKS 的密文 c 作为 IBE 密文.
- Decrypt(sk_{id}, c): 输入用户私钥 $sk_{id} = (t_0, t_1)$ 和密文 c, 如果 PEKS.Test$(t_0, c) = 1$, 则输出 0; 如果 PEKS.Test$(t_1, c) = 1$, 则输出 1.

♣

构造 6.1的安全性由下面的引理保证:

引理 6.1

如果 PEKS 满足 CI-CKA 安全性, 则构造 6.1中的 IBE 是 IND-CCA 安全的.

♡

笔记 在不考虑安全性的情况下, 利用一个 IBE 方案按照图 6.3所示的对应方式可以构造一个满足正确性 (不一定安全) 的 PEKS 方案. 将一个固定消息空间 $0^{|M|}$ 的 IBE 密文 IBE.Encrypt$(mpk, w, 0^{|M|})$ 作为关键词 w 的 PEKS 密文. 检索匹配算法只需要利用 w 对应的标识密钥解密该密文, 如果解密出的结果与固定消息 $0^{|M|}$ 一致, 则检索成功. 然而, IBE 的加密算法并不要求身份标识是保密的, 也就是说 IBE 密文可能会泄漏身份的信息. 此外, IBE 解密算法不一定满足一致性, 利用不同身份标识的用户私钥可能解密出正确的结果. 2005 年, Abdalla 等 [214] 指出, 解决这两个问题可以选择一个匿名的身份加密方案并将固定消息 $0^{|M|}$ 替换为随机消息 R, 将 IBE.Encrypt(mpk, w, R) 和 R 同时作为 PEKS 的密文. 在匿名的身份加密方案中, 由于密文不会泄漏身份的信息, 故 PEKS 密文不会泄漏关键词的信息. 又由于加密的是随机消息, 一个不匹配的检索令牌 (用户的标识密钥) 解密出的消息与 R 一致的可能性是可以忽略的.

6.1.2 可搜索公钥加密的构造

PEKS 方案的构造. 下面介绍 Boneh 等在 2004 年提出的第一个 PEKS 方案, 记作 BDOP-PEKS 方案.

构造 6.2 (BDOP-PEKS 方案)

- Setup(1^κ): 运行 GenBLGroup(1^κ) 生成一个 Type-I 双线性映射 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$. 选择两个密码学哈希函数 $\mathsf{H}_1 : \{0,1\}^* \to \mathbb{G}$ 和 $\mathsf{H}_2 : \mathbb{G}_T \to \{0,1\}^{\log q}$. 输出系统参数 $pp = (\mathbb{G}, \mathbb{G}_T, q, g, e, \mathsf{H}_1, \mathsf{H}_2)$.
- KeyGen(pp): 随机选择 $\alpha \xleftarrow{R} \mathbb{Z}_q$, 计算 $h = g^\alpha$, 输出公钥 $pk = h$ 和私钥 $sk = \alpha$.
- Encrypt(pk, w): 对于任意关键词 $w \in \{0,1\}^*$, 随机选择 $r \xleftarrow{R} \mathbb{Z}_q$, 计算 $t = e(\mathsf{H}_1(w), h^r)$, 输出密文 $C = (g^r, \mathsf{H}_2(t))$.

- $\mathsf{TokenGen}(sk, w)$: 对于任意关键词 $w \in \{0,1\}^*$, 输出检索令牌 $t_w = \mathsf{H}_1(w)^\alpha$.
- $\mathsf{Test}(t_w, c)$: 对于密文 $c = (A, B)$ 和检索令牌 t_w, 判断等式 $\mathsf{H}_2(e(t_w, A)) = B$ 是否成立. 如果成立, 则输出 1, 否则输出 0.

笔记 BDOP-PEKS 方案是在 Boneh 和 Franklin 的身份加密方案, 记作 BF-IBE 方案 [267], 基础上设计的. 由于 BF-IBE 方案满足身份匿名性, 利用前面讨论的从匿名 IBE 到 PEKS 的转化思路, 设计出 BDOP-PEKS 方案是比较自然的.

BDOP-PEKS 方案的安全性基于 BDH 问题的困难性. 双线性映射上的 BDH 问题描述如下: 给定一个双线性映射 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$, 输入 $g, g^\alpha, g^\beta, g^\gamma \in \mathbb{G}$, 计算 $e(g,g)^{\alpha\beta\gamma}$. BDOP-PEKS 方案的安全性由下面的定理 6.1保证.

定理 6.1

如果 BDH 假设相对于 GenBLGroup 成立, 则在随机谕言机模型下 BDOP-PEKS 方案是 CI-CKA 安全的.

定理 6.1可通过安全归约思想来证明. 模拟算法 $\mathcal{B}$ 可以将一个待解决的 BDH 问题实例嵌入到模拟的 BDOP-PEKS 方案中, 并借助 H_1 和 H_2 的随机谕言机性质, 可以控制 H_1 和 H_2 的输出形式以回答敌手的检索令牌询问. 下面介绍归约的具体过程.

证明 令 $\mathcal{A}$ 是一个以 ϵ 优势攻击 BDOP-PEKS 方案 CI-CKA 安全性的敌手, $g, u_1 = g^\alpha, u_2 = g^\beta, u_3 = g^\gamma \in \mathbb{G}$ 是双线性映射 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$ 上的一个 BDH 问题实例. 若 $\mathcal{A}$ 成功的概率不可忽略, 归约证明的目标是构造一个算法 $\mathcal{B}$, 以 BDH 问题实例为输入, 借助敌手 $\mathcal{A}$ 的能力以不可忽略的概率解决该 BDH 问题实例, 从而推出矛盾. 假设 $\mathcal{A}$ 最多询问 q_{H_2} 次哈希函数, q_T 次检索令牌. $\mathcal{B}$ 按如下方式模拟 $\mathcal{A}$ 在游戏中的视图环境, 即原始游戏中挑战者的行为.

- 初始化: $\mathcal{B}$ 根据双线性映射的参数 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$ 选择两个密码学哈希函数 $\mathsf{H}_1 : \{0,1\}^* \to \mathbb{G}$ 和 $\mathsf{H}_2 : \mathbb{G}_T \to \{0,1\}^{\log q}$, 并令系统参数为 $pp = (\mathbb{G}, \mathbb{G}_T, q, g, e, \mathsf{H}_1, \mathsf{H}_2)$, 用户的公钥为 $pk = u_1$. $\mathcal{B}$ 将系统参数 pp 和公钥 pk 发送给 $\mathcal{A}$. 显而易见, 这里隐含地选择了用户私钥为挑战 BDH 问题实例中的 α. 尽管 $\mathcal{B}$ 不知道 α 的具体值 (也不能知道该秘密, 否则就没法进行归约证明了), 但是 α 是随机选择的, 所以 $\mathcal{B}$ 模拟的系统参数和用户公钥与实际游戏环境是一致的.
- 阶段 1 询问: 在真实游戏中, 挑战者只需要回答敌手的检索令牌询问和挑战询问, 而任意元素的哈希值计算是公开的. 为了使算法 $\mathcal{B}$ 能够模拟 $\mathcal{A}$ 的视图环境, 需要将哈希函数 H_1 和 H_2 看作随机谕言机, 即在随机谕言机模型中模拟敌手的各类谕言机查询结果. 具体如下:
  - H_1 和 H_2 询问: 在任何时候, 敌手 $\mathcal{A}$ 都可以询问随机谕言机 H_1 或 H_2. 对于 H_1 哈希询问, $\mathcal{B}$ 维护一个形如 $\langle w_j, h_j, a_j, c_j \rangle$ 且初始化为空的 H_1-列表. 当 $\mathcal{A}$ 询问 $w_i \in \{0,1\}^*$ 的 H_1 哈希值时, 算法 $\mathcal{B}$ 按如下方式进行回答:
    1. 如果 w_i 已经在 H_1-列表元素 $\langle w_j, h_j, a_j, c_j \rangle$ 中, 则算法 $\mathcal{B}$ 返回 $\mathsf{H}_1(w_i) = h_i \in \mathbb{G}$.
    2. 否则, $\mathcal{B}$ 随机选择一比特 $c_i \in \{0,1\}$, 使得 $\Pr[c_i = 0] = 1/(q_T + 1)$.
    3. $\mathcal{B}$ 随机选择 $a_i \in \mathbb{Z}_q$, 并计算
    $$h_i = \begin{cases} u_2 \cdot g^{a_i}, & \text{如果 } c_i = 0 \\ g^{a_i}, & \text{如果 } c_i = 1 \end{cases}$$
    4. $\mathcal{B}$ 将元素 $\langle w_j, h_j, a_j, c_j \rangle$ 添加到 H_1-列表中并将哈希值 $\mathsf{H}_1(w_i) = h_i$ 返回给 $\mathcal{A}$. 显而易见, 不论随机比特 c_i 取值如何, 哈希值 h_i 都是群 $\mathbb{G}$ 中的一个随机元素且与 $\mathcal{A}$ 当前的视图独立无关. 这与 H_1 是一个随机谕言机的假设一致.

    类似地, $\mathcal{A}$ 可以在任何时候询问 H_2 的哈希值. 此时, $\mathcal{B}$ 维护一个形如 $\langle t_i, V_i \rangle$ 且初始化为空的 H_2-列表. 当 $\mathcal{A}$ 询问 t_i 的 H_2 哈希值时, 如果 t_i 在 H_2-列表元素 $\langle t_i, V_i \rangle$ 中, 则 $\mathcal{B}$ 返回 V_i; 否则, $\mathcal{B}$ 随机选择 $V_i \in \{0,1\}^{\log q}$, 将 $\langle t_i, V_i \rangle$ 添加到 H_2-列表中, 并将哈希值 $\mathsf{H}_2(t_i) = V_i$ 返回给 $\mathcal{A}$.
  - 检索令牌询问: 当 $\mathcal{A}$ 询问关键词 w_i 的检索令牌时, $\mathcal{B}$ 按下面的方式进行回答:
    1. $\mathcal{B}$ 通过 H_1 哈希询问方式获取 w_i 的哈希值 $h_i \in \mathbb{G}$, 即 $\mathsf{H}_1(w_i) = h_i$. 令 $\langle w_j, h_j, a_j, c_j \rangle$ 是 H_1-列表

中的相应元素. 如果 $c_i = 0$, 则 $\mathcal{B}$ 模拟失败并终止游戏.

2. 否则, $c_i = 1$ 且 $h_i = g^{a_i} \in \mathbb{G}$. $\mathcal{B}$ 计算 $T_i = u_1^{a_i}$. 注意到 $\mathsf{H}_1(w_i) = g^{a_i}$ 且 $u_1 = g^{\alpha}$, 所以 $T_i = \mathsf{H}_1(w_i)^{\alpha}$ 是 w_i 的一个正确的检索令牌. $\mathcal{B}$ 将 T_i 发送给 $\mathcal{A}$.

- 挑战: 当阶段 1 询问结束时, $\mathcal{A}$ 选择两个挑战关键词 $w_0, w_1 \in \{0,1\}^*$ 发送给 $\mathcal{B}$. 算法 $\mathcal{B}$ 按下面的方式生成挑战 PEKS 密文:

1. $\mathcal{B}$ 通过两次 H_1 哈希询问获取 w_0 和 w_1 的哈希值 $h_0, h_1 \in \mathbb{G}$ 且满足 $\mathsf{H}_1(w_0) = h_0$ 和 $\mathsf{H}_1(w_1) = h_1$. 假设 $\langle w_0, h_0, a_0, c_0\rangle$ 和 $\langle w_1, h_1, a_1, c_1\rangle$ 分别是相应的 H_1-列表中的元素. 如果 $c_0 = 1$ 且 $c_1 = 1$, 则 $\mathcal{B}$ 模拟失败并终止游戏.
2. 否则, c_0 和 c_1 中至少有一个等于 0. $\mathcal{B}$ 随机选择 $b \in \{0,1\}$ 使得 $c_b = 0$.
3. 算法 $\mathcal{B}$ 随机选择 $J \in \{0,1\}^{\log q}$ 并将 $C^* = (u_3, J)$ 作为挑战密文返回给 $\mathcal{A}$.

值得注意的是, 挑战密文隐含地定义了 $\mathsf{H}_2(e(\mathsf{H}_1(w_b), u_1^{\gamma})) = J$. 由此可知

$$J = \mathsf{H}_2(e(\mathsf{H}_1(w_b), u_1^{\gamma})) = \mathsf{H}_2(e(u_2 g^{a_b}, g^{\alpha\gamma})) = \mathsf{H}_2(e(g,g)^{\alpha\gamma(\beta+a_b)})$$

是 w_b 的一个合法密文.

- 阶段 2 询问: $\mathcal{A}$ 可以继续进行哈希询问和关键词的检索令牌询问, 但是不允许询问挑战关键词的检索令牌.
- 输出: 最终, $\mathcal{A}$ 将输出一猜测比特 $b' \in \{0,1\}$. $\mathcal{B}$ 从 H_2-列表中随机选择一个元素 $\langle t, v\rangle$, 计算 $T = t/e(u_1, u_3)^{a_b}$ 作为 BDH 问题解 $e(g,g)^{\alpha\beta\gamma}$ 的一个猜测结果, 其中 a_b 是挑战阶段使用的元素. 如果 $\mathcal{A}$ 询问过 H_2 的哈希值 $\mathsf{H}_2(e(\mathsf{H}_1(w_0), u_1^{\gamma}))$ 或 $\mathsf{H}_2(e(\mathsf{H}_1(w_1), u_1^{\gamma}))$, 那么, H_2-列表以 1/2 的概率包含一个元素 $\langle t, v\rangle$, 其中 $t = \mathsf{H}_2(e(\mathsf{H}_1(w_b), u_1^{\gamma})) = \mathsf{H}_2(e(g,g)^{\alpha\gamma(\beta+a_b)})$. 因此, $T = t/e(u_1, u_3)^{a_b} = e(g,g)^{\alpha\beta\gamma}$.

注记 6.3

通过模拟检索令牌的过程可以看出起初设计两种不同方式回答 H_1 查询的目的. 模拟者希望敌手查询检索令牌的关键词 w_i 都是按照 $c_i = 1$ 的方式计算的, 这样模拟者就知道了 $\mathsf{H}_1(w_i)$ 关于 g 的离散对数 a_i, 从而可以在不知道 α 的情况下利用 Diffie-Hellman 密钥交换原理计算出 w_i 的检索令牌 $t_{w_i} = \mathsf{H}_1(w_i)^{\alpha} = (g^{\alpha})^{a_i}$. 然而, 当 $c_i = 0$ 时, 模拟者就无法回答敌手的检索令牌询问, 此时只能终止游戏. 那么, 能否始终按照 $c_i = 1$ 的方式提供哈希查询呢? 答案是否定的, 为了将 BDH 问题嵌入到挑战密文中, 模拟者又希望对挑战关键词 w_b 按照 $c_i = 0$ 的方式回答哈希查询.

至此, 完成了算法 $\mathcal{B}$ 的描述. 通过一系列的 "操控", $\mathcal{B}$ 已成功地将 BDH 问题实例的解 $e(g,g)^{\alpha\beta\gamma}$ 嵌入到挑战密文的元素 J 中. 如果敌手 $\mathcal{A}$ 查询了 J 对应的 H_2 谕言机输入的元素 $e(g,g)^{\alpha\gamma(\beta+a_b)}$, 那么, $\mathcal{B}$ 可以从中恢复出 $e(g,g)^{\alpha\beta\gamma}$, 从而攻破 BDH 问题的实例. 这里需要解决两个问题: 一是模拟者的这些 "操控" 对于敌手而言, 必须和真实攻击环境一样 (不可区分), 这可通过引理 6.2和引理 6.3保证; 二是敌手会查询 $\mathsf{H}_2(e(\mathsf{H}_1(w_b), u_1^{\gamma}))$ 的 H_2 哈希询问, 这可以通过引理 6.4来保证.

下面主要是分析 $\mathcal{B}$ 正确输出 BDH 问题实例解 $e(g,g)^{\alpha\beta\gamma}$ 的概率 ϵ'. 首先分析 $\mathcal{B}$ 在模拟游戏中不终止的概率. 定义以下两个事件:

- E_1: 表示事件 $\mathcal{B}$ 在回答 $\mathcal{A}$ 的检索令牌询问时不终止游戏.
- E_2: 表示事件 $\mathcal{B}$ 在挑战阶段不终止游戏.

上述两个事件的概率下界由下面的引理保证.

引理 6.2

$\mathcal{B}$ 在回答 $\mathcal{A}$ 的所有检索令牌查询结果时不终止游戏的概率至少为 $1/e$, 即 $\Pr[E_1] \geq 1/e$.

证明 假设 w_i 是 $\mathcal{A}$ 的第 i 次询问检索令牌的关键词. 在 i 次询问中, $\mathcal{B}$ 终止游戏的条件是 w_i 相应的 H_1-列表元素 $\langle w_i, h_i, a_i, c_i\rangle$ 中, $c_i = 0$. 尽管哈希值 $\mathsf{H}_1(w_i)$ 的生成方式与 c_i 有关, 但是 $H_1(w_i)$ 的分布与 $c_i = 0$ 还是 $c_i = 1$ 无关. 根据 c_i 的分布, 可知 $\mathcal{B}$ 在回答本次询问过程中终止游戏的概率最多为 $\Pr[c_i = 0] = 1/(q_T + 1)$. 由于 $\mathcal{A}$ 进行检索令牌查询的次数最多为 q_T, 所以 $\mathcal{B}$ 在所有检索令牌询问中都不终止游戏的概率至少为 $(1 - 1/(q_T + 1))^{q_T} \geq 1/e$.

引理 6.2证毕! □

引理 6.3

$\mathcal{B}$ 在挑战密文生成阶段不终止游戏的概率至少为 $1/q_T$, 即 $\Pr[E_2] \geq \frac{1}{q_T}$. ♡

证明 在挑战阶段, $\mathcal{B}$ 终止游戏的条件是挑战关键词 w_0 和 w_1 相应的 H_1-列表元素 $\langle w_0, h_0, a_0, c_0\rangle$ 和 $\langle w_1, h_1, a_1, c_1\rangle$ 中, $c_0 = c_1 = 1$. 由于 $\mathcal{A}$ 不允许询问 w_0 和 w_1 的检索令牌, 所以 c_0 和 c_1 的值独立于 $\mathcal{A}$ 的当前视图, 且 c_0 和 c_1 的取值是相互独立的. 根据 c_i 的分布, 可知 $\Pr[c_0 = 1] = \Pr[c_1 = 1] = 1 - 1/(q_T + 1)$, 由此可知 $\Pr[c_0 = c_1 = 1] = (1 - 1/(q_T + 1))^2 \leq 1 - 1/q_T$. 所以 $\mathcal{B}$ 在挑战密文生成阶段不终止游戏的概率至少为 $1/q_T$.

引理 6.3证毕! □

由于 $\mathcal{A}$ 是不允许询问挑战关键词 w_0 和 w_1 的检索令牌, 所以两个事件 E_1 和 E_2 是相互独立的. 因此, $\Pr[E_1 \wedge E_2] \geq 1/(eq_T)$.

最后, 分析 $\mathcal{A}$ 询问哈希值 $\mathsf{H}_2(e(\mathsf{H}_1(w_b), u_1^\gamma))$ 的概率下界, 由以下引理保证:

引理 6.4

假设在真实游戏中, 给定 $\mathcal{A}$ 系统参数 pp, 公钥 $pk = u_1$. 当询问挑战关键词 w_0 和 w_1 的密文时, 返回给 $\mathcal{A}$ 的结果是 $C^* = (u_3 = g^\gamma, J)$. 则 $\mathcal{A}$ 在真实游戏中询问 H_2 的哈希值 $\mathsf{H}_2(e(\mathsf{H}_1(w_0), u_1^\gamma))$ 或 $\mathsf{H}_2(e(\mathsf{H}_1(w_1), u_1^\gamma))$ 的概率至少为 2ϵ. ♡

证明 令 E_3 表示事件 "敌手 $\mathcal{A}$ 在真实游戏中询问了哈希值 $\mathsf{H}_2(e(\mathsf{H}_1(w_0), u_1^\gamma))$ 或 $\mathsf{H}_2(e(\mathsf{H}_1(w_1), u_1^\gamma))$". 显然, 若事件 E_3 未发生, 在随机谕言机模型下, 挑战密文中的元素 J 完全独立于 $\mathcal{A}$ 的当前视图, 从而挑战阶段 $b \in \{0, 1\}$ 的取值与 $\mathcal{A}$ 的视图独立. 所以, $\Pr[b = b'|\overline{E_3}] = 1/2$. 根据假设, 敌手 $\mathcal{A}$ 在真实游戏中成功的优势至少为 $|\Pr[b = b'] - 1/2| \geq \epsilon$. 又由于

$$\begin{aligned}
\Pr[b = b'] &= \Pr[b = b'|E_3]\Pr[E_3] + \Pr[b = b'|\overline{E_3}]\Pr[\overline{E_3}] \\
&\leq \Pr[E_3] + \Pr[b = b'|\overline{E_3}]\Pr[\overline{E_3}] \\
&= \Pr[E_3] + \frac{1}{2}\Pr[\overline{E_3}] \\
&= \frac{1}{2} + \frac{1}{2}\Pr[E_3] \\
\Pr[b = b'] &= \Pr[b = b'|E_3]\Pr[E_3] + \Pr[b = b'|\overline{E_3}]\Pr[\overline{E_3}] \\
&\geq \Pr[b = b'|\overline{E_3}]\Pr[\overline{E_3}] \\
&= \frac{1}{2}\Pr[\overline{E_3}] \\
&= \frac{1}{2} - \frac{1}{2}\Pr[E_3]
\end{aligned}$$

所以 $\epsilon \leq |\Pr[b = b'] - 1/2| \leq \frac{1}{2}\Pr[E_3]$. 由此可得 $\Pr[E_3] \geq 2\epsilon$.

引理 6.4证毕! □

若 $\mathcal{B}$ 不终止游戏, 根据算法 $\mathcal{B}$ 的描述, 则 $\mathcal{B}$ 模拟的游戏环境与真实游戏环境是完全一样的. 根据引理 6.4, $\mathcal{A}$ 询问哈希值 $\mathsf{H}_2(e(\mathsf{H}_1(w_0), u_1^\gamma))$ 或 $\mathsf{H}_2(e(\mathsf{H}_1(w_1), u_1^\gamma))$ 的概率至少为 2ϵ. 由于 b 是独立于 w_0 和 w_1 随机选取的, 所以 $\mathcal{A}$ 询问哈希值 $\mathsf{H}_2(e(\mathsf{H}_1(w_b), u_1^\gamma))$ 的概率至少为 ϵ. 因此, 以至少 ϵ 的概率, H_2-列表中存在形如 $\langle e(\mathsf{H}_1(w_b), u_1^\gamma), \cdot\rangle$ 的元素. 如果 $\mathcal{B}$ 不终止游戏, 则 $\mathcal{B}$ 正确选取到元素 $\langle e(\mathsf{H}_1(w_b), u_1^\gamma), \cdot\rangle$ 的概率至少为 $\epsilon/q_{\mathsf{H}_2}$. 结合 $\mathcal{B}$ 不终止游戏的概率至少为 $1/(eq_T)$, 所以 $\mathcal{B}$ 成功解决 BDH 问题的概率至少为 $\epsilon/(eq_T q_{\mathsf{H}_2})$.

定理 6.1证毕! □

PKE-PEKS 方案的构造. PKE-PEKS 方案将公钥加密和可搜索公钥加密相结合, 目的是为了解决了 PEKS 方案缺少对消息加密的不足之处. 然而, 许多工作刻画的 PKE-PEKS 安全模型并不太完善. 例如, 文献 [216, 217] 中的安全模型仅考虑消息的 IND-CCA 安全性, 而关键词的安全性仅停留在语义安全性, 并不支持对关键词密文的匹配检索查询. 此外, 设计一个 PKE-PEKS 方案并不是那么容易. 困难之一是 CCA 安全性要求密文不能具有任何的可

延展性, 简单地将一个 CCA 安全的 PKE 方案和一个 PEKS 方案组合并不能达到消息的 CCA 安全性, 一般还需要支持辅助标签输入等特殊结构. 这种直接组合的效率也不高, 需要保存 PKE 和 PEKS 两个私钥. 下面介绍一种比较完备的 PKE-PEKS 安全模型及其通用构造方法.

定义 6.2 (PKE-PEKS)

一个 PKE-PEKS 方案包含以下 6 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 系统参数生成算法以安全参数 1^κ 为输入, 输出系统公开参数 pp. 公开参数定义了消息空间 M、关键词空间 W、密文空间 C 和检索令牌空间 T. 该算法由可信第三方生成并公开, 系统中的所有用户共享, 所有算法均将 pp 作为输入的一部分.
- $\mathsf{KeyGen}(pp)$: 密钥生成算法以公开参数 pp 为输入, 输出一对公钥/私钥 (pk, sk).
- $\mathsf{Encrypt}(pk, m, w)$: 加密算法以公钥 pk, 消息 $m \in M$ 和关键词 $w \in W$ 为输入, 输出 PKE-PEKS 密文 c.
- $\mathsf{Decrypt}(sk, C)$: 解密算法以私钥 sk 和 PKE-PEKS 密文 $c \in C$ 为输入, 输出明文 $m \in M$ 或者一个特殊符号 $\perp$ 表示 c 是一个无效密文.
- $\mathsf{TokenGen}(sk, w)$: 检索令牌生成算法以私钥 sk 和关键词 $w \in W$ 为输入, 输出关键词 w 的一个检索令牌 t_w.
- $\mathsf{Test}(t_{w'}, c)$: 检索算法以关键词 w' 的检索令牌 $t_{w'}$ 和关键词 w 的密文 c 为输入, 如果 $w = w'$, 则输出 1; 否则, 输出 0.

正确性. 对于任意系统参数 $pp \leftarrow \mathsf{Setup}(1^\kappa)$, 任意公/私钥对 $(pk, sk) \leftarrow \mathsf{KeyGen}(pp)$, 任意消息 $m \in M$, 任意关键词 $w \in W$ 和任意检索令牌 $t_w \leftarrow \mathsf{TokenGen}(sk, w)$, 需要满足

$$\mathsf{Decrypt}(sk, \mathsf{Encrypt}(pk, m, w)) = m \text{ 且 } \mathsf{Test}(t_w, \mathsf{Encrypt}(pk, m, w)) = 1.$$

一致性. 除了正确性, 类似 PEKS, 还需要刻画 PKE-PEKS 的一致性. 一般来讲, 如果对于任意 $m \in M$ 和 $w \neq w'$, 有 $\mathsf{Test}(t_{w'}, \mathsf{Encrypt}(pk, m, w)) = 0$, 则称 PKE-PEKS 方案满足一致性. PKE-PEKS 的一致性的形式化定义可以参考 PEKS 的一致性来定义.

一个安全的 PKE-PEKS 方案不仅需要保障数据隐私还要保障关键词隐私, 分别通过下面两个安全模型来刻画.

DT-Priv 安全性. 定义 PKE-PEKS 方案的数据隐私敌手 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}^{\text{DT-Priv}}(\kappa) = \left| \Pr\left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ (m_0^*, m_1^*, w^*, state) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{decrypt}}, \mathcal{O}_{\mathsf{tokengen}}, \mathcal{O}_{\mathsf{test}}}(pp, pk); \\ \beta \xleftarrow{R} \{0,1\}; \\ c^* \leftarrow \mathsf{Encrypt}(pk, m_\beta^*, w^*); \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{decrypt}}, \mathcal{O}_{\mathsf{tokengen}}, \mathcal{O}_{\mathsf{test}}}(pp, pk, state, c^*); \end{array} \right] - \frac{1}{2} \right|$$

其中, 谕言机的定义分别如下:

- $\mathcal{O}_{\mathsf{decrypt}}$ 表示解密谕言机, 其在接收到密文 c 的询问后, 输出 $m \leftarrow \mathsf{Decrypt}(sk, c)$.
- $\mathcal{O}_{\mathsf{tokengen}}$ 表示检索令牌谕言机, 其在接收到关键词 w 的询问后, 输出 $t_w \leftarrow \mathsf{TokenGen}(sk, w)$.
- $\mathcal{O}_{\mathsf{test}}$ 表示检索测试谕言机, 其在接收到关键词 w 和密文 C 的询问后, 输出 $0/1 \leftarrow \mathsf{Test}(t_w, c)$, 其中 $t_w \leftarrow \mathsf{TokenGen}(sk, w)$.

在猜测阶段, 敌手不能访问挑战密文 c^* 的解密询问, 而对于检索令牌询问和检索测试询问没有任何限制. 在上述定义中, 对于任意 PPT 敌手 $\mathcal{A}$, 若优势函数均为可忽略函数, 则称 PKE-PEKS 方案满足数据隐私安全性, 简称 DT-Priv 安全性.

KW-Priv 安全性. 定义 PKE-PEKS 方案的关键词隐私敌手 $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}^{\text{KW-Priv}}_{\mathcal{A},\text{PKE-PEKS}}(\kappa)=\left|\Pr\left[\beta'=\beta:\begin{array}{l} pp\leftarrow\mathsf{Setup}(1^\kappa);\\ (pk,sk)\leftarrow\mathsf{KeyGen}(pp);\\ (w_0^*,w_1^*,m^*,state)\leftarrow\mathcal{A}_1^{\mathcal{O}_{\mathsf{decrypt}},\mathcal{O}_{\mathsf{tokengen}},\mathcal{O}_{\mathsf{test}}}(pp,pk);\\ \beta\xleftarrow{R}\{0,1\};\\ c^*\leftarrow\mathsf{Encrypt}(pk,m^*,w_\beta^*);\\ \beta'\leftarrow\mathcal{A}_2^{\mathcal{O}_{\mathsf{decrypt}},\mathcal{O}_{\mathsf{tokengen}},\mathcal{O}_{\mathsf{test}}}(pp,pk,state,c^*);\end{array}\right]-\frac{1}{2}\right|$$

其中, 谕言机的定义同 DT-Priv 安全性中的定义. 在任意阶段, 敌手都不能访问挑战关键词 w_0^* 和 w_1^* 的检索令牌谕言机, 也不能访问 (c^*,w_0^*) 和 (c^*,w_1^*) 的检索测试谕言机. 而对于解密询问没有任何限制. 在上述定义中, 对于任意 PPT 敌手 $\mathcal{A}$, 若优势函数均为可忽略函数, 则称 PKE-PEKS 方案满足关键词隐私安全性, 简称 KW-Priv 安全性.

注记 6.4

一般地, 在描述 PKE-PEKS 方案的数据隐私安全性时, 并不需要提供检索测试谕言机. 这是因为当敌手查询 (c,w) 的检索测试谕言机时, 可以先通过查询 w 的检索令牌谕言机获得检索令牌 t_w, 然后自己运行检索匹配算法. 但是在描述关键词隐私安全性时, 提供检索测试谕言机是有必要的, 因为敌手不能查询挑战关键词的检索令牌, 但是可以查询挑战关键词的检索测试谕言机.

定义 6.3 (Jointly CCA 安全性)

对于任意 PPT 敌手 $\mathcal{A}$, 如果两个游戏中的优势函数 $\mathsf{Adv}^{\text{DT-Priv}}_{\mathcal{A}}(\kappa)$ 和 $\mathsf{Adv}^{\text{KW-Priv}}_{\mathcal{A}}(\kappa)$ 都是可忽略的, 则称 PKE-PEKS 方案满足联合选择密文攻击安全性, 简称 Jointly CCA 安全性.

下面基于 IBE 和 OTS 构造一个 Jointly CCA 安全的 PKE-PEKS 方案.

构造 6.3 (基于 IBE 的 PKE-PEKS 方案)

构造所需的组件是:

- 一个身份加密方案 $\text{IBE}=(\mathsf{Setup},\mathsf{KeyGen},\mathsf{Extract},\mathsf{Encrypt},\mathsf{Decrypt})$, 消息空间是 $\{0,1\}^n$, 身份空间是 $\{0,1\}^*$.
- 一个一次签名方案 $\text{OTS}=(\mathsf{Setup},\mathsf{KeyGen},\mathsf{Sign},\mathsf{Verify})$, 验证密钥空间是 $\{0,1\}^n$.

构造 PKE-PEKS 方案如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp_1\leftarrow\text{IBE.}\mathsf{Setup}(1^\kappa)$ 和 $pp_2\leftarrow\text{OTS.}\mathsf{Setup}(1^\kappa)$, 输出系统参数 $pp=(pp_1,pp_2)$.
- $\mathsf{KeyGen}(pp)$: 运行 $(mpk,msk)\leftarrow\text{IBE.}\mathsf{KeyGen}(pp_1)$, 输出公钥和私钥 $(pk,sk)\leftarrow(mpk,msk)$.
- $\mathsf{Encrypt}(pk,m,w)$: 输入公钥 pk、消息 m 和关键词 w, 执行以下步骤:
  1. 运行 $(vk,sigk)\leftarrow\text{OTS.}\mathsf{KeyGen}(pp_2)$.
  2. 用身份 $0||vk$ 加密消息 m, $u\leftarrow\text{IBE.}\mathsf{Encrypt}(pk,0||vk,m)$.
  3. 用身份 $1||w$ 加密验证公钥 vk, $s\leftarrow\text{IBE.}\mathsf{Encrypt}(pk,1||w,vk)$.
  4. 计算 $\sigma\leftarrow\text{OTS.}\mathsf{Sign}(sigk,u||s)$, 输出密文 $c=(vk,u,s,\sigma)$.
- $\mathsf{Decrypt}(sk,c)$: 输入私钥 sk 和密文 c, 执行以下步骤:
  1. 将密文 c 拆分为 (vk,u,s,σ).
  2. 如果 $\text{OTS.}\mathsf{Verify}(vk,u||s,\sigma)=1$, 计算 $dk\leftarrow\text{IBE.}\mathsf{Extract}(sk,0||vk)$, 输出 $m\leftarrow\text{IBE.}\mathsf{Decrypt}(dk,u)$.

  否则输出 $\perp$.
- $\mathsf{TokenGen}(sk,w)$: 输入私钥 sk 和关键词 w, 计算 $t_w\leftarrow\text{IBE.}\mathsf{Extract}(sk,1||w)$, 输出检索令牌 t_w.
- $\mathsf{Test}(t_w,c)$: 输入检索令牌 t_w 和密文 c, 执行以下步骤:
  1. 将 c 拆分为 (vk,u,s,σ).

2. 如果 $\mathsf{OTS.Verify}(vk, c||s, \sigma) = 1$,
   如果 $vk = \mathsf{IBE.Decrypt}(t_w, s)$, 则输出 1; 否则, 输出 0.
   否则, 输出 0.

♣

正确性. 构造 6.3的正确性可由 IBE 方案的正确性直接验证.

构造 6.3的安全性可由定理 6.2保证. 要证明定理 6.2, 需要分别证明构造满足 DT-Priv 安全性, 即引理 6.5和 KW-Priv 安全性, 即引理 6.6.

定理 6.2

如果 IBE 方案满足 sIND-CPA 安全性、ANO-IBE-CCA 身份匿名性和弱健壮性, 一次签名 OTS 满足 sEUF-CMA 安全性, 则构造 6.3中的 PKE-PEKS 是 jointly CCA 安全的.

♡

笔记 IBE 方案的健壮性类似 PEKS 方案健壮性的定义. 简单来说, 弱健壮性是指在允许查询若干身份标识密钥的前提下, 敌手依然无法输出两个不同的身份标识 id_1 和 id_2, 以及一个消息 m, 使得在身份 id_1 下对消息 m 加密的结果, 无法使用 id_2 的标识密钥来解密并且解密结果不等于 $\perp$ 的概率是可以忽略的.

在下面两个引理的证明中, 如果 $\mathsf{OTS.Verify}(vk, u||s, \sigma) = 1$, 则称 $c = (vk, u, s, \sigma)$ 是一个有效的 PKE-PEKS 密文. 令 $c^* = (vk^*, u^*, s^*, \sigma^*)$ 表示敌手 $\mathcal{A}$ 收到的挑战 PKE-PEKS 密文.

引理 6.5

如果 IBE 是 sIND-CPA 安全的, OTS 是 sEUF-CMA 安全的, 则构造 6.3中的 PKE-PEKS 是 DT-Priv 安全的.

♡

证明 假设 $\mathcal{A}$ 是一个以时间 t 和优势 $\mathsf{Adv}^{\text{DT-Priv}}_{\mathcal{A},\text{PKE-PEKS}}$ 攻击 PKE-PEKS 方案的 DT-Priv 安全性的敌手. 令 Forge 表示事件 "$\mathcal{A}$ 提交了一个形式为 (vk^*, u, s, σ) 的合法密文到解密谕言机" (这里假设 vk^* 在游戏开始之前就已确定.). SuccA 表示事件 "敌手 $\mathcal{A}$ 在游戏中成功". 根据 DT-Priv 安全模型的定义, 则有:

$$\begin{aligned}
\mathsf{Adv}_{\mathcal{A}} &= \left|\Pr[\mathsf{SuccA}] - 1/2\right| \\
&= \left|\Pr[\mathsf{SuccA} \wedge \mathsf{Forge}] + \Pr[\mathsf{SuccA} \wedge \overline{\mathsf{Forge}}] - 1/2\right| \\
&= \left|\Pr[\mathsf{SuccA}|\mathsf{Forge}] \cdot \Pr[\mathsf{Forge}] + \Pr[\mathsf{SuccA}|\overline{\mathsf{Forge}}] \cdot \Pr[\overline{\mathsf{Forge}}] - 1/2\right| \\
&= \left|\Pr[\mathsf{SuccA}|\mathsf{Forge}] \cdot \Pr[\mathsf{Forge}] - \Pr[\mathsf{SuccA}|\overline{\mathsf{Forge}}] \cdot \Pr[\mathsf{Forge}] + \Pr[\mathsf{SuccA}|\overline{\mathsf{Forge}}] - 1/2\right| \\
&\leq \Pr[\mathsf{Forge}] \cdot \left|\Pr[\mathsf{SuccA}|\mathsf{Forge}] - \Pr[\mathsf{SuccA}|\overline{\mathsf{Forge}}]\right| + \left|\Pr[\mathsf{SuccA}|\overline{\mathsf{Forge}}] - 1/2\right| \\
&\leq \Pr[\mathsf{Forge}] + \left|\Pr[\mathsf{SuccA}|\overline{\mathsf{Forge}}] - 1/2\right|
\end{aligned} \tag{6.3}$$

下面分别证明以下两个公式成立:

$$\Pr[\mathsf{Forge}] \leq \mathsf{Adv}_{\mathcal{F}}(\kappa) \tag{6.4}$$

$$\left|\Pr[\mathsf{SuccA}|\overline{\mathsf{Forge}}] - 1/2\right| \leq \mathsf{Adv}_{\mathcal{D}}(\kappa) \tag{6.5}$$

其中, $\mathcal{F}$ 表示攻击一次签名方案 sEUF-CMA 安全性的 PPT 敌手, $\mathcal{D}$ 表示攻击身份加密方案 IND-CPA 安全性的 PPT 敌手.

公式 (6.4)的证明. 利用敌手 $\mathcal{A}$ 构造一个伪造算法 $\mathcal{F}$ 攻击 OTS 的 sEUF-CMA 安全性. $\mathcal{F}$ 按照下面的方式模拟 $\mathcal{A}$ 在 DT-Priv 游戏中的挑战者行为:

- 初始化: 输入安全参数 κ 和一次签名的验证公钥 vk^* (由 $\mathsf{OTS.KeyGen}(pp_2)$ 生成, 其中 $pp_2 \leftarrow \mathsf{OTS.Setup}(1^\kappa)$), $\mathcal{F}$ 运行 $\mathsf{IBE.Setup}(1^\kappa)$ 获取 IBE 的公开参数 pp_1, 运行 $\mathsf{IBE.KeyGen}(pp_1)$ 获取 IBE 的主公钥和主私钥 (mpk, msk) 并将其作为 PKE-PEKS 的公钥和私钥 (pk, sk). $\mathcal{F}$ 将系统参数 $pp = (pp_1, pp_2)$ 和公钥 pk 发送给 $\mathcal{A}$.
- 阶段 1 询问: 由于 $\mathcal{F}$ 知道 PKE-PEKS 的私钥 sk, 所以 $\mathcal{F}$ 可以回答敌手的检索令牌询问、检索测试询问和解密询问. 如果 $\mathcal{A}$ 在该阶段提交了一个有效密文 (vk^*, u, s, σ) 到解密谕言机, 则 $\mathcal{F}$ 输出 $(u||s, \sigma)$ 作为自己的伪造结果并终止游戏.

- 挑战: 当 $\mathcal{A}$ 输出两个挑战消息 m_0^* 和 m_1^*, 以及一个挑战关键词 w^* 时, $\mathcal{F}$ 按以下方式处理: 选择一个随机比特 b, 计算 $u^* \leftarrow \text{IBE.Encrypt}(pk, 0||vk^*, m_b^*)$, $s^* \leftarrow \text{IBE.Encrypt}(pk, 1||w^*, vk^*)$, 并通过询问自己的一次签名谕言机获取消息 $u^*||s^*$ 的签名 σ^*. 最后, $\mathcal{F}$ 发送挑战密文 $(vk^*, u^*, s^*, \sigma^*)$ 给敌手 $\mathcal{A}$.
- 阶段 2 询问: 如果 $\mathcal{A}$ 在该阶段询问了一个合法的解密查询 (vk^*, u, s, σ), 其中 $(u, s, \sigma) \neq (u^*, s^*, \sigma^*)$, 则 $\mathcal{F}$ 直接输出 $(u||s, \sigma)$ 作为伪造的签名.
- 猜测: 最终, $\mathcal{A}$ 将输出一个猜测比特 b' 作为对 b 的猜测结果.

显而易见, $\mathcal{F}$ 模拟的上述游戏环境与敌手 $\mathcal{A}$ 在真实 DT-Priv 游戏中的视图是完全一样的并且 $\mathcal{F}$ 的成功概率与 $\Pr[\text{Forge}]$ 相同. 根据 OTS 的安全性定义, 则有 $\Pr[\text{Forge}] \leq \text{Adv}_{\mathcal{F}}(\kappa)$. 从而, 公式 (6.4)成立!

公式 (6.5)的证明. 利用敌手 $\mathcal{A}$ 构造一个区分算法 $\mathcal{D}$ 攻击 IBE 的选择身份 IND-CPA 安全性. $\mathcal{D}$ 按照下面的方式模拟 $\mathcal{A}$ 在 DT-Priv 游戏中的挑战者行为:

- 初始化: 输入 IBE 的公开参数 pp_1, $\mathcal{D}$ 运行 $\text{OTS.Setup}(1^\kappa)$ 生成 OTS 的公开参数 pp_2, 再运行 $\text{OTS.KeyGen}(pp_2)$ 生成 $(vk^*, sigk^*)$. 接下来, 选择一个身份 $id^* = 0||vk^*$ 并发送给 $\mathcal{D}$ 的挑战者 (即, IBE 方案的挑战者) 作为目标身份, 并获取 IBE 方案的主公钥 mpk. $\mathcal{D}$ 将 PKE-PEKS 的公钥设置为 $pk = mpk$ 并发送给攻击者 $\mathcal{A}$.
- 阶段 1 询问: 当收到敌手 $\mathcal{A}$ 的检索令牌询问、检索测试询问和解密询问时, $\mathcal{D}$ 按以下方式回答:
  - 检索令牌询问 $\langle w \rangle$: $\mathcal{D}$ 查询身份 $\langle 1||w \rangle$ 的 IBE 密钥, 将其作为关键词 w 的检索令牌返回给 $\mathcal{A}$.
  - 检索测试询问 $\langle C, w \rangle$: $\mathcal{D}$ 首先按照询问检索令牌的方式获取 w 的检索令牌 t_w, 然后运行 $\text{Test}(t_w, c)$, 将结果返回给 $\mathcal{A}$.
  - 解密询问 $\langle c \rangle$: $\mathcal{D}$ 将 c 拆分为 (vk, u, s, σ). 如果 $\text{OTS.Verify}(vk, u||s, \sigma) = 0$, 则 $\mathcal{D}$ 拒绝解密, 返回 $\perp$. 否则, $\mathcal{D}$ 先通过 IBE 的密钥询问, 获取身份 $0||vk$ 对应的解密密钥 dk, 再计算 $\text{IBE.Decrypt}(dk, u)$ 并将结果返回给 $\mathcal{A}$.
- 挑战: 当 $\mathcal{A}$ 输出两个挑战消息 m_0^* 和 m_1^* 及一个挑战关键词 w^* 时, $\mathcal{D}$ 按以下方式处理: $\mathcal{D}$ 将 m_0^* 和 m_1^* 发送给 IBE 挑战者, 并获取挑战密文 $u^* \leftarrow \text{IBE.Encrypt}(pk, 0||vk^*, m_b^*)$, 其中 b 是 $\mathcal{D}$ 的挑战者随机选择的. 接下来, $\mathcal{D}$ 计算 $s^* \leftarrow \text{IBE.Encrypt}(pk, 1||w^*, vk^*)$, $\sigma^* \leftarrow \text{OTS.Sign}(sigk^*, u^*||s^*)$. 最后, $\mathcal{D}$ 将 $c^* = (vk^*, u^*, s^*, \sigma^*)$ 发送给 $\mathcal{A}$ 作为挑战密文.
- 阶段 2 询问: $\mathcal{A}$ 可以自适应地进行更多的检索令牌询问、检索测试询问和解密询问. 由于 IBE 挑战者允许 $\mathcal{D}$ 询问身份标识为 $\langle 1||w \rangle$ 的用户密钥, 所以 $\mathcal{D}$ 可以回答 $\mathcal{A}$ 的所有检索令牌询问和检索测试询问. 对于解密询问, $\mathcal{D}$ 的回答方式同阶段 1 询问, 但是对于挑战密文 $\langle c^* \rangle$, $\mathcal{D}$ 直接返回 $\perp$.
- 猜测: 最终, $\mathcal{A}$ 输出一比特 b' 作为对 b 的猜测结果. $\mathcal{D}$ 将 b' 作为自己的输出结果返回给 IBE 挑战者.

在上述模拟游戏中, 若事件 Forge 未发生, 则 $\mathcal{D}$ 攻击 IBE 方案的选择身份 IND-CPA 安全性的密钥询问都是合法有效的. 所以, $\mathcal{D}$ 完美地模拟了 $\mathcal{A}$ 在 DT-Priv 游戏中的环境. 令 SuccD 表示事件 "$\mathcal{D}$ 在选择身份 IND-CPA 实验中输出正确的猜测比特". 显而易见: $\Pr[\text{SuccD}] = \Pr[\text{SuccA}|\overline{\text{Forge}}]$. 根据 IBE 的安全性, 可得

$$\left|\Pr[\text{SuccA}|\overline{\text{Forge}}] - 1/2\right| = |\Pr[\text{SuccD}] - 1/2| \leq \text{Adv}_{\mathcal{D}}(\kappa)$$

从而公式 (6.5)成立!

基于公式 (6.4)和公式 (6.5), 可以得到 $\text{Adv}_{\mathcal{A}}$ 的具体上界. 引理 6.5得证! □

笔记 在证明公式 (6.5)时, 敌手提交的解密查询 $C = (vk, u, s, \sigma)$ 有两种形式: 一种是 $vk = vk^*$. 此时, 模拟者 $\mathcal{D}$ 不能询问 IBE 挑战者关于 $0||vk^*$ 的身份密钥, 从而无法回答这类密文的解密询问. 二是 $vk \neq vk^*$. 此时, 模拟者 $\mathcal{D}$ 可以正常询问 IBE 挑战者关于 $0||vk$ 的身份密钥, 继而用于解密回答. 由于事件 Forge 从未发生, 所以第一种情况不会出现, 从而模拟算法能够正确运行.

引理 6.6

如果 IBE 满足 ANO-IBE-CCA 匿名性和弱健壮性, OTS 是 sEUF-CMA 安全的, 则构造 6.3中的 PKE-PEKS 是 KW-Priv 安全的. ♡

证明 假设 $\mathcal{A}$ 是一个以优势 $\text{Adv}_{\mathcal{A}}^{\text{KW-Priv}}$ 攻击 PKE-PEKS 方案 KW-Priv 安全性的敌手. 令 Forge 表示事件 "$\mathcal{A}$ 在阶段 2 询问提交了一个检索测试询问 $\langle c, w \rangle$, 其中 $c = (vk^*, u, s, \sigma)$ 是一个有效的 PKE-PEKS 密文, $w \in \{w_0^*, w_1^*\}$".

令 Break 表示事件 "密文 s^* 在身份 $1||w_{1-b}^*$ 下的解密结果不等于 $\perp$". 根据 KW-Priv 安全模型的定义, 则有

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}}^{\text{KW-Priv}} &= \left|\Pr[\mathsf{SuccA}] - 1/2\right| \\ &= \left|\Pr[\mathsf{SuccA} \wedge (\mathsf{Forge} \vee \mathsf{Break})] + \Pr[\mathsf{SuccA} \wedge \overline{\mathsf{Forge} \vee \mathsf{Break}}] - 1/2\right| \\ &\leq \Pr[\mathsf{Forge} \vee \mathsf{Break}] + \left|\Pr[\mathsf{SuccA}|\overline{\mathsf{Forge} \vee \mathsf{Break}}] - 1/2\right| \\ &\leq \Pr[\mathsf{Forge}] + \Pr[\mathsf{Break}] + \left|\Pr[\mathsf{SuccA}|\overline{\mathsf{Forge} \vee \mathsf{Break}}] - 1/2\right|\end{aligned} \tag{6.6}$$

下面分别证明以下三个公式成立:

$$\Pr[\mathsf{Forge}] \leq \mathsf{Adv}_{\mathcal{F}}(\kappa) \tag{6.7}$$

$$\Pr[\mathsf{Break}] \leq \mathsf{Adv}_{\mathcal{B}}(\kappa) \tag{6.8}$$

$$\left|\Pr[\mathsf{SuccA}|\overline{\mathsf{Forge} \vee \mathsf{Break}}] - 1/2\right| \leq \mathsf{Adv}_{\mathcal{D}}(\kappa) \tag{6.9}$$

其中, $\mathcal{F}$ 表示攻击一次签名方案 sEUF-CMA 安全性的 PPT 敌手, $\mathcal{B}$ 表示攻击身份加密方案弱健壮性的 PPT 敌手, $\mathcal{D}$ 表示攻击身份加密方案 ANO-IBE-CCA 匿名性的 PPT 敌手.

公式 (6.7)的证明. 利用敌手 $\mathcal{A}$ 构造一个伪造算法 $\mathcal{F}$ 攻击 OTS 的 sEUF-CMA 安全性. $\mathcal{F}$ 按照下面的方式模拟 $\mathcal{A}$ 在 KW-Priv 游戏中的挑战者行为:

- 初始化: 输入安全参数 κ 和一次签名的验证公钥 vk^* (由 $\mathsf{OTS.KeyGen}(pp_2)$ 生成, 其中 $pp_2 \leftarrow \mathsf{OTS.Setup}(1^\kappa)$), $\mathcal{F}$ 运行 $\mathsf{IBE.Setup}(1^\kappa)$ 获取 IBE 的公开参数 pp_1, 运行 $\mathsf{IBE.KeyGen}(pp_1)$ 获取 IBE 的主公钥 mpk 和主私钥 msk 并将其分别作为 PKE-PEKS 的公钥 pk 和私钥 sk. $\mathcal{F}$ 将公开参数 $pp = (pp_1, pp_2)$ 和公钥 pk 发送给 $\mathcal{A}$.
- 阶段 1 询问: 由于 $\mathcal{F}$ 知道 PKE-PEKS 的私钥 sk, 所以 $\mathcal{F}$ 可以回答敌手的检索令牌询问、检索测试询问和解密询问.
- 挑战: 当 $\mathcal{A}$ 输出两个挑战关键词 w_0^* 和 w_1^*, 以及一个挑战消息 m^* 时, $\mathcal{F}$ 按以下方式处理: 选择一个随机比特 b, 计算 $u^* \leftarrow \mathsf{IBE.Encrypt}(pk, 0||vk^*, m^*)$, $s^* \leftarrow \mathsf{IBE.Encrypt}(pk, 1||w_b^*, vk^*)$, 并通过询问自己的一次签名谕言机获取消息 $u^*||s^*$ 的一个签名 σ^*. 最后, $\mathcal{F}$ 将挑战密文 $(vk^*, u^*, s^*, \sigma^*)$ 发送给敌手 $\mathcal{A}$.
- 阶段 2 询问: 如果 $\mathcal{A}$ 提交了一个合法的匹配测试询问 $\langle c, w\rangle$, 其中 $c = (vk^*, u, s, \sigma)$, w 等于 w_0^* 或 w_1^*, 由于 $(u, s, \sigma) \neq (u^*, s^*, \sigma^*)$, $\mathcal{F}$ 直接将 $(c||s, \sigma)$ 作为伪造的签名输出.
- 猜测: 最终, $\mathcal{A}$ 将输出一个猜测比特 b' 作为对 b 的猜测结果.

显而易见, $\mathcal{F}$ 模拟的上述游戏环境与敌手 $\mathcal{A}$ 在真实 KW-Priv 游戏中的视图是完全一样的并且 $\mathcal{F}$ 的成功概率与 $\Pr[\mathsf{Forge}]$ 相同. 由此推出 $\Pr[\mathsf{Forge}] \leq \mathsf{Adv}_{\mathcal{F}}(\kappa)$, 从而公式 (6.7)成立! □

公式 (6.8)的证明. 公式 (6.8)成立的基础是 IBE 方案具有弱健壮性, 可通过下面的归约方式证明. 利用 $\mathcal{A}$ 构造一个算法 $\mathcal{B}$ 攻击 IBE 方案的弱健壮性. $\mathcal{B}$ 按照以下方式模拟 $\mathcal{A}$ 在 PKE-PEKS 的 KW-Priv 游戏中的挑战者: 输入安全参数 κ, IBE 的公开参数 pp_1 和主公钥 mpk, $\mathcal{B}$ 运行 $\mathsf{OTS.Setup}(1^\kappa)$ 生成一次签名的公开参数 pp_2, $\mathcal{B}$ 将 PKE-PEKS 的公开参数 $pp = (pp_1, pp_2)$ 和公钥 $pk = mpk$ 发送给 $\mathcal{A}$. 由于 $\mathcal{B}$ 可以询问 IBE 挑战者形如 $1||w$ 和 $0||vk$ 的身份密钥, 所以 $\mathcal{B}$ 可以回答 $\mathcal{A}$ 的检索令牌询问、检索测试询问和解密询问. 当 $\mathcal{A}$ 输出一个挑战消息 m^* 及两个挑战关键词 w_0^* 和 w_1^* 时, $\mathcal{B}$ 运行 $\mathsf{OTS.KeyGen}(pp_2)$ 生成签名方案的一对公钥和私钥 $(vk^*, sigk^*)$, 选择一个随机比特 b, 将两个身份标识 $1||w_b^*$ 和 $1||w_{1-b}^*$ 及消息 vk^* 发送给 IBE 挑战者. 假设 $s^* \leftarrow \mathsf{IBE.Encrypt}(pk, 1||w_b^*, vk^*)$ 是 $\mathcal{B}$ 的挑战者生成的密文. $\mathcal{B}$ 将其作为 PKE-PEKS 密文的一部分. 因此, $\mathcal{B}$ 在 IBE 的弱健壮性实验中成功的概率恰好是 $\Pr[\mathsf{Break}]$. 由 IBE 的弱健壮性可知公式 (6.8)成立. □

笔记 在模拟 KW-Priv 安全性环境时, 由于敌手不能查询挑战关键词 w_0^* 和 w_1^* 的检索令牌, 所以模拟者 (算法 $\mathcal{B}$) 也不用向 IBE 挑战者查询身份标识为 $1||w_0^*$ 和 $1||w_1^*$ 的密钥. 因此, 模拟者可以回答敌手所有合法的询问. IBE 挑战者返回给模拟者的密文 $s^* \leftarrow \mathsf{IBE.Encrypt}(pk, 1||w_b^*, vk^*)$ 的分布和 KW-Priv 安全模型中的密文分布一致. 因此, 若事件 Break 发生, 则密文 s^* 在身份 $1||w_{1-b}^*$ 下的解密结果不等于 $\perp$, 这等于 $\mathcal{B}$ 攻破了 IBE 方案的弱健壮性.

公式 (6.9)的证明. 利用 $\mathcal{A}$ 构造一个区分算法 $\mathcal{D}$ 以攻击 IBE 的 ANO-IBE-CCA 匿名性. $\mathcal{D}$ 按照下面的方式模拟 $\mathcal{A}$ 在 KW-Priv 游戏中的挑战者行为:

- 初始化: 输出 IBE 的公开参数 pp_1 和主公钥 mpk, $\mathcal{D}$ 运行 OTS.Setup(1^κ) 生成 OTS 的公开参数 pp_2. 接下来, $\mathcal{D}$ 将公开参数 $pp = (pp_1, pp_2)$ 以及 PKE-PEKS 的公钥 $pk = mpk$ 发送给敌手 $\mathcal{A}$.
- 阶段 1 询问: 当收到 $\mathcal{A}$ 的检索令牌询问、检索测试询问和解密询问时, $\mathcal{D}$ 按以下方式回答:
  - 检索令牌询问 $\langle w \rangle$: $\mathcal{D}$ 向 IBE 挑战者查询身份 $\langle 1||w \rangle$ 的用户密钥, 将查询结果发送给 $\mathcal{A}$.
  - 检索测试询问 $\langle c, w \rangle$: $\mathcal{D}$ 将 c 拆分为 (vk, u, s, σ). 如果 OTS.Verify$(vk, u||s, \sigma) = 0$, $\mathcal{D}$ 输出 0. 否则, $\mathcal{D}$ 向 IBE 挑战者查询在身份 $\langle 1||w, s \rangle$ 下的解密结果. 如果解密结果等于 vk, 则 $\mathcal{D}$ 返回 1, 否则返回 0
  - 解密询问 $\langle c \rangle$: $\mathcal{D}$ 将 c 拆分为 (vk, u, s, σ). 如果 OTS.Verify$(vk, u||s, \sigma) = 0$, 则 $\mathcal{D}$ 拒绝解密并返回 $\perp$. 否则, $\mathcal{D}$ 询问 IBE 挑战者的解密询问 $(0||vk, u)$, 并将结果返回给敌手 $\mathcal{A}$.
- 挑战: 当 $\mathcal{A}$ 输出一个挑战消息 m^* 及两个挑战关键词 w_0^* 和 w_1^* 时, $\mathcal{D}$ 按以下方式处理:
  1. 运行 $(vk^*, sk_\sigma^*) \leftarrow$ OTS.KeyGen(pp_2).
  2. 计算消息 m^* 的密文 $u^* \leftarrow$ IBE.Encrypt$(pk, 0||vk^*, m^*)$.
  3. 将 vk^* 作为消息同两个挑战身份标识 $1||w_0^*$ 和 $1||w_1^*$ 发送给 IBE 挑战者, 从而得到消息 vk^* 在身份 $1||w_b^*$ 下的密文 s^*, 其中 b 是 $\mathcal{D}$ 的挑战者随机选取的比特.
  4. 计算签名 $\sigma^* \leftarrow$ OTS.Sign$(sk_\sigma^*, u^*||s^*)$.
  5. 将 $c^* = (vk^*, u^*, s^*, \sigma^*)$ 作为挑战密文发送给敌手 $\mathcal{A}$.
- 阶段 2 询问: $\mathcal{A}$ 可以继续自适应地询问检索令牌谕言机、检索测试谕言机和解密谕言机, $\mathcal{D}$ 按以下方式回答:
  - 检索令牌询问 $\langle w \rangle$: 只要 $w \neq w_0^*, w_1^*$, $\mathcal{D}$ 就可以利用 IBE 挑战者查询身份标识为 $1||w$ 的用户密钥, 并将该密钥作为检索令牌发送给 $\mathcal{A}$.
  - 检索测试询问 $\langle c, w \rangle$: 若询问 $\langle c^*, w_0^* \rangle$ 或 $\langle c^*, w_1^* \rangle$ 的检索测试, 根据 KW-Priv 游戏规则, $\mathcal{D}$ 将拒绝回答. 否则, $\mathcal{D}$ 将 c 拆分为 (vk, u, s, σ), 首先验证 OTS.Verify$(vk, u||s, \sigma) = 1$ 是否成立. 如果不成立, 则 $\mathcal{D}$ 返回 0. 如果成立并且 w 不等于 w_0^* 或 w_1^*, 则 $\mathcal{D}$ 查询 IBE 挑战者的解密询问 $\langle 1||w, s \rangle$, 如果解密结果等于 vk, 则 $\mathcal{D}$ 返回 1; 如果不等于 vk, 则返回 0. 否则, $\mathcal{D}$ 按以下方式处理:
    - 情形 1: $vk = vk^*$. 此时, 事件 Forge 发生 (w 等于 w_0^* 或 w_1^*. 对于一个合法的询问, 必然有 $c \neq c^*$), 则 $\mathcal{D}$ 终止游戏并返回一个随机比特.
    - 情形 2: $vk \neq vk^*$ 且 $s \neq s^*$, 则 $\mathcal{D}$ 利用 IBE 挑战者获取 $\langle 1||w, s \rangle$ 的解密结果, 若解密结果等于 vk, 则返回 1, 否则返回 0.
    - 情形 3: $vk \neq vk^*$ 且 $s = s^*$, $\mathcal{D}$ 返回 0.
  - 解密询问 $\langle c \rangle$: $\mathcal{D}$ 按照阶段 1 询问中的方式进行回答 $\mathcal{A}$ 的解密查询. 由于 IBE 挑战者允许 $\mathcal{D}$ 询问所有形如 $\langle 0||vk, u \rangle$ 的解密查询, 所以 $\mathcal{D}$ 可以正确地回答所有解密询问.
- 猜测: 最终, $\mathcal{A}$ 输出一比特 b' 作为对 b 的猜测结果, $\mathcal{D}$ 将 b' 返回给 HIBE 挑战者, 作为自己的猜测结果.

在上述模拟游戏中, $\mathcal{D}$ 攻击 IBE 的 ANO-IBE-CCA 匿名性的策略是合法的. 在事件 Forge 和 Break 都未发生的条件下, 检索测试询问中的情形 1 和情形 3 不会出现, 因此 $\mathcal{D}$ 完美地模拟了 $\mathcal{A}$ 在 KW-Priv 游戏中的环境. 令 SuccD 表示事件 "$\mathcal{D}$ 在 ANO-IBE-CCA 实验中输出正确的猜测比特". 显而易见: $\Pr[\mathsf{SuccD}] = \Pr[\mathsf{SuccA}|\overline{\mathsf{Forge} \vee \mathsf{Break}}]$. 由 IBE 的匿名性, 公式 (6.9)成立, 即 $|\Pr[\mathsf{SuccA}|\overline{\mathsf{Forge} \vee \mathsf{Break}}] - 1/2| = |\Pr[\mathsf{SuccD} - 1/2| \leq \mathsf{Adv}_{\mathcal{D}}(\kappa)$.

根据公式 (6.7)、公式 (6.8)和公式 (6.9), 可以得到 $\mathsf{Adv}_{\mathcal{A}}$ 的上界, 即

$$\begin{aligned} |\Pr[\mathsf{SuccA}] - 1/2| &\leq \Pr[\mathsf{Forge}] + \Pr[\mathsf{Break}] + \left|\Pr[\mathsf{SuccA}|\overline{\mathsf{Forge} \vee \mathsf{Break}}] - 1/2\right| \\ &\leq \mathsf{Adv}_{\mathcal{F}}(\kappa) + \mathsf{Adv}_{\mathcal{B}}(\kappa) + \mathsf{Adv}_{\mathcal{D}}(\kappa) \end{aligned}$$

综上, 引理 6.6得证!

PAEKS 方案的构造. PAEKS 可以看作是对 PEKS 和 PKE-PEKS 安全模型的一种提升. Boneh 等 [185] 提出 PEKS 安全模型时, 仅定义了密文中的关键词隐私性, 无法保障检索陷门中的关键词隐私. PKE-PEKS 方案尽管在加密功能上增加了关键词的解密服务, 其安全模型依然仅考虑了密文中的关键词或消息的隐私性. 事实上, 无论是 PEKS 还是 PKE-PEKS 或者其他可搜索公钥加密方案, 如果关键词加密算法是公开可计算的, 则敌手在获得一个检索令牌时可能获取检索令牌中的关键词信息. 这是因为攻击者可以猜测一个关键词并生成该关键词的密文, 然后利用检索测试算法判断该密文与获取的检索令牌是否匹配, 从而获取检索令牌中的关键词信息. 该攻击通常称为关键

词猜测攻击 (keyword guessing attacks, KGA). 如果关键词空间较小, 则该攻击是非常有效的. 例如, 韦氏字典中仅包含大约 22500 $\approx 2^{18}$ 个关键词. 攻击者从一个检索令牌中获取关键词信息的概率至少为 $1/2^{18}$. 目前, 抵抗关键词猜测攻击的技术主要有以下几种:

- **扩大关键词空间技术.** 2009 年, Tang 等 [339] 首次提出基于关键词注册的 PEKS 方案. 该方案的基本思想是引入一个关键词注册服务器, 用户在进行关键词加密或生成检索令牌前, 需要利用安全信道将该关键词发送给注册服务器, 注册服务器利用自己的密钥将关键词映射成一个新的 (无语义的) 关键词并通过安全信道传送给用户, 如图 6.4所示, 将原始关键词 W 映射到 $W' = \mathsf{H}(K, W)$, 其中 H 是一个哈希函数. 为了减少安全通信代价, 还可以将密钥 k 替换为关键词的盲签名, 不仅能够隐藏注册关键词的信息还可以在公开信道上传输. 此时只需要将注册服务器替换为一个关键词匿名签名服务器 [340]. 该类技术一般需要注册服务器保持在线, 所以注册服务器的可靠性和安全性对系统的影响非常大.

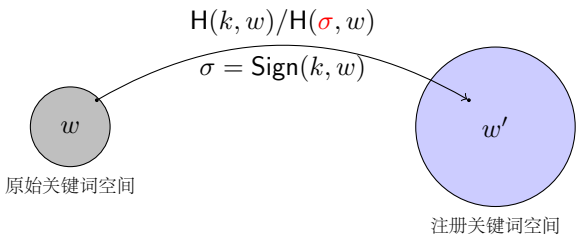


图 6.4: 扩大关键词空间技术

- **指定验证者技术.** 2008 年, Baek 等 [342] 提出无安全信道可搜索公钥加密方案的概念, 也称为指定验证者的可搜索公钥加密方案 (designated PEKS, dPEKS), 如图 6.5所示. 其目的在于去掉用户和服务器之间的安全信道, 提高方案效率. 在 dPEKS 中, 关键词密文由接收者和指定检索服务器的公钥联合加密而来, 只有指定的服务器才可以利用检索令牌进行密文检索. 然而, 该方案后来被发现仍然存在安全缺陷, 并不能抵抗离线关键词猜测攻击 [343]. 事实上, 该技术本身存在一定的安全隐患, 这是因为敌手在加密关键词时可以不使用指定服务器的公钥或者选择一个自己生成的公钥, 从而使得该敌手可以进行检索测试操作. 因此, 许多方案后来发现并不安全 [344, 345].

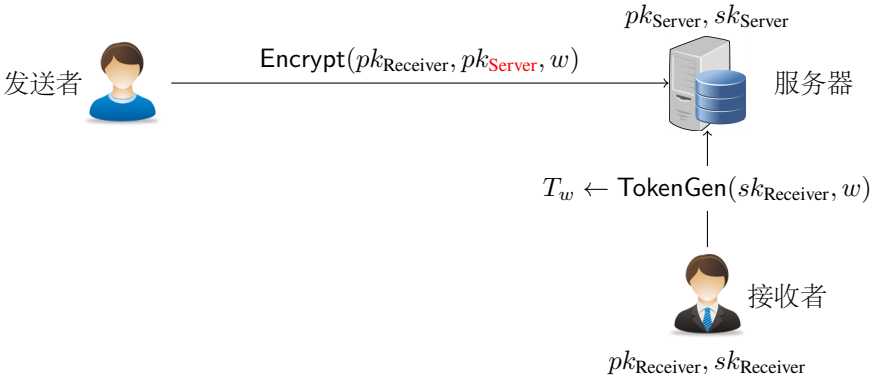


图 6.5: 指定验证者技术

- **指定发送者技术.** 2017 年, Huang 等 [341] 提出一种称作可搜索公钥认证加密的概念 (PAEKS), 如图 6.6所示. 在加密关键词时, 通过引入发送者的私钥使得检索令牌仅能用于检索指定发送者的关键词密文, 从而使关键词密文和检索令牌同时满足不可区分性. 当前, 该思想已被推广到构造无证书、基于身份等环境下的可搜索公钥加密方案 [346, 347, 348].

下面重点介绍 PAEKS 的基本概念和方案构造.

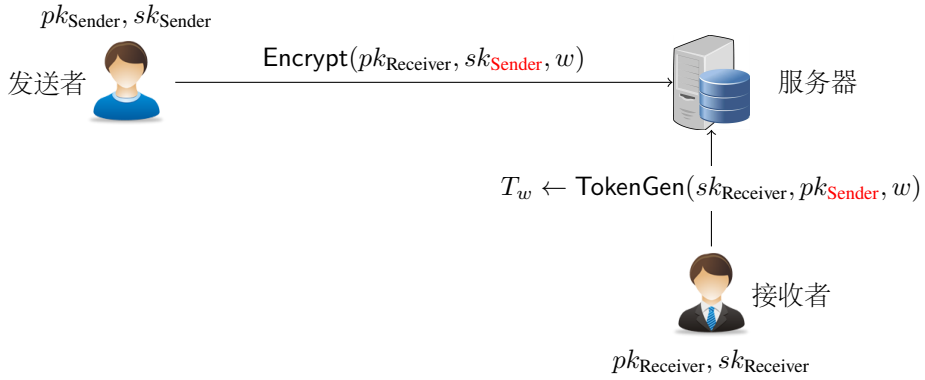


图 6.6: 指定发送者技术

定义 6.4 (可搜索公钥认证加密)

一个可搜索公钥认证加密方案 PAEKS 包含以下 6 个 PPT 算法:

- $\text{Setup}(1^\kappa)$: 系统参数生成算法以安全参数 1^κ 为输入, 输出系统公开参数 pp, 其中 pp 包含了用户的公钥空间 PK、私钥空间 SK、关键词空间 W、密文空间 C 和检索令牌空间 T 的描述. 类似公钥加密方案, 该算法由可信第三方生成并公开, 系统中的所有用户共享, 所有算法均将 pp 作为输入的一部分.
- $\text{KeyGen}_S(pp)$: 数据发送者密钥生成算法以公开参数 pp 为输入, 输出一对公/私钥对 (pk_S, sk_S), 其中公钥 pk_S 公开, 私钥 sk_S 秘密保存.
- $\text{KeyGen}_R(pp)$: 数据接收者密钥生成算法以公开参数 pp 为输入, 输出一对公/私钥对 (pk_R, sk_R), 其中公钥 pk_R 公开, 私钥 sk_R 秘密保存.
- $\text{Encrypt}(sk_S, pk_R, w)$: 关键词加密算法以发送者私钥 sk_S、接收者公钥 pk_R 和关键词 $w \in W$ 为输入, 输出关键词 w 的密文 $c_w \in C$.
- $\text{TokenGen}(sk_R, pk_S, w)$: 检索令牌生成算法以接收者私钥 sk_R、发送者公钥 pk_S 和关键词 $w \in W$ 为输入, 输出关键词 w 的检索令牌 t_w.
- $\text{Test}(pk_S, pk_R, t_{w'}, c_w)$: 检索算法以发送者公钥 pk_S、接收者公钥 pk_R、关键词 w' 的检索令牌 $T_{w'}$ 和关键词 w 的密文 c_w 为输入, 如果 $w = w'$, 则输出 1, 否则, 输出 0.

♣

正确性和一致性. 类似 PEKS 和 PKE-PEKS, PAEKS 的正确性保证了关键词密文的可检索功能, 而一致性降低了检索的错误率. 具体地, 对于任意密钥 (pk_R, sk_R) 和 (pk_S, sk_S), 任意两个关键词 w 和 w', 令 $c \leftarrow \text{Encrypt}(pk_R, sk_S, w)$, $t_{w'} \leftarrow \text{TokenGen}(sk_R, pk_S, w')$. 如果 $w = w'$, 则 $\Pr[\text{Test}(pk_R, pk_S, c, t_{w'}) = 1] = 1 - \text{negl}(\kappa)$; 如果 $w \neq w'$, 则 $\Pr[\text{Test}(pk_R, pk_S, c, t_{w'}) = 0] = 1 - \text{negl}(\kappa)$.

笔记 如果去掉 PAEKS 方案中数据发送者的密钥生成算法, 从而将数据发送者密钥从加密算法和检索令牌生成算法参数列表中去除, 则上述定义退化为标准的 PEKS 方案的定义.

PAEKS 方案的安全模型包含两个方面: 关键词密文不可区分性 (cipher-keyword indistinguishability, CI) 和检索令牌不可区分性 (trapdoor indistinguishability, TI), 分别保障关键词密文和检索令牌的隐私. 令 (pk_S, sk_S) 和 (pk_R, sk_R) 分别是一组受攻击的数据发送者和攻击者. 在这两种模型中, 敌手具有下面两种攻击能力:

- **选择关键词密文攻击 (chosen keyword to cipher-keyword, CKC):** 在 CKC 攻击中, 敌手拥有获取任意关键词密文的能力, 即敌手可以选择一个关键词 w 和指定的任意接收者公钥 pk, 获取该关键词相应的密文. 具体地, 敌手具有访问选择关键词密文谕言机 $\mathcal{O}_{\text{encrypt}}(sk_S, \cdot, \cdot)$ 的能力. 敌手可以自适应地选择一个关键词和一个接收者公钥 pk, 通过该谕言机获取关键词密文 $c_w = \text{Encrypt}(sk_S, pk, w)$.
- **选择检索令牌攻击 (chosen keyword to trapdoor, CKT):** 在 CKT 攻击中, 敌手拥有获取任意关键词检索令牌的能力, 即敌手可以选择一个关键词 w 和指定的任意发送者的公钥 pk, 获取该关键词相应的检索令牌. 类

似地, 敌手这一能力通过一个检索令牌生成谕言机 $\mathcal{O}_{\mathsf{TokenGen}}(sk_R,\cdot,\cdot)$ 来刻画; 敌手可以自适应地选择一个关键词 w 和一个发送者公钥 pk, 通过该谕言机获取关键词检索令牌 $t_w=\mathsf{TokenGen}(sk_R,pk,w)$.

令 w_0^* 和 w_1^* 是敌手选择的两个挑战关键词, 则敌手在访问上面两个谕言机时必须有所限制, 否则从理论上无法保障任何安全性. 例如在 CI 安全模型中, 敌手会收到某一个挑战关键词的密文 $c_{w_b^*}$. 显而易见, 敌手不能访问挑战关键词的检索令牌. 否则, 敌手可以通过检索测试算法直接打破 CI 安全性. 除了这种平凡攻击外, 有些 CI 安全模型还限制敌手访问挑战关键词的密文. 如果不限制敌手访问挑战关键词的密文, 这种选择关键词密文攻击也称为完全选择关键词密文攻击 (fully CKC attacks).

关键词密文不可区分安全性. 定义可搜索公钥认证加密方案敌手 $\mathcal{A}=(\mathcal{A}_1,\mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{CI}}(\kappa)=\left|\Pr\left[\beta'=\beta:\begin{array}{l}pp\leftarrow\mathsf{Setup}(1^\kappa);\\(pk_S,sk_S)\leftarrow\mathsf{KeyGen}_S(pp);\\(pk_R,sk_R)\leftarrow\mathsf{KeyGen}_R(pp);\\(w_0,w_1,state)\leftarrow\mathcal{A}_1^{\mathcal{O}_{\mathsf{encrypt}},\mathcal{O}_{\mathsf{tokengen}}}(pp,pk_S,pk_R);\\\beta\xleftarrow{\mathrm{R}}\{0,1\};\\c^*\leftarrow\mathsf{Encrypt}(sk_S,pk_R,w_\beta);\\\beta'\leftarrow\mathcal{A}_2^{\mathcal{O}_{\mathsf{encrypt}},\mathcal{O}_{\mathsf{tokengen}}}(pp,pk_S,pk_R,state,c^*);\end{array}\right]-\frac{1}{2}\right|$$

在上述定义中, 敌手可以提交任意形如 (pk,w) 的询问到关键词密文谕言机, 但是不能提交形如 (pk_R,w_b^*) 的询问到检索令牌谕言机. 如果任意的 PPT 敌手 $\mathcal{A}$ 在上述定义中的优势函数均为可忽略函数, 则称可搜索公钥认证加密方案 PAEKS 是 (fully) CI 安全的.

PAEKS 的加密算法并不是完全公开可计算的, 需要知道数据发送者的私钥才能计算. 因此, PAEKS 加密算法可以看成是一个对称加密算法. 众所周知, 在对称加密算法中, 若攻击者是一个自适应选择明文攻击敌手, 则单密文不可区分蕴含多密文不可区分. 这里的自适应选择明文攻击敌手是允许攻击者选择挑战消息并获取相应的加密密文. 对于窃听攻击者, 该结论不一定成立, 详细内容可以参考文献 [349] 定理 3.24. 正因为如此, 有必要定义 PAEKS 的多关键词密文不可区分性 [350]. 多关键词密文安全性游戏的定义类似前面的关键词密文不可区分性安全模型的定义, 不同之处在于挑战阶段, 敌手提交两组关键词 $(w_{0,1}^*,w_{0,2}^*,\ldots,w_{0,n}^*)$ 和 $(w_{1,1}^*,w_{1,2}^*,\ldots,w_{1,n}^*)$, 而挑战者随机选择一组关键词进行加密, 从而有 n 个挑战关键词密文.

类似对称加密方案, 如果 PAEKS 敌手也是自适应的, 能够选择并获取挑战关键词的密文, 则完全关键词密文不可区分性 (fully CI-security) 蕴含了完全多密文不可区分性 (fully MCI-security), 即下面的引理成立:

引理 6.7

如果一个 PAEKS 方案是完全关键词密文不可区分的, 则该方案也是完全多关键词密文不可区分的. ♡

笔记 早期的一些 PAEKS 方案的 CI 安全模型并不允许敌手查询挑战关键词的密文, 如 [341, 351]. 此时, 单关键词密文不可区分未必蕴含多关键词密文不可区分. 事实如此, 秦等在文献 [350, 352] 中指出早期的方案在多关键词密文不可区安全模型中并不安全.

在前面的安全模型中, 敌手允许访问非挑战接收者公钥加密的密文或者是检索非挑战发送者公钥加密密文的检索令牌, 这种情景也称为多用户环境. 2019 年, Noroozi 和 Eslami [351] 指出单用户环境下的 PAEKS 方案在多用户环境下并不一定安全. 事实如此, 早期的 PAEKS 方案在多用户环境下不一定能够保证关键词密文的不可区分性.

下面将已有的几种针对关键词密文不可区分性的代表性 PAEKS 安全模型总结于表 6.1, 其中 $b\in\{0,1\}$, $i\in\{1,\ldots,n\}$, 符号 "$\star$" 表示任意公钥或关键词. 从比较结果可以看出, 通过是否允许敌手访问其他用户公钥下的关键词密文或者关键词检索令牌, 是否允许访问挑战关键词的密文, 不同安全模型达到的应用环境有所不同. 在这四种模型中, QCZ+21 方案对敌手访问两个谕言机的限制最少, 安全性最高.

检索令牌不可区分性的定义类似关键词密文不可区分性, 不同之处在于挑战信息是一个检索令牌, 而敌手可以提交任意形式的公钥/关键词对 (pk,w) 到检索令牌谕言机, 但是不能询问挑战公钥/关键词 (pk_R,w_b^*) 的关键词密文. 否则, 敌手通过检索匹配算法直接打破方案的安全性. 下面给出 (fully) TI 安全性的形式化定义.

表 6.1: PAEKS 方案的密文不可区分安全模型对比

模型	密文不可区分安全性		适用环境
	关键词密文查询谕言机	检索令牌查询谕言机	
HL17 [341]	$pk = pk_R \wedge w \neq w_b^*$	$pk = pk_S \wedge w \neq w_b^*$	单用户、单密文
NE19 [351]	$(pk, w) \neq (pk_R, w_b^*)$	$(pk, w) \neq (pk_S, w_b^*)$	多用户、单密文
QCH+20 [350]	$pk = pk_R \wedge w \neq w_{b,i}^*$	$pk = pk_S \wedge w \neq w_{b,i}^*$	单用户、多密文
QCZ+21 [352]	$(pk, w) = (\star, \star)$	$(pk, w) \neq (pk_S, w_{b,i}^*)$	多用户、多密文

检索令牌不可区分安全性. 定义可搜索公钥认证加密方案敌手 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{TI}}(\kappa) = \left| \Pr \left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk_S, sk_S) \leftarrow \mathsf{KeyGen}_S(pp); \\ (pk_R, sk_R) \leftarrow \mathsf{KeyGen}_R(pp); \\ (w_0, w_1, state) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{encrypt}}, \mathcal{O}_{\mathsf{tokengen}}}(pp, pk_S, pk_R); \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ t^* \leftarrow \mathsf{TokenGen}(sk_R, pk_S, w_\beta); \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{encrypt}}, \mathcal{O}_{\mathsf{tokengen}}}(pp, pk_S, pk_R, state, t^*); \end{array} \right] - \frac{1}{2} \right|$$

在上述定义中, 敌手可以提交任意形如 (pk, w) 的询问到检索令牌谕言机, 但是不能提交形如 (pk_R, w_b^*) 的询问到关键词密文谕言机. 如果任意的 PPT 敌手 $\mathcal{A}$ 在上述定义中的优势函数均为可忽略函数, 则称可搜索公钥认证加密方案 PAEKS 是 (fully) TI 安全的.

笔记 在检索令牌不可区分性安全模型中, 完全检索令牌不可区分安全性适用于多用户、多检索令牌不可区分的应用环境. 在实际应用中, 一个 PAEKS 方案在实现多关键词密文不可区分性的同时, 可能无法同时满足多检索令牌的不可区分性. 此时, 要求敌手不能询问挑战关键词的检索令牌查询. 代表性的几个 TI 安全模型 [341, 351, 352] 都有这种限制. 此外, 文献 [341] 定义的安全模型仅适用于单用户环境. 尽管一些方案声称能够同时实现多关键词密文、多检索令牌的安全性, 但是在关键词密文和检索令牌查询上都有所限制, 并没有达到完全安全性, 甚至许多方案存在安全性问题. 能否同时实现多关键词密文和多检索陷门的完全安全性值得进一步研究.

下面介绍一种基于双线性配对群的 PAEKS 方案. 该方案由秦等 [352] 提出, 也是第一个在多用户环境下满足完全关键词密文不可区分安全性和检索令牌不可区分安全性的可搜索公钥加密方案.

构造 6.4 (基于双线性映射的 PAEKS 方案)

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{GenBLGroup}(1^\kappa)$ 生成一个对称双线性映射 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$. 选择 3 个密码学哈希函数 $\mathsf{H}_1 : \{0,1\}^* \to G$, $\mathsf{H}_2 : G \to \{0,1\}^{\log q}$ 和 $\mathsf{H}_3 : G \to \{0,1\}^{hLen}$, 其中 $hLen$ 是密码哈希函数如 SHA-1 输出的长度. 输出公开参数为 $pp = (\mathbb{G}, \mathbb{G}_T, q, g, e\mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3)$.
- $\mathsf{KeyGen}_S(pp)$: 随机选择 $u \xleftarrow{\mathrm{R}} \mathbb{Z}_q$, 计算并输出数据发送者的公钥 $pk_S = g^u$ 和私钥 $sk_S = u$.
- $\mathsf{KeyGen}_R(pp)$: 随机选择 $x, v \xleftarrow{\mathrm{R}} \mathbb{Z}_q$, 计算并输出数据接收者的公钥 $pk_R = (g^x, g^v)$ 和私钥 $sk_R = (x, v)$.
- $\mathsf{Encrypt}(sk_S, pk_R, w)$: 数据发送者随机选择 $r \xleftarrow{\mathrm{R}} \mathbb{Z}_q$, 计算 $A = g^r$, $B = \mathsf{H}_2(e(h^r, g^x))$, 其中 $h = \mathsf{H}_1(w||pk_s||pk_R||k)$, $k = \mathsf{H}_3(g^{vu})$. 输出关键词 w 的密文 $c_w = (A, B)$.
- $\mathsf{TokenGen}(sk_R, pk_S, w)$: 数据接收者计算并输出关键词 w 的检索令牌 $t_w = h^x$, 其中 $h = \mathsf{H}_1(w||pk_s||pk_R||k)$, $k = \mathsf{H}_3(g^{uv})$.
- $0/1 \leftarrow \mathsf{Test}(pk_S, pk_R, t_w, c_{w'})$: 对于关键词 w' 的密文 $c_{w'} = (A, B)$, 关键词 w 的检索令牌 t_w, 检索服务器判断 $\mathsf{H}_2(e(t_w, A)) \stackrel{?}{=} B$ 是否成立. 若成立, 则输出 1; 否则, 输出 0.

方案的正确性可以直接得到验证, 方案的安全性分别由定理 6.3和定理 6.4保证. 在分析方案的安全性之前, 先介绍安全性证明依赖的两个困难问题: BDH 问题和 ODH 问题. 其中, BDH 问题是标准的计算双线性配对 Diffie-Hellman 问题, ODH 问题含判定性 ODH 问题 (decisional oracle diffie-hellman problems, DODH) [353] 和计算性 ODH 问题 (computational oracle diffie-hellman problems, CODH) [352].

令 $\mathbb{G}$ 是一个素数阶循环群, q 是群的阶, g 是群的一个随机生成元. 给定 g^u, g^v 和谕言机 $\mathcal{O}_v(\cdot)$, 其中谕言机输入 $X \in \mathbb{G}$ 输出 $\mathcal{H}_v(X) = H(X^v)$, 则 DODH 问题的目标是区分 $H(g^{uv})$ 和一个随机比特串 $k \in \{0,1\}^{hLen}$, 其中 H 是一个密码学哈希函数, 值域为 $\{0,1\}^{hLen}$. 根据文献 [353], 只要敌手不询问谕言机 $\mathcal{H}_v$ 在元素 g^u 上的值, 则 DODH 是困难的.

定义 6.5 (DODH 假设)

令 $\mathsf{H}: \{0,1\}^* \to \{0,1\}^{hLen}$ 是一个密码学哈希函数. 对于任意 PPT 敌手 $\mathcal{A}$, 如果下面的优势函数关于安全参数 κ 是可忽略的, 则称 DODH 假设成立.

$$\mathsf{Adv}^{\mathrm{DODH}}_{\mathcal{A}}(\kappa) \quad = \quad \left|\Pr[\mathcal{A}^{\mathcal{O}_v(\cdot)}(g^u, g^v, \mathsf{H}(g^{uv})) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_v}(g^u, g^v, k) = 1]\right| \leq \mathsf{negl}(\kappa)$$

其中 $u, v \xleftarrow{R} \mathbb{Z}_q$, $k \xleftarrow{R} \{0,1\}^{hLen}$ 和 $\mathcal{O}_v(X) = \mathsf{H}(X^v)$, 且 $\mathcal{A}$ 不能查询 g^u 的结果 $\mathsf{H}_v(g^u)$. ♣

与区分 $\mathsf{H}(g^{uv})$ 和一个随机比特串相反, 秦等提出的 CODH 问题目标是计算哈希值 $\mathsf{H}(g^{uv})$. 给定 g^u 和 g^v, 在 CODH 问题中, 敌手除了可以查询谕言机 $\mathcal{O}_v(X) = \mathsf{H}(X^v)$, 还可以查询谕言机 $\mathcal{O}_u(X) = \mathsf{H}(X^u)$. 只要敌手不查询谕言机 $\mathcal{O}_u(g^v)$ 或 $\mathcal{O}_v(g^u)$, 则 CODH 问题被认为也是困难的.

定义 6.6 (CODH 假设)

令 $\mathsf{H}: \{0,1\}^* \to \{0,1\}^{hLen}$ 是一个密码学哈希函数. 对于任意 PPT 敌手 $\mathcal{A}$, 如果下面的优势函数关于安全参数 κ 是可忽略的, 则称 CODH 假设成立.

$$\mathsf{Adv}^{\mathrm{CODH}}_{\mathcal{A}}(\kappa) \quad = \quad \Pr[\mathcal{A}^{\mathcal{O}_u, \mathcal{O}_v}(g^u, g^v, \mathsf{H}) = \mathsf{H}(g^{uv})]$$

其中 $u, v \xleftarrow{R} \mathbb{Z}_q$, $\mathcal{O}_u(X) = \mathsf{H}(X^u)$ 和 $\mathcal{O}_v(X) = \mathsf{H}(X^v)$, 且 $\mathcal{A}$ 不能查询 g^v 和 g^u 的谕言机 $\mathcal{O}_u(g^v)$ 和 $\mathcal{O}_v(g^u)$. ♣

笔记 一般来地, 一个问题的计算性版本肯定比判定性版本要困难, 至少不会更容易, 例如 DDH 问题与 CDH 问题, 或者 DBDH 问题与 CBDH 问题等. 对于 DODH 问题与 CODH 问题, 细心的读者可能会发现, CODH 问题允许敌手访问的谕言机要比 DODH 问题允许敌手访问的谕言机多. 因此, CODH 问题比 DODH 问题困难的结论并不是那么直接. 尽管如此, Qin 等证明 CODH 问题并不比 DODH 问题容易解决, 见引理 6.8. 因此, 若 DODH 问题是困难的, 则 CODH 问题一定是困难的.

引理 6.8

令 $\mathbb{G}$ 是一个阶为素数 q 的循环群, g 是 $\mathbb{G}$ 的一个随机生成元, H 一个密码学哈希函数, 则有

$$\mathsf{Adv}^{\mathrm{CODH}}_{\mathcal{A}}(\kappa) \leq \mathsf{Adv}^{\mathrm{DODH}}_{\mathcal{B}}(\kappa) + \frac{1}{2^{hLen}}.$$

♡

定理 6.3

如果 BDH 假设成立, 则构造 6.4 中的 PAEKS 在随机谕言机模型下满足完全关键词密文不可区分性. ♡

定理 6.3的证明思路类似 Boneh 等的 PEKS 方案的安全性证明, 核心思路是利用随机谕言机去构造 "看起来随机" 但是在回答敌手的不同询问时都可以使用的哈希值, 同时将 BDH 问题实例嵌入到挑战关键词密文中.

证明 通过归约的方式组织证明. 下面描述如何利用 (算法)$\mathcal{A}$ 作为子程序, 构造一个攻击 BDH 问题的算法 $\mathcal{B}$. 令 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$ 是一个双线性映射. $\mathcal{B}$ 的输入是 BDH 问题的一个实例 $(g, X = g^x, Y = g^y, Z = g^z) \in \mathbb{G}^4$, 目标是计算 $T = e(g,g)^{xyz}$. $\mathcal{B}$ 按以下方式模拟 $\mathcal{A}$ 在 CI 游戏中的视图.

初始化: $\mathcal{B}$ 选择 3 个密码学哈希函数 $\mathsf{H}_1, \mathsf{H}_2$ 和 H_3. 令 Q_{H_2} 和 Q_T 分别是 $\mathcal{A}$ 访问 H_2 哈希谕言机和检索令牌谕言机的最大次数. 设置公开参数为 $pp = (\mathbb{G}, \mathbb{G}_T, q, g, e\mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3)$. 接下来, 选择一个随机元素 $u \in \mathbb{Z}_q$, 设置 $pk^*_S = g^u$ 作为数据发送者的公钥, $sk^*_S = u$ 作为相应的私钥. 类似地, 选择一个随机元素 $v \in \mathbb{Z}_q$, 设置 $pk^*_R = (X, g^v)$ 作为接收者的公钥, $sk^*_R = (x, v)$ 作为相应的私钥, 其中 x 对于 $\mathcal{B}$ 是未知的. 将公钥 pk^*_S 和 pk^*_R, 以及公开参数 pp 发送给敌手 $\mathcal{A}$.

哈希询问: 在证明中, H_3 看作是标准的密码学哈希函数, 输入 I, $\mathcal{A}$ 和 $\mathcal{B}$ 都可以自行计算相应的哈希值 $\mathsf{H}_3(I)$. 而 H_1 和 H_2 被看作是随机谕言机, 工作过程如下:

- H_1-询问: $\mathcal{B}$ 维护一个列表 $\langle I_i, a_i, c_i, h_i\rangle$, 称为 H_1-列表, 其中 $I_i = w_i||pk_S||pk_R||k_i$. 当 $\mathcal{A}$ 提交询问 $I_i = w_i||pk_S||pk_R||k_i$ 时, $\mathcal{B}$ 按以下方式回答:
  1. $\mathcal{B}$ 检查 H_1 列表中是否存在包含 I_i 的元素. 如果存在, 则返回相应的 h_i 给 $\mathcal{A}$; 否则, 选择一个随机比特 $c_i \in \{0,1\}$ 满足 $\Pr[c_i = 0] = \frac{1}{1+Q_T}$.
  2. 选择一个随机元素 $a_i \in \mathbb{Z}_q$. 如果 $c_i = 0$, 设置 $h_i = Y \cdot g^{a_i}$; 如果 $c_i = 1$, 设置 $h_i = g^{a_i}$.
  3. $\mathcal{B}$ 将 h_i 返回给 $\mathcal{A}$ 作为 I_i 的哈希值 $\mathsf{H}_1(I_i)$, 并将 $\langle I_i, a_i, c_i, h_i\rangle$ 添加到 H_1-列表中.
- H_2-询问: $\mathcal{B}$ 维护一个形如 $\langle t_i, V_i\rangle$ 的 H_2-列表. 当 $\mathcal{A}$ 提交询问 $t_i \in \mathbb{G}_T$ 时, $\mathcal{B}$ 首先检查列表中是否存在元素 t_i. 如果存在, 则 $\mathcal{B}$ 返回相应的值 V_i. 否则, $\mathcal{B}$ 选择一个随机值 $V_i \in \{0,1\}^{\log q}$. 最后, $\mathcal{B}$ 将 V_i 作为哈希值 $\mathsf{H}_2(t_i)$ 返回给 $\mathcal{A}$, 并将 $\langle t_i, V_i\rangle$ 添加到 H_2-列表中.

关键词密文询问: 当 $\mathcal{A}$ 提交关键词密文询问 $(pk_R = (pk_{R,1}, pk_{R,2}), w_i) \in \mathbb{G}^2 \times \{0,1\}^*$ 时, $\mathcal{B}$ 首先计算 $k_i = \mathsf{H}_3(pk_{R,2}^u)$. 然后, 通过 H_1-哈希询问的方式获取 $I_i = w_i||pk_S^*||pk_R||k_i$ 的哈希值 $\mathsf{H}_1(I_i) = h_i$. 接下来, 选择一个随机元素 $r \in \mathbb{Z}_q$, 计算 $A = g^r$ 和 $t_i = e(h_i, pk_{R,1})^r$. $\mathcal{B}$ 通过 H_2-哈希询问的方式获取 V_i 的哈希值 $\mathsf{H}_2(t_i) = V_i$. 最后, $\mathcal{B}$ 设置 $B = V_i$, 并将关键词密文 $c_{w_i} = (A, B)$ 发送给 $\mathcal{A}$.

检索令牌询问: 当 $\mathcal{A}$ 提交检索令牌询问 $(pk_S, w_i) \in \mathbb{G} \times \{0,1\}^*$ 时, $\mathcal{B}$ 首先计算 $k_i = \mathsf{H}_3(pk_S^v)$, 通过 H_1-哈希询问的方式获取 $I_i = w_i||pk_S||pk_R^*||k_i$ 的哈希值 $\mathsf{H}_1(I_i) = h_i$, 及相应的元素 $\langle I_i, a_i, c_i, h_i\rangle$. 如果 $c_i = 0$, $\mathcal{B}$ 终止游戏. 否则, $h_i = g^{a_i}$, 从而 $\mathcal{B}$ 可以计算 $t_{w_i} = X^{a_i}$ $(= \mathsf{H}_1(I_i)^x)$. 最终, $\mathcal{B}$ 将检索令牌 t_{w_i} 发送给 $\mathcal{A}$.

注记 6.5

在回答检索令牌询问时, 模拟者 $\mathcal{B}$ 希望 $c_i = 1$, 此时可以知道 h_i 的离散对数, 从而可以计算相应的检索令牌.

挑战: 当 $\mathcal{A}$ 提交两个挑战关键词 w_0^* 和 w_1^* 时, $\mathcal{B}$ 首先选择一个随机比特 $b \in \{0,1\}$. 然后, 计算 $k^* = \mathsf{H}_3(g^{uv})$, 通过 H_1-哈希询问的方式获取 $I^* = w_b^*||pk_S^*||pk_R^*||k^*$ 的哈希值 $\mathsf{H}_1(I^*) = h^*$, 及相应的元素 $\langle I^*, a^*, c^*, h^*\rangle$. 如果 $c^* = 1$, $\mathcal{B}$ 终止游戏. 否则, $c = 0$ 且 $h^* = Yg^{a^*}$. $\mathcal{B}$ 选择一个随机元素 $V^* \in \{0,1\}^{\log q}$, 将挑战关键词密文 $c^* = (Z, V^*)$ 返回给 $\mathcal{A}$. 这隐含地定义了 $\mathsf{H}_2(t^*) = V^*$ 和 $t^* = e(\mathsf{H}_1(I^*), X)^z = e(g^y g^{a^*}, g^x)^z = e(g,g)^{xz(y+a^*)}$.

注记 6.6

在模拟挑战密文时, 模拟者 $\mathcal{B}$ 希望 $c_i = 0$, 此时可以将 BDH 问题实例的解嵌入到挑战密文中.

更多询问: $\mathcal{A}$ 可以继续进行关键词密文和检索陷门询问, 所受限制是 $\mathcal{A}$ 不能询问检索令牌谕言机关于 (pk_S^*, w_0^*) 和 (pk_S^*, w_1^*) 的检索陷门.

猜测: 最终, $\mathcal{A}$ 输出一个比特 b' 作为对挑战关键词密文中 c^* 随机比特 b 的猜测结果. 同时, $\mathcal{B}$ 从 H_2-列表中选择一个随机元素 $\langle t, V\rangle$, 输出 $t/e(X,Z)^{a^*}$ 作为对 $e(g,g)^{xyz}$ 的猜测结果.

至此, 算法 $\mathcal{B}$ 描述完毕. 下面, 分析 $\mathcal{B}$ 成功的概率. 令 $I_0 = w_0^*||pk_S^*||pk_R^*||k^*$ 和 $I_1 = w_1^*||pk_S^*||pk_R^*||k^*$, 其中 $k^* = \mathsf{H}_3(g^{uv})$. 令 F 表示事件 "$\mathcal{A}$ 在游戏中查询了哈希值 $\mathsf{H}_2(e(\mathsf{H}_1(I_0), X)^z)$ 或者 $\mathsf{H}_2(e(\mathsf{H}_1(I_1), X)^z)$". 则有以下引理:

引理 6.9

如果敌手 $\mathcal{A}$ 以优势 ϵ 攻破 PAEKS 方案的完全关键词密文不可区分性, 则有 $\Pr[F] \geq 2\epsilon/(e(1+Q_T))$.

证明 令 E_1 和 E_2 分别表示事件 "$\mathcal{B}$ 在回答检索令牌询问阶段不终止游戏" 和 "$\mathcal{B}$ 在回答挑战关键词密文询问阶段不终止游戏". 对于 $\mathcal{A}$ 的第 i 次检索令牌查询 (pk_s, w_i), 存在元素组 $\langle I_i, a_i, c_i, h_i\rangle$ 满足 $I_i = w_i||pk_S||pk_R^*||\mathsf{H}_3(pk_S^v)$. 不管 $c_i = 0$ 还是 $c_i = 1$, h_i 都具有相同的分布且与 $\mathcal{A}$ 的视图独立, 所以 $\mathcal{B}$ 在回答每次检索令牌询问时终止游戏的概率最多为 $1/(1+Q_T)$. 由于 $\mathcal{A}$ 最多查询 Q_T 次检索陷门谕言机, 所以有 $\Pr[E_1] = (1 - 1/(1+Q_T))^{Q_T} \geq \frac{1}{e}$.

类似地, 在查询挑战关键词密文之前, $\mathcal{A}$ 的视图与 c^* 独立, 所以 $\Pr[E_2]=\Pr[c^*=0]=\frac{1}{1+Q_T}$.

由于 $\mathcal{A}$ 被禁止查询 (pk_S^*, w_0^*) 和 (pk_S^*, w_1^*) 的检索令牌, 所以事件 E_1 和 E_2 相互独立. 故有 $\Pr[E_1\wedge E_2]\geq \frac{1}{e\cdot(1+Q_T)}$.

令 E_3 表示事件 "在 $\mathcal{B}$ 不终止游戏的情况下, $\mathcal{A}$ 询问哈希值 $\mathsf{H}_2(e(\mathsf{H}_1(I_0^*),X)^z)$ 或 $\mathsf{H}_2(e(\mathsf{H}_1(I_1^*),X)^z)$". 显然, 如果 E_3 从不发生, 则 $\mathcal{A}$ 猜测 b 的优势为零. 由于 $\Pr[b'=b]=\Pr[b'=b|E_3]\cdot\Pr[E_3]+\Pr[b'=b|\overline{E_3}]\cdot\Pr[\overline{E_3}]$, 则有

$$\begin{aligned}
&\Pr[b'=b|\overline{E_3}]\cdot\Pr[\overline{E_3}]\leq\Pr[b'=b]\leq\Pr[E_3]+\frac{1}{2}\cdot\Pr[\overline{E_3}]\\
\Rightarrow\quad&\frac{1}{2}-\frac{1}{2}\cdot\Pr[E_3]\leq\Pr[b'=b]\leq\frac{1}{2}+\frac{1}{2}\cdot\Pr[E_3]\\
\Rightarrow\quad&\left|\Pr[b'=b]-\frac{1}{2}\right|\leq\frac{1}{2}\cdot\Pr[E_3].
\end{aligned}$$

由此可得, $\Pr[E_3]\geq 2\epsilon$. 当 $\mathcal{B}$ 不终止游戏时, 上述游戏完美地模拟了完全关键词密文不可区分安全性游戏, 则有 $\Pr[F|(E_1\wedge E_2)]=\Pr[E_3]$. 因此,

$$\begin{aligned}
\Pr[F]&=\Pr[F|(E_1\wedge E_2)]\cdot\Pr[E_1\wedge E_2]+\Pr[F|\overline{E_1\wedge E_2}]\cdot\Pr[\overline{E_1\wedge E_2}]\\
&\geq\Pr[E_3]\Pr[E_1\wedge E_2]\geq\frac{2\epsilon}{e(1+Q_T)}.
\end{aligned}$$

引理 6.9证毕! □

注意到事件 F 的发生意味着 H_2 列表包含满足 $t=e(\mathsf{H}_1(I_b),X)^z$ 的元素组 $\langle t,V\rangle$ 的概率至少为 $1/2$. 由于,

$$t=e(\mathsf{H}_1(I_b),X)^z=e(\mathsf{H}_1(I^*),X)^z=e(Yg^{a^*},X)^z,$$

所以 $e(g,g)^{xyz}=t/e(X,Z)^{a^*}$.

又由于 $\mathcal{B}$ 选择到正确元素组 $\langle t,V\rangle$ 的概率是 $1/Q_{\mathsf{H}_2}$, 所以 $\mathcal{B}$ 成功解决 BDH 问题的概率至少为 $\Pr[F]/(2Q_{\mathsf{H}_2})$, 即,

$$\frac{\Pr[F]}{2Q_{\mathsf{H}_2}}\geq\frac{\epsilon}{eQ_{\mathsf{H}_2}(1+Q_T)}.$$

定理 6.3证毕! □

定理 6.4

如果 CODH 假设成立, 则构造 6.4中 PAEKS 在随机谕言机模型下满足检索令牌不可区分性. ♡

证明 下面通过归约的方式组织证明. 首先描述如何利用 (算法)$\mathcal{A}$ 作为子程序, 构造一个攻击 CODH 问题的算法 $\mathcal{B}$. 令 $(\mathbb{G},\mathbb{G}_T,q,g,e)$ 是一个双线性映射. $\mathcal{B}$ 的输入是一个 CODH 问题实例 $(g,U=g^u,V=g^v,\mathsf{H}_3)$ 及两个谕言机 $\mathcal{O}_u(X)=\mathsf{H}_3(X^u)$ 和 $\mathcal{O}_v(Y)=\mathsf{H}_3(Y^v)$, 目标是计算 $\mathsf{H}_3(g^{uv})$. $\mathcal{B}$ 按以下方式模拟 $\mathcal{A}$ 在 TI 游戏中的视图.

初始化: $\mathcal{B}$ 选择 2 个密码学哈希函数 H_1 和 H_2. 令 Q_{H_1} 是 $\mathcal{A}$ 访问 H_1 哈希谕言机的最大次数设置公开参数为 $pp=(\mathbb{G},\mathbb{G}_T,q,g,e,\mathsf{H}_1,\mathsf{H}_2,\mathsf{H}_3)$. 然后, 设置 $pk_S^*=U$ 作为数据发送者的公钥, $sk_S^*=u$ 作为相应的私钥 (这里的 u 对于 $\mathcal{B}$ 未知). $\mathcal{B}$ 选择一个随机元素 $x\in\mathbb{Z}_q$, 设置 $pk_R^*=(g^x,V)$ 作为数据接收者的公钥, $sk_R^*=(x,v)$ 作为相应的私钥 (这里的 x 对于 $\mathcal{B}$ 已知). $\mathcal{B}$ 将公钥 pk_S^* 和 pk_R^*, 以及公开参数 pp 发送给 $\mathcal{A}$.

哈希询问: 在证明中, 哈希函数 H_2 和 H_3 看作是标准的密码哈希函数, 而 H_1 看作是随机谕言机, 工作如下:

- H_1-询问: $\mathcal{B}$ 维护一个形如 $\langle I_i,h_i\rangle$ 的 H_1 列表, 其中 $I_i=w_i||pk_S||pk_R||k_i$. 当 $\mathcal{A}$ 提交查询 $I_i=w_i||pk_S||pk_R||k_i$ 时, $\mathcal{B}$ 随机选择 $h_i\in G$ 作为回答结果, 并将 $\langle I_i,h_i\rangle$ 添加到 H_1 列表中.
  $\mathcal{B}$ 自己也可能查询特殊元素 I_i 的 H_1 谕言机, 其中 k_i 用特殊符号 "$\star$" 代替, 表示 CODH 问题的未知解 $\mathsf{H}_3(g^{uv})$. 此时, $\mathcal{B}$ 选择一个随机元素 $h_i\in G$ 并设置 $\mathsf{H}_1(I_i)=h_i$.

关键词密文询问: 当 $\mathcal{A}$ 询问 $(pk_R=(pk_{R,1},pk_{R,2}),w_i)\in\mathbb{G}^2\times\{0,1\}^*$ 的关键词密文时, 如果 $pk_{R,2}\neq V$, $\mathcal{B}$ 通过查询谕言机 $\mathcal{O}_u(pk_{R,2})$ 获取 $k_i=\mathsf{H}_3(pk_{R,2}^u)$. 否则, $\mathcal{B}$ 设置 $k_i=\star$. 然后, $\mathcal{B}$ 通过查询谕言机 H_1 获取 $I_i=w_i||pk_S^*||pk_R||k_i$ 的哈希值 $\mathsf{H}_1(I_i)=h_i$. 接下来, 随机选择 $r\xleftarrow{R}\mathbb{Z}_q$, 计算 $A=g^r$ 和 $B=\mathsf{H}_2(e(h_i,pk_{R,1})^r)$. 最后, $\mathcal{B}$ 将密文 $c_{w_i}=(A,B)$ 返回给 $\mathcal{A}$.

检索令牌询问: 当 $\mathcal{A}$ 询问 $(pk_S, w_i) \in G \times \{0,1\}^*$ 的检索令牌时, 如果 $pk_S \neq U$, $\mathcal{B}$ 通过查询谕言机 $\mathcal{O}_v(pk_S)$ 以获取 $k_i = \mathsf{H}_3(pk_S^v)$. 否则, $\mathcal{B}$ 设置 $k_i = \star$. 然后, $\mathcal{B}$ 通过查询谕言机 H_1 以获取 $I_i = w_i||pk_S||pk_R^*||k_i$ 的哈希值 $\mathsf{H}_1(I_i) = h_i$. $\mathcal{B}$ 计算检索令牌 $t_{w_i} = h_i^x\ (= \mathsf{H}_1(I_i)^x)$ 并返回给敌手 $\mathcal{A}$.

挑战: 当 $\mathcal{A}$ 提交两个挑战关键词 w_0^* 和 w_1^* 时, $\mathcal{B}$ 首先挑选一个随机比特 $b \in \{0,1\}$ 并通过查询谕言机 H_1 以获取 $I^* = w_b^*||pk_S^*||pk_R^*||\star$ 的哈希值 $\mathsf{H}_1(I^*) = h^*$. $\mathcal{B}$ 计算挑战检索令牌 $t_{w_b^*} = (h^*)^x$ 并返回给敌手 $\mathcal{A}$.

更多询问: $\mathcal{A}$ 可以继续进行关键词密文和检索令牌询问, 所受限制是 $\mathcal{A}$ 不能询问 (pk_R^*, w_0^*) 和 (pk_R^*, w_1^*) 的关键词密文以及 (pk_S^*, w_0^*) 和 (pk_S^*, w_1^*) 的检索令牌.

猜测: 最终, $\mathcal{A}$ 输出一个比特 b' 作为对挑战检索令牌 $t_{w_b^*}$ 中随机比特 b 的猜测结果. 同时, $\mathcal{B}$ 从 (除去特殊形式询问的)H_1 列表中随机选择一个元素组 $\langle I = w||pk_S||pk_R||k, h\rangle$ 将其中的 k 作为 CODH 问题的解 $\mathsf{H}_3(g^{uv})$.

至此, 算法 $\mathcal{B}$ 模拟的 TI 游戏描述完毕. 下面分析 $\mathcal{B}$ 成功的概率.

令 $I_0 = w_0^*||pk_S^*||pk_R^*||k^*$ 和 $I_1 = w_1^*||pk_S^*||pk_R^*||k^*$, 其中 $k^* = \mathsf{H}_3(g^{uv})$. 由于 $\mathcal{A}$ 在询问挑战检索令牌前, 不能查询关键词密文 $\mathsf{Encrypt}(sk_S^*, pk_R^*, w_i^*)$ 和检索令牌 $\mathsf{TokenGen}(sk_R^*, pk_S^*, w_i^*)$ $(i = 0, 1)$, 所以哈希值 $\mathsf{H}_1(I_i)$ 与 $\mathcal{A}$ 的视图独立. 此外, 不管 $I^* = I_0$ 还是 $I^* = I_1$, 相应的哈希值具有相同的分布. 因此, 如果 $\mathcal{A}$ 从未询问 $\mathsf{H}_1(I_0)$ 或 $\mathsf{H}_1(I_1)$, 则敌手区分挑战检索令牌的优势为零. 令 E 表示事件 "$\mathcal{A}$ 查询过 $\mathsf{H}_1(I_0)$ 或 $\mathsf{H}_1(I_1)$". 下面证明, 如果 $\mathcal{A}$ 以不可忽略的优势 ϵ 区分挑战检索令牌, 则事件 E 发生的概率也是不可忽略的. 这是因为,

$$\Pr[b' = b] = \Pr[b' = b|E] \cdot \Pr[E] + \Pr[b' = b|\overline{E}] \cdot \Pr[\overline{E}]$$

且

$$\begin{aligned} & \Pr[b' = b|\overline{E}] \cdot \Pr[\overline{E}] \leq \Pr[b' = b] \leq \Pr[E] + \frac{1}{2} \cdot \Pr[\overline{E}] \\ \Rightarrow \quad & \frac{1}{2} - \frac{1}{2} \cdot \Pr[E] \leq \Pr[b' = b] \leq \frac{1}{2} + \frac{1}{2} \cdot \Pr[E] \\ \Rightarrow \quad & \left|\Pr[b' = b] - \frac{1}{2}\right| \leq \frac{1}{2} \cdot \Pr[E]. \end{aligned}$$

从而可得 $\Pr[E] \geq 2\epsilon$.

如果事件 E 发生, 则哈希列表 H_1 至少以 $1/2$ 概率存在形如 $I_b = w_b^*||pk_S^*||pk_R^*||\mathsf{H}_3(g^{uv})$ 的哈希询问. 因此, $\mathcal{B}$ 从 H_1 列表中随机选取到元素组 $\langle I_b, h^*\rangle$ 的概率至少为 $1/Q_{\mathsf{H}_1}$.

综上, $\mathcal{B}$ 找到 CODH 问题解的概率至少为 $\epsilon/Q_{\mathsf{H}_1}$. 定理 6.4得证! □

6.2 可托管公钥加密

六耳不同谋. 且去, 来日来.

— 宋 · 释普济《五灯会元》

在标准的公钥加密中, 密文所对应的明文只有发送方和预定的接收方可以解密. 该设定在实际应用中存在以下两个问题:

- 接收方若丢失私钥则无法再打开密文.
- 第三方若想解密, 仅能通过非技术手段强制接收方打开密文.

可托管公钥加密 (escrow PKE) 正是为了解决上述问题所提出的公钥加密扩展. 与公钥加密相比, 可托管公钥加密中增设了密钥托管中心 (key escrow center) 这一实体, 其拥有托管解密私钥, 起到万能钥匙的作用, 可正确解密任意公钥加密的密文. 在实际应用中, 密钥托管中心既可以向丢失私钥的用户提供解密服务, 也可以对所有密文实施穿透式监管审计. 陈等在 [354] 中给出了可托管公钥加密的严格定义.

定义 6.7 (可托管公钥加密)

可托管公钥加密方案包含 5 个 PPT 算法: Setup, KeyGen, Encrypt, Decrypt 和 $\mathsf{Decrypt}'$. 其中的 KeyGen, Enc 和 Decrypt 算法与标准的公钥加密完全相同, 不同的是算法 Setup 将额外输出托管解密私钥, 算法 $\mathsf{Decrypt}'$ 可使用托管解密私钥解密任意密文.

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 κ 为输入, 输出系统公开参数 pp 和托管解密私钥 edk. 该算法由密钥托管中心运行, 所有用户均可访问 pp, edk 由密钥托管中心秘密保存.
- $\mathsf{Decrypt}'(edk, c)$: 以托管解密私钥 edk 和密文 c 为输入, 输出明文 m 或 $\perp$ 表示解密失败.

♣

注记 6.7

在可托管公钥加密中, 密钥托管中心在解密时需要知晓密文的接收方公钥, 因此我们默认密文中总是显式或隐式的包含接收方的公钥信息.

♠

可托管公钥加密需要满足以下的正确性和安全性:

正确性. 对于 $\forall m \in M$, 我们有:

$$\Pr[\mathsf{Decrypt}(sk, c) = m = \mathsf{Decrypt}'(edk, c)] \geq 1 - \mathsf{negl}(\kappa) \tag{6.10}$$

其中公式 (6.10)的概率建立在算法 $\mathsf{Setup}(1^\kappa) \to pp$、$\mathsf{KeyGen}(pp) \to (pk, sk)$ 和 $\mathsf{Encrypt}(pk, m) \to c$ 的随机带上.

一致性. 正确性仅保证了当密文由发送方诚实生成时, 接收方和密钥托管中心的解密结果一致. 在可托管公钥加密的应用场景中, 发送方存在非诚实生成密文的动机 (如规避监管). 因此, 除了正确性, 可托管公钥加密还需要考虑一致性, 以确保即使对于非法生成的密文, 接收方和密钥托管中心的解密结果仍然一致. 为了精确定义一致性, 我们首先定义由公钥索引的一族 $\mathcal{NP}$ 语言, 即 $L_{pk} = \{c \mid \exists m, r \text{ s.t. } c = \mathsf{Encrypt}(pk, m; r)\}$, 表征的是由 pk 加密的所有合法密文集合. 以下给出一致性的严格定义. 令 $\mathcal{A}$ 是针对一致性的敌手, 定义其优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}} = \Pr\left[\begin{array}{c|c} c \notin L_{pk} \wedge & (pp, edk) \leftarrow \mathsf{Setup}(1^\kappa); \\ \mathsf{Decrypt}(sk, c) \neq \mathsf{Decrypt}'(edk, c) & : \ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ & c \leftarrow \mathcal{A}(pp, pk); \end{array}\right].$$

我们称可托管公钥加密方案在计算意义 (resp. 统计意义) 下是一致的当且仅当任意 PPT(resp. unbounded) 的敌手在一致性试验中的优势均是可忽略的.

安全性. 令 $\mathcal{A}$ 是针对安全性的敌手, 定义其优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}} = \left| \Pr \left[\beta = \beta' : \begin{array}{l} (pp, edk) \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ (m_0, m_1) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{decrypt}}}(pp, pk); \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}, c^* \leftarrow \mathsf{Enc}(pk, m_\beta); \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{decrypt}}}(pp, pk, c^*); \end{array} \right] - \frac{1}{2} \right|.$$

这里 $\mathcal{O}_{\mathsf{decrypt}}$ 是解密谕言机. $\mathcal{A}$ 可向 $\mathcal{O}_{\mathsf{decrypt}}$ 发起多项式次询问, 惟一的限制是不得在阶段 2 询问挑战密文 c^* 的解密结果. 可托管公钥加密方案是 IND-CCA 安全的当且仅当任意 PPT 的在安全性试验中的优势均是可忽略的. IND-CCA1 或 IND-CPA 安全可以通过仅允许 $\mathcal{A}$ 在第 1 阶段访问 $\mathcal{O}_{\mathsf{decrypt}}$ 或完全禁止 $\mathcal{A}$ 访问 $\mathcal{O}_{\mathsf{decrypt}}$ 进行类比的定义.

早期的可托管公钥加密构造存在种种缺陷, 或依赖抗篡改的物理硬件, 或需要公钥和私钥之间存在陷门单向关系. 以下介绍可托管公钥加密的两个通用构造.

6.2.1 基于公钥加密和非交互式零知识证明的构造

表面上, 广播加密 (broadcast encryption) 似乎就平凡的蕴含了可托管公钥加密, 即令发送方在生成密文时设定接收方公钥集合包含真实接收方和密钥托管中心的公钥. 然而, 由于广播加密总是假设发送方诚实生成密文, 因此无法保证所得构造的一致性.

以下, 我们展示如何借助非交互式零知识证明将任何公钥加密方案编译成可托管公钥加密方案. 构造的思路是密钥托管中心在创建加密系统时自行生成一个密钥对 (pk_γ, sk_γ), 将其中 pk_γ 包含进系统公开参数, 使用私钥 sk_γ 作为托管解密私钥. 发送方在向公钥为 pk 的接收方传输明文 m 时, 将分别使用公钥 pk 和 pk_γ 对 m 独立加密两次, 再使用非交互式零知识证明生成加密的一致性证明. 解密密文时, 接收方和密钥托管中心首先验证零知识证明的有效性, 验证通过后再用各自的私钥进行解密. 完整的构造如下:

构造 6.5 (基于 NIZK 和 PKE 的构造)

构造所需的组件是:

- 公钥加密方案 PKE
- 非交互式零知识证明协议 NIZK

构造可托管公钥加密如下:

- $\mathsf{Setup}(1^\kappa)$: 运行 $pp_{\mathrm{pke}} \leftarrow \mathsf{PKE.Setup}(1^\kappa)$, $(pk_\gamma, sk_\gamma) \leftarrow \mathsf{PKE.KeyGen}(pp_{\mathrm{pke}})$, $pp_{\mathrm{nizk}} \leftarrow \mathsf{NIZK.Setup}(1^\kappa)$, 生成 $crs \leftarrow \mathsf{NIZK.CRSGen}(pp_{\mathrm{nizk}})$, 输出公开参数 $pp = (pp_{\mathrm{pke}}, pp_{\mathrm{nizk}}, crs, pk_\gamma)$ 和托管解密私钥 $edk = sk_\gamma$.
- $\mathsf{KeyGen}(pp)$: 以 $pp = (pp_{\mathrm{pke}}, pp_{\mathrm{nizk}}, crs, epk)$ 为输入, 输出 $(pk, dk) \leftarrow \mathsf{PKE.KeyGen}(pp_{\mathrm{pke}})$.
- $\mathsf{Encrypt}(pk, m)$: 随机并独立选取两个随机数 r_1 and r_2, 计算 $c_1 \leftarrow \mathsf{PKE.Encrypt}(pk, m; r_1)$ 和 $c_2 \leftarrow \mathsf{PKE.Encrypt}(pk_\gamma, m; r_2)$, 生成 $\pi \leftarrow \mathsf{NIZK.Prove}(crs, (pk, c_1, c_2), (r_1, r_2, m))$, 输出密文 $c = (pk, c_1, c_2, \pi)$. 这里, π 证明了 (c_1, c_2) 是使用公钥 pk 和 pk_γ 对同一明文加密的结果, 即 $(pk, c_1, c_2) \in L_{\mathrm{consistency}}$, 其中 $L_{\mathrm{consistency}}$ is defined as below:

$$L_{\mathrm{consistency}} = \{(pk, c_1, c_2) \mid \exists m, r_1, r_2 \text{ s.t.}$$
$$c_1 = \mathsf{PKE.Encrypt}(pk, m; r_1) \wedge c_2 = \mathsf{PKE.Encrypt}(pk_\gamma, m; r_2)\}$$

- $\mathsf{Decrypt}(sk, c)$: 以私钥 sk 和密文 $c = (pk, c_1, c_2, \pi)$ 为输入, 首先运行 $\mathsf{NIZK.Verify}(crs, (pk, c_1, c_2), \pi)$ 检验证明的有效性; 如果检验失败则返回 $\perp$, 否则输出 $m \leftarrow \mathsf{PKE.Decrypt}(sk, c_1)$.
- $\mathsf{Decrypt}'(edk, c)$: 以托管解密私钥 $edk = sk_\gamma$ 和密文 $c = (pk, c_1, c_2, \pi)$ 为输入, 首先运行 $\mathsf{NIZK.Verify}(crs, (pk, c_1, c_2), \pi)$ 检验证明的有效性; 如果检验失败则返回 $\perp$, 否则输出 $m \leftarrow \mathsf{PKE.Decrypt}(sk_\gamma, c_2)$.

构造 6.5的正确性由 PKE 和 NIZK 的正确性保证, 一致性由 NIZK 的自适应合理性 (adaptive soundness) 保证, 安全性由以下定理保证:

定理 6.5

如果 PKE 是 IND-CPA 安全的并且 NIZK 是自适应安全 (resp. 模拟合理自适应安全) 的, 那么构造 6.5是 IND-CCA1 (resp. IND-CCA) 安全的.

证明 证明的过程与构造选择密文安全公钥加密方案的 Naor-Yung 双密钥加密范式 [9] 和 Sahai 范式 [24] 相似, 这里略去证明细节, 留给读者作为练习.

注记 6.8

构造 6.5中使用两个独立的随机数分别在接收方公钥和密钥托管中心公钥加密同一明文两次. 当底层 PKE 满足称为 "randomness fusion" 的温和性质时, 可重用随机数, 使用 twisted Naor-Yung 范式 [25] 代替标准的 Naor-Yung 范式, 同时提升构造的计算效率和通信效率.

笔记 构造 6.5与 Naor-Yung 双密钥加密范式在形式上完全一致. 在 Naor-Yung 双密钥加密范式中, 两个公钥均属于接收方, 零知识证明用于获得选择密文安全. 构造 6.5中, 一个公钥属于接收方, 另一个公钥属于密钥托管中心, 零知识证明用于确保密钥托管中心与接收方拥有相同的解密能力, 使得同一密文的解密视图始终相同. 之前可托管公钥加密方案构造 [355, 356, 357] 获得可托管功能的途径是每个用户在注册公钥时均需要向证书中心 (certificate authority, CA) 提交私钥可恢复证明, 而本构造在加密每个消息时生成密文合法性证明. 本构造的优势在于可与标准 CA 流程完全兼容, 且构造自动满足选择密文安全性.

6.2.2 基于三方非交互式密钥协商和对称加密的构造

本小节展示可托管公钥加密的另外一个通用构造, 使用 KEM+DEM 的公钥加密设计范式. 我们首先将密钥封装机制的定义延拓到可托管场景中, 得到可托管密钥封装机制.

定义 6.8 (可托管密钥封装机制)

可托管密钥封装机制包含 5 个 PPT 算法: Setup, KeyGen, Encaps, Decaps, Decaps′. 其中, 算法 KeyGen, Encaps 和 Decaps 与标准密钥封装机制的对应算法相同, 算法 Setup 将额外输出托管解封装私钥, 算法 Decaps′ 使用托管解封装私钥进行解封装.

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 输出系统公开参数 pp 和托管解封装私钥 edk. 该算法由密钥托管中心运行. 不失一般性, 假定 pp 包括对会话密钥空间 K 的描述.
- $\mathsf{Decaps}'(edk, c)$: 以托管解封装私钥 edk 和密文 c 为输入, 输出会话密钥 $k \in K$ 或 $\perp$ 表示解封装失败.

正确性. 我们要求公式 (6.11)成立,

$$\Pr[\mathsf{Decaps}(sk, c) = k = \mathsf{Decaps}'(edk, c)] \geq 1 - \mathsf{negl}(\kappa) \tag{6.11}$$

其中概率建立在算法 $\mathsf{Setup}(1^\kappa) \to (pp, edk)$, $\mathsf{KeyGen}(pp) \to (pk, sk)$ 和 $\mathsf{Encaps}(pk) \to (c, k)$ 的随机带上.

一致性. 可托管密钥封装机制的一致性要求即使对于非法生成的密文, 接收方和密钥托管中心的解封装结果仍然一致. 首先定义由 pk 索引的一族 $\mathcal{NP}$ 语言, 即 $L_{pk}^{\mathrm{kem}} = \{c \mid \exists r \text{ s.t. } (c, k) = \mathsf{Encaps}(pk; r)\}$, 表征 pk 所有合法封装的密文集合. 令 $\mathcal{A}$ 是针对一致性的敌手, 定义其优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}} = \Pr\left[\begin{array}{c} c \notin L_{pk}^{\mathrm{kem}} \wedge \\ \mathsf{Decap}(sk, c) \neq \mathsf{Decap}'(edk, c) \end{array} : \begin{array}{l} (pp, edk) \leftarrow \mathsf{Setup}(1^\lambda); \\ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ c \leftarrow \mathcal{A}(pp, pk); \end{array}\right].$$

我们称可托管密钥封装机制在计算意义 (resp. 统计意义) 下是一致的当且仅当任意 PPT(resp. unbounded) 的敌手在一致性试验中的优势均是可忽略的.

安全性. 令 $\mathcal{A}$ 是针对安全性的敌手, 定义其优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}} = \left| \Pr \left[\beta = \beta' : \begin{array}{l} (pp, edk) \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk, sk) \leftarrow \mathsf{KeyGen}(pp); \\ (c^*, k_0^*) \leftarrow \mathsf{Encaps}(pk), k_1^* \leftarrow K; \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{decaps}}}(pp, pk, c^*, k_\beta^*); \end{array} \right] - \frac{1}{2} \right|.$$

这里 $\mathcal{O}_{\mathsf{decaps}}$ 是解封装谕言机. $\mathcal{A}$ 可向 $\mathcal{O}_{\mathsf{decaps}}$ 发起多项式次询问, 惟一的限制是不得询问挑战密文 c^* 的解封装结果. 可托管密钥封装机制是 IND-CCA 安全的当且仅当任意 PPT 的在安全性试验中的优势均是可忽略的. IND-CPA 安全可以通过禁止 $\mathcal{A}$ 访问 $\mathcal{O}_{\mathsf{decaps}}$ 进行类比的定义.

以下展示如何基于可托管密钥封装机制和对称加密方案构造可托管公钥加密.

构造 6.6 (基于可托管密钥封装机制和对称加密方案的构造)

以可托管密钥封装机制和对称加密方案为底层组件, 构造可托管公钥加密如下:

- $\mathsf{Setup}(1^\kappa)$: 以安全参数 1^κ 为输入, 运行 $(pp_{\mathrm{kem}}, edk) \leftarrow \mathsf{KEM.Setup}(1^\kappa)$, $pp_{\mathrm{ske}} \leftarrow \mathsf{SKE.Setup}(1^\kappa)$, 输出系统公开参数 $pp = (pp_{\mathrm{kem}}, pp_{\mathrm{ske}})$ 和托管解密私钥 edk.
- $\mathsf{KeyGen}(pp)$: 以系统公开参数 $pp = (pp_{\mathrm{kem}}, pp_{\mathrm{ske}})$ 为输入, 输出 $(pk, sk) \leftarrow \mathsf{KEM.KeyGen}(pp_{\mathrm{kem}})$.
- $\mathsf{Encrypt}(pk, m)$: 计算 $(c_{\mathrm{kem}}, k) \leftarrow \mathsf{KEM.Encaps}(pk)$, $c_{\mathrm{ske}} \leftarrow \mathsf{SKE.Enc}(k, m)$, 输出密文 $c = (c_{\mathrm{kem}}, c_{\mathrm{ske}})$.
- $\mathsf{Decrypt}(sk, c)$: 以密文 $c = (c_{\mathrm{kem}}, c_{\mathrm{ske}})$ 为输入, 计算 $k \leftarrow \mathsf{KEM.Decaps}(sk, c_{\mathrm{ske}})$; 如果 $k = \perp$ 则输出 $\perp$, 否则输出 $m \leftarrow \mathsf{SKE.Decrypt}(k, c_{\mathrm{ske}})$.
- $\mathsf{Decrypt}'(edk, c)$: 以密文 $c = (c_{\mathrm{kem}}, c_{\mathrm{ske}})$ 为输入, 计算 $k \leftarrow \mathsf{KEM.Decaps}'(edk, c_{\mathrm{ske}})$; 如果 $k = \perp$ 则输出 $\perp$, 否则输出 $m \leftarrow \mathsf{SKE.Dec}(k, c_{\mathrm{ske}})$.

♣

上述构造的正确性由可托管密钥封装机制和对称加密的正确性保证. 以下分析构造的一致性. 首先定义由对称密钥 k 索引的一族 $\mathcal{NP}$ 语言, 即 $L_k^{\mathrm{ske}} = \{c \mid \exists m, r \text{ s.t. } c = \mathsf{Enc}(k, m; r)\}$. 自然的, 表征 pk 合法密文集合的语言 $L_{pk} = \{(c_{\mathrm{kem}}, c_{\mathrm{ske}}) \mid \exists m, r \text{ s.t. } (c_{\mathrm{kem}}, k) = \mathsf{KEM.Encaps}(pk; r) \wedge c_{\mathrm{ske}} = \mathsf{SKE.Encrypt}(k, m)\}$. 无论 $c_{\mathrm{kem}} \in L_{pk}^{\mathrm{kem}}$ 与否, 可托管密钥封装机制的一致性保证了解密方和密钥托管中心解封装结果的一致性, 进而保证了最终解密结果的一致性. 构造的正确性由以下定理保证:

定理 6.6

如果可托管密钥封装机制是 IND-CPA (resp. IND-CCA) 安全的并且堆成加密方案是 IND-CPA (resp. IND-CCA) 安全的, 那么上述构造是 IND-CPA (resp. IND-CCA) 安全的.

♡

证明 证明的过程与 KEM+DEM 的混合加密设计范式相似, 读者可作为练习自行完成.

以上构造指出可托管公钥加密方案的构造可归结为可托管密钥封装机制的构造. 那么如何构造可托管密钥封装机制呢? 我们在经典公钥加密方案章节中提到, ElGamal PKE 是基于 Diffie-Hellman 非交互密钥协商协议构造得出的. 该构造蕴含了从任意两方非交互式密钥协商协议出发到密钥封装机制的一个通用构造 [273], 即发送方首先运行非交互式密钥协商方案的密钥生成算法生成临时密钥对, 将公钥作为封装密文, 将临时公钥和接收方公钥对应的会话密钥作为封装密钥. 接收方在解封装时运行非交互式密钥协商方案的密钥协商算法即可. 基于以上思考, 我们可以将上述构造思想凝练为 "NIKE-in-the-head" 范式, 并且推广到三方情形用于构造可托管密钥封装机制. 构造的具体思路如下: (i) 密钥托管中心首先运行三方 NIKE 的密钥生成算法生成密钥对 (pk_γ, sk_γ), 将 pk_γ 纳入系统公开参数, 将 sk_γ 秘密保存作为托管解封装私钥; (ii) 发送方向公钥为 $pk = pk_\beta$ 的接收方传递消息时, 首先生成随机密钥对 (pk_α, sk_α), 再在头脑中运行三方 NIKE 协议, 计算 $\{pk_\alpha, pk_\beta, pk_\gamma\}$ 三方的会话密钥, 将 pk_α 作为封装密文, 将会话密钥作为封装的密钥. NIKE 的功能性确保了密钥托管中心和接收方可导出同样的会

话密钥, NIKE 的安全性保证了会话密钥在敌手视角中是伪随机的.

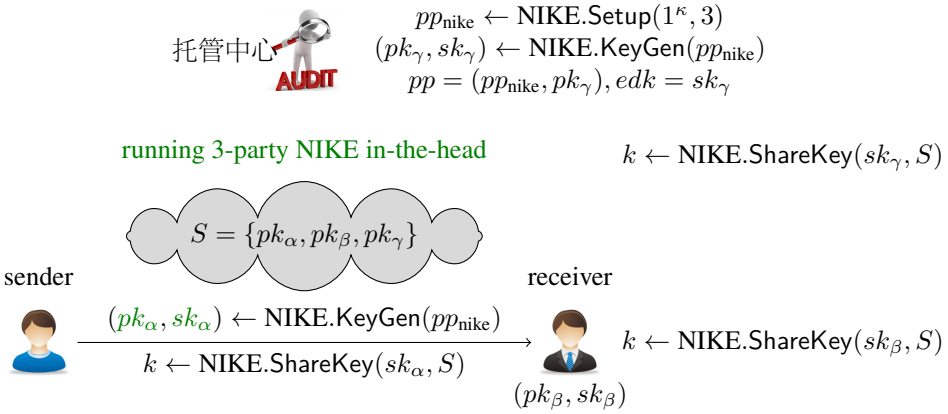


图 6.7: NIKE-in-the-head 构造思路

构造的思路如图 6.7所示, 细节如下:

构造 6.7 (基于 NIKE 的可托管密钥封装机制构造)

以三方 NIKE 协议为底层方案, 构造可托管密钥封装机制如下:

- $\text{Setup}(1^\kappa)$: 运行 $pp_{\text{nike}} \leftarrow \text{NIKE.Setup}(1^\kappa)$ 和 $(pk_\gamma, sk_\gamma) \leftarrow \text{NIKE.KeyGen}(pp_{\text{nike}})$, 输出系统公开参数 $pp = (pp_{\text{nike}}, pk_\gamma)$ 和托管解封装私钥 $edk = sk_\gamma$.
- $\text{KeyGen}(pp)$: 以系统公开参数 $pp = (pp_{\text{nike}}, pk_\gamma)$ 为输入, 运行 $\text{NIKE.KeyGen}(pp_{\text{nike}})$ 生成密钥对 (pk, sk).
- $\text{Encaps}(pk)$: 以接收方公钥 $pk = pk_\beta$ 为输入, 运行 $\text{NIKE.KeyGen}(pp_{\text{nike}})$ 生成临时密钥对 (pk_α, sk_α), 构造协商公钥集合 $S = \{pk_\alpha, pk_\beta, pk_\gamma\}$, 计算 $k_S \leftarrow \text{NIKE.ShareKey}(sk_\alpha, S)$, 输出密文 $c = (pk_\alpha, pk_\beta)$ 和会话密钥 $k = k_S$. 定义 $\mathcal{NP}$ 语言 $L_{pk}^{\text{KEM}} = \{(pk_\alpha, pk) \mid pk_\alpha \in PK\}$ 公钥 pk 封装的所有合法密文集合.
- $\text{Decaps}(sk, c)$: 以解封装私钥 $sk = sk_\beta$ 和密文 $c = (pk_\alpha, pk_\beta)$ 为输入, 构造协商公钥集合 $S = \{pk_\alpha, pk_\beta, pk_\gamma\}$, 计算 $k_S \leftarrow \text{ShareKey}(sk_\beta, S)$, 输出会话密钥 $k = k_S$.
- $\text{Decaps}'(edk, c)$: 以托管解封装私钥 $edk = sk_\gamma$ 和密文 $c = (pk_\alpha, pk_\beta)$ 为输入, 构造协商公钥集合 $S = \{pk_\alpha, pk_\beta, pk_\gamma\}$, 计算 $k_S \leftarrow \text{NIKE.ShareKey}(sk_\gamma, S)$, 输出会话密钥 $k = k_S$.

♣

构造 6.7的正确性和一致性由底层的三方 NIKE 保证, 安全性由以下定理保证.

定理 6.7

如果三方 NIKE 在 HKR (resp. DKR) 设定下是 CKS-light 安全的, 那么构造 6.7是 IND-CPA (resp. IND-CCA) 安全的.

♡

证明 我们给出上述定理 IND-CCA 安全情形的证明, IND-CPA 安全情形的证明可以类似给出. 假定存在 PPT 敌手 $\mathcal{A}$ 能够打破可托管密钥封装机制的 IND-CCA 安全, 则能够构造出 PPT 敌手 $\mathcal{B}$ 以相同的优势打破三方 NIKE 在 DKR setting 下的 CKS-light 安全性. $\mathcal{B}$ 扮演挑战者与 $\mathcal{A}$ 在可托管密钥封装机制的 IND-CCA 安全试验中交互如下.

系统建立: 给定 pp_{nike}, $\mathcal{B}$ 询问 $\mathcal{O}_{\text{regH}}$ 三次, 获得公钥集合 $S = (pk_\alpha, pk_\beta, pk_\gamma)$ 和 k^*, 其中 k^* 或是 k_S 或是随机会话密钥. $\mathcal{B}$ 设定系统公开参数 $pp = (pp_{\text{nike}}, pk_\gamma)$, 公钥 $pk = pk_\beta$, 将 (pp, pk) 发送给 $\mathcal{A}$.

挑战: $\mathcal{B}$ 设定 $c^* = (pk_\alpha, pk_\beta)$ 为挑战密文, 将 (c^*, k^*) 发送给 $\mathcal{A}$.

解封装询问: $\mathcal{A}$ 可自适应的发起解封装询问. 对于解封装询问 $c \neq c^*$, 如果 $c \notin L_{pk_\beta}^{\text{KEM}}$, $\mathcal{B}$ 则根据算法 Decaps 的定义直接拒绝, 返回 $\perp$; 否则 $\mathcal{B}$ 询问 $(pk, pk_\beta, pk_\gamma)$ 的会话密钥 k, 其中 pk 是 c 中的第一个元素. $\mathcal{B}$ 将 k 转发给 $\mathcal{A}$. 注意到 $c \neq c^*$ 的限制确保了 $(pk, pk_\beta, pk_\gamma) \neq S$, 因此 $\mathcal{B}$ 的会话密钥询问总是可容许的.

猜测: $\mathcal{A}$ 输出对 b 的猜测 b', $\mathcal{B}$ 将 b' 转发给它的挑战者.

如果 $k^* = k_S$, 那么 k^* 是由公钥 pk_β 封装在密文 pk_α 中的会话密钥. 如果 k^* 是随机密钥, 那么 k^* 也是一个随机的会话密钥. 因此, $\mathcal{B}$ 对挑战者的模拟是完美的. 综上, $\mathcal{B}$ 将以与 $\mathcal{A}$ 打破可托管密钥封装机制 IND-CCA 安全相同的优势打破三方 NIKE 在 DKR 设定下的 CKS-light 安全. 定理得证!

6.2.2.1 基于放宽三方 NIKE 的优化

事实上, 构造 6.7并不需要三方 NIKE 的全部威力, 弱化版本的三方 NIKE 即可满足需求. 标准的三方 NIKE 默认系统中所有用户的公钥来自同一密钥空间, 具备相同的代数属性. 我们可以对这一点进行放宽: 即允许系统中存在 Type-A、Type-B 和 Type-C 这三种类型的公钥, 集齐三种类型公钥的三方可进行协商出共同的会话密钥. 在可托管密钥封装机制的构造中, 可以令用户的公钥为 Type-A 类型, 充当封装密文的临时公钥为 Type-B 类型, 密钥托管中心的公钥为 Type-C 类型. "公钥多样性放宽" 的意义在于可以扩大 NIKE 协议选择的空间, 从而提升可托管密钥封装机制的效率. 为了验证这一洞察, 我们下面展示如何放宽 Joux 三方 NIKE [268], 并用其构造可托管密钥封装机制.

文献 [358, 262] 指出了基于双线性映射密码学 (pairing-based cryptography) 中的理论与实际不一致: 出于描述简洁、假设更弱的优点, 学术论文中的密码方案多使用对称双线性映射进行设计, 而在工程实现中往往使用计算和通信效率均更优的非对称双线性映射 (如 Type-III 型) 构造. 经典的 Joux 三方 NIKE 正是基于对称双线性映射构造, 且难以迁移为基于非对称双线性映射的构造. 因此, 原始 Joux 三方 NIKE 的效率劣势使其无法导出高效的可托管密钥封装机制, 其蕴含的可托管 ElGamal 方案 [267] 效率低下. 我们的解决思路是对 Joux 三方 NIKE 进行 "公钥多样性放宽", 使得放宽后的版本可以基于 Type-III 型双线性映射构造. 为了使所得可托管密钥封装机制的公钥尺寸尽可能的小, 对公钥的设定如下: 令 Type-A 型公钥的代数结构为 $g_1^b \in \mathbb{G}_1$, Type-B 型公钥的代数结构为 $g_2^c \in \mathbb{G}_2$, Type-C 型公钥的代数结构为 $(g_1^a, g_2^a) \in \mathbb{G}_1 \times \mathbb{G}_2$. 所得的可托管密钥封装机制如下所示:

构造 6.8 (基于放宽 Joux 三方 NIKE 的可托管密钥封装机制)

构造包括以下 5 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 运行 $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T g_1, g_2, e) \leftarrow \mathsf{GenBLGroup}(1^\kappa)$, 随机选取 $edk \xleftarrow{R} \mathbb{Z}_q$, 计算 $pk_\gamma^1 \leftarrow g_1^{edk} \in \mathbb{G}_1$, $pk_\gamma^2 \leftarrow g_2^{edk} \in \mathbb{G}_2$, 输出系统公开参数 $pp = (pk_\gamma^1, pk_\gamma^2)$ 和托管解封装私钥 edk. 公钥空间为 $\mathbb{G}_1$, 封装密文空间为 $\mathbb{G}_2$, 会话密钥空间为 $\mathbb{G}_T$.
- $\mathsf{KeyGen}(pp)$: 随机选取私钥 $sk \xleftarrow{R} \mathbb{Z}_q$, 计算公钥 $pk \leftarrow g_1^{sk} \in \mathbb{G}_1$.
- $\mathsf{Encaps}(pk)$: 令 $pk = pk_\beta$, 随机选取 $sk_\alpha \xleftarrow{R} \mathbb{Z}_q$, 计算封装密文 $c \leftarrow g_2^{sk_\alpha} \in \mathbb{G}_2$, 计算会话密钥 $k \leftarrow e(pk_\beta, pk_\gamma^2)^{sk_\alpha}$.
- $\mathsf{Decaps}(sk, c)$: 令 $sk = sk_\beta$, 输出 $k \leftarrow e(pk_\gamma^1, c)^{sk_\beta}$.
- $\mathsf{Decaps}'(edk, c)$: 令 $edk = sk_\gamma$, 输出 $k \leftarrow e(pk_\beta, c)^{sk_\gamma}$.

6.3 代理重加密

呼童转赠之, 入门即翩翻.

— 明 · 吴宽《答济之谢送鹤》

公钥加密的一个基本目标是只允许在加密时选择的一个或多个密钥才能解密密文. 例如, 使用 Alice 的 RSA 公钥 (N, e) 加密消息 m 的密文 $c = m^e \bmod N$, 仅能使用 Alice 选择的满足条件 $ed = 1 \bmod \phi(N)$ 的密钥 (N, d) 解密. 要将密文 c 改变为 Bob 的密钥加密的密文, 则需要获取原始消息 m 及 Bob 的合法公钥 $(\hat{N}, \hat{e})$. 众多密码技术都希望具有这种基本且理想的性质, 以防止不受信任的实体改变信息的 (加密) 密钥. 恰恰相反, 代理重加密 (proxy re-encryption, PRE) [359] 试图将改变密文的加密密钥同时不泄漏解密密钥或原始消息成为一种现实. 如图 6.8所示, 在代理重加密中, 存在一个代理密钥或转换密钥, 记作 $rk_{A\to B}$, 允许一个非可信实体, 即代理者 (proxy), 利用代理密钥将授权人 Alice 公钥加密的密文转换为被授权人 Bob 公钥加密的密文, 而代理服务器不会获取该密文对应明文的任何信息.

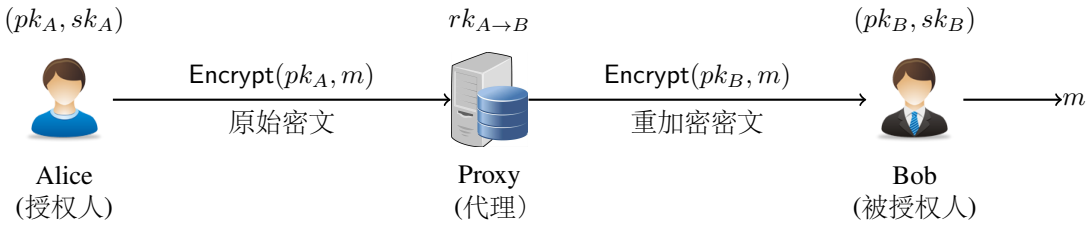


图 6.8: 代理重加密方的应用模式

一般地, 授权人 Alice 的私钥必须参与到代理密钥生成算法之中, 否则任何不可信实体 Carol 都可以生成一个从 Alice 到 Carol 的代理密钥, 从而破坏 Alice 密文的机密性. 根据代理密钥生成的不同方式, 可以将代理密钥分为对称代理密钥 (symmetric proxy key) 和非对称代理密钥 (asymmetric proxy key). 对称代理密钥一般由 Alice 和 Bob 的私钥联合产生, 利用代理密钥和一方的私钥可能推导出另一方的私钥, 因此二者必须相互信任. 非对称代理密钥一般由 Alice 独立产生 (需知道 Bob 的公钥) 或同 Bob 联合产生, 但是不会危害 Bob 的密钥安全性. 根据非对称代理密钥的特点, 一个对称加密方案可以利用非对称代理密钥转化为一个公钥加密方案, 而方案的公钥即为授权人的对称私钥和代理密钥, 任何用户可以先利用公开的授权人的对称私钥加密消息, 再利用代理密钥转化为接收者 (被授权人) 私钥加密的密文.

目前, 代理重加密技术在许多领域有着重要的应用, 如智能卡等资源受限环境的密钥管理和密码运算, 加密垃圾邮件过滤, 安全网络文件存储等. 特别地, Ateniese 等 [360] 设计了一种文件存储系统, 使用不可信访问控制服务器管理存储在分布式、不可信块存储区中的加密文件, 使用代理重加密技术来实现访问控制, 不需要向访问控制服务器提供完全解密权限. 该系统是首个使用代理重加密技术的实验性实施和评估系统, 充分说明了代理重加密技术可在实践中可以有效发挥作用.

下面介绍代理重加密的基本概念、性质和构造方法.

6.3.1 代理重加密的定义与安全性

根据代理密钥的功能, 代理重加密可以分为双向代理重加密 (bidirectional proxy re-encryption) 和单向代理重加密 (unidirectional proxy re-encryption). 在双向代理重加密中, 代理密钥可以相互转换两个用户的密文, 而在单向代理重加密中, 代理密钥仅能转换授权人的密文, 反之无法进行. 下面以单向代理重加密为例, 介绍代理重加密的形式化定义及安全模型.

定义 6.9 (单向代理重加密)

一个单向代理重加密方案包含以下 6 个 PPT 算法:

- $\mathsf{Setup}(1^\kappa)$: 系统参数生成算法以安全参数 1^κ 为输入, 输出系统公开参数 pp, 其中 pp 包含了用户的公钥空间 PK、私钥空间 SK、消息空间 M 和密文空间 C. 类似公钥加密方案, 该算法由可信第三方生成并公开, 系统中的所有用户共享, 所有算法均将 pp 作为输入的一部分.
- $\mathsf{KeyGen}(pp)$: 密钥生成算法以公开参数 pp 为输入, 输出一对公/私钥 (pk, sk), 其中 pk 公开, sk 保密. 一个 PRE 方案至少包含授权人 Alice 和被授权人 Bob 两个实体, 二者的密钥分别记作 (pk_A, sk_A) 和 (pk_B, sk_B).
- $\mathsf{ReKeyGen}(pk_A, sk_A^\dagger, pk_B, sk_B^*)$: 重加密密钥生成算法以授权人和被授权人的密钥为输入, 输出一个代理密钥 $rk_{A\to B}$. 在生成非对称代理密钥时, 该算法的第四部分输入 sk_B^* 一般为空, 此时称重加密密钥生成算法是非交互的. 算法的第二部分输入一般是授权人的私钥 sk_A, 也可能是 Alice 到 Carol 的代理密钥 $sk_{A\to C}$ 和 Carol 的私钥 sk_C.
- $\mathsf{Encrypt}(pk, m)$: 加密算法公钥 $pk \in PK$ 和消息 $m \in M$ 为输入, 输出一个密文 $c \in C$.
- $\mathsf{Decrypt}(sk, c)$: 解密算法以私钥 $sk \in SK$ 和密文 $c \in C$ 为输入, 输出一个消息 $m \in M$.
- $\mathsf{ReEncrypt}(rk_{A\to B}, c_A)$: 重加密算法以代理密钥 $rk_{A\to B}$ 和授权人 Alice 的密文 c_A 为输入, 输出一个重加密密文 c_B.

♣

正确性. 代理重加密的加密算法和解密算法的形式不一定是唯一的, 可能包含若干个不同的算法. 例如有些方案包含两个不同层次的加密方式, 第一层次加密的密文不能够被代理密钥进行转换, 而第二层次加密的密文可以被代理密钥转换. 这为发送者在使用 Alice 同一个公钥进行加密时, 可以有选择地决定仅将消息加密给 Alice, 还是加密给 Alice 和其他用户 (被授权人). 不管发送者采用哪种方式对消息 m 进行加密, Alice 应该能够选择一种解密方式利用自己的私钥 sk_A 恢复出原始消息 m, 而对于任意转换后的密文 $c_B = \mathsf{ReEncrypt}(rk_{A\to B}, c_A)$, 被授权人 Bob 也可以利用自己的私钥进行解密且解密结果与 Alice 解密的结果需要一致.

严格地说, 对于任意由密钥生成算法 $\mathsf{KeyGen}(pp)$ 产生的 Alice 的公私钥对 (pk_A, sk_A) 和 Bob 的公私钥对 (pk_B, sk_B), 对于任意消息 $m \in M$, 代理重加密方案的正确性分为两种情况: 一是针对授权人 Alice 的. 对于任意加密算法 Encrypt, 总存在一种解密算法 Decrypt, 使得如下等式成立:

$$\mathsf{Decrypt}(sk_A, \mathsf{Encrypt}(pk_A, m)) = m$$

二是针对被授权人 Bob 的. 总存在一种加密算法 Encrypt 和一种解密算法 Decrypt 使得如下等式成立:

$$\mathsf{Decrypt}(sk_B, \mathsf{ReEncrypt}(rk_{A\to B}, \mathsf{Encrypt}(pk_A, m))) = m$$

安全性. 一般来说, 一个代理重加密方案应该类似传统公钥加密方案具有刻画被加密消息隐私性的安全模型, 如语义安全性. 在代理重加密方案中, 攻击者能够获得的信息要比传统公钥加密方案复杂. 除了公钥, 加密算法和解密查询之外, 还可能获得一些代理密钥, 甚至是和部分用户合谋获得的用户私钥. 攻击者的目标之一是截获 Bob 的密文并从中恢复出关于明文的任何有用信息. 在现实环境中, 敌手可能与其他用户如 X 合谋, 获取 X 的公私钥对 (pk_X, sk_X) 以及从 X 到 Bob 的密文转换密钥 $rk_{X\to B}$. 此外, Bob 还可能与合法用户 Carol 相互授权解密, 从而使得敌手可能获取二者之间的代理密钥 $rk_{B\to C}$ 和 $rk_{C\to B}$. 显然, 一个实用的 PRE 方案要能防止上述敌手获取密文中消息的任何有用信息. 下面给出这类安全性的形式化定义.

代理重加密的选择明文不可区分性. 定义一个 PRE 方案敌手 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left| \Pr\left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk_B, sk_B) \leftarrow \mathsf{KeyGen}(pp); \\ (m_0, m_1, state) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{keygen}}, \mathcal{O}_{\mathsf{rekeygen}}, \mathcal{O}_{\mathsf{corrupt}}}(pp, pk_B); \\ \beta \xleftarrow{R} \{0,1\}; \\ c^* \leftarrow \mathsf{Encrypt}(pk_b, m_\beta); \\ \beta' \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathsf{keygen}}, \mathcal{O}_{\mathsf{rekeygen}}, \mathcal{O}_{\mathsf{corrupt}}}(state, c^*); \end{array} \right] - \frac{1}{2} \right|$$

在上述定义中, 三个查询谕言机的定义分别如下:

- $\mathcal{O}_{\mathsf{keygen}}$ 表示公钥查询谕言机, 输入用户身份 i, 如果集合 K 中包含用户 i 的密钥, 输出 pk_i; 否则, 运行 $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(pp)$, 输出 pk_i, 并存储 (i, pk_i, sk_i) 到初始化为 $\{(\text{Bob}, pk_B, sk_B)\}$ 的集合 K 中.
- $\mathcal{O}_{\mathsf{rekeygen}}$ 表示代理密钥查询谕言机, 输入用户身份 i, j, 如果集合 K 中未包含用户 i 或 j 的密钥, 输出 $\perp$; 如果 $(i = \text{“Bob”}) \wedge (j \in X)$, 输出 $\perp$; 否则, 输出 $rk_{i\to j} \leftarrow \mathsf{ReKeyGen}(pk_i, sk_i, pk_j, sk_j^*)$, 并且若 $i =$ “Bob” 则将 j 存储到初始化为空的集合 Y 中.
- $\mathcal{O}_{\mathsf{corrupt}}$ 表示私钥查询谕言机, 输入用户身份 i, 如果集合 K 中包含用户 i 的密钥且 $i \notin Y$ 且 $i \neq$ “Bob”, 输出 sk_i, 并将 i 存储到初始化为空的集合 X 中; 否则, 输出 $\perp$.

如果任意的 PPT 敌手 $\mathcal{A}$ 在上述定义中的优势函数是可忽略的, 则称代理重加密方案 PRE 是 IND-CPA 安全的. 如果增加敌手访问解密谕言机的权限, 则称代理重加密方案 PRE 是 IND-CCA 安全的.

笔记 在上面的三个查询谕言机中, 集合 K 相当于存储了所有用户的公私钥对; 集合 X 存储了所有被腐化或者说与敌手 $\mathcal{A}$ 合谋的恶意用户的身份信息. 对于这类用户, 是不允许敌手获取从 Bob 到该用户的代理密钥, 否则无法保障 Bob 的密文的任何安全性, 这也是在代理密钥查询谕言机中不允许 $i =$ “Bob” 的原因; 集合 Y 存储了所有授权访问 Bob 密文的用户身份信息. 敌手不能访问集合 Y 中用户的私钥, 否则敌手可以利用 Bob 授权给该用户的代理密钥直接解密 Bob 的密文, 这也是再私钥查询谕言机中不允许 $i \in Y$ 的原因. 对于双向代理重加密方案, 还需要进一步限制敌手访问从恶意用户到 Bob 的代理密钥.

如果用户 Bob 将解密权限授权给一个恶意用户, 会有什么影响? 显然, 敌手可以与恶意用户合谋直接解密挑战密文 c^*, 从而使得代理重加密的选择密文不可区分安全性是无法实现的. 那么, 授权解密权限与用户主密钥的完全泄漏是否等价呢? 从直观上看, 解密权限是通过代理密钥实现的, 并非直接将授权人的主密钥发送给被授权人, 因此, 保护授权人主密钥的安全性是可行且必要的. 接下来介绍一种刻画授权人主密钥安全性 (master secret security, MSS) 的模型.

代理重加密的主密钥安全性. 定义一个 PRE 方案敌手 $\mathcal{A}$ 的优势函数如下:

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \Pr\left[sk = sk_B : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\kappa); \\ (pk_B, sk_B) \leftarrow \mathsf{KeyGen}(pp); \\ sk \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{keygen}}, \mathcal{O}_{\mathsf{rekeygen}}, \mathcal{O}_{\mathsf{corrupt}}}(pp, pk_B); \end{array}\right]$$

在上述定义中, 三个查询谕言机的定义分别如下:

- $\mathcal{O}_{\mathsf{keygen}}$ 表示公钥查询谕言机, 输入用户身份 i, 如果集合 K 中包含用户 i 的密钥, 输出 pk_i; 否则计算 $(pk_i, sk_i) \leftarrow \mathsf{KeyGen}(pp)$, 输出 pk_i, 并存储 (i, pk_i, sk_i) 到初始化为 $\{(\text{Bob}, pk_B, sk_B)\}$ 的集合 K 中.
- $\mathcal{O}_{\mathsf{rekeygen}}$ 表示代理密钥查询谕言机, 输入用户身份 i, j, 如果集合 K 中未包含用户 i 或 j 的密钥, 输出 $\perp$; 否则, 输出 $rk_{i\to j} \leftarrow \mathsf{ReKeyGen}(pk_i, sk_i, pk_j, sk_j^*)$.
- $\mathcal{O}_{\mathsf{corrupt}}$ 表示私钥查询谕言机, 输入用户身份 i, 如果集合 K 中包含用户 i 的密钥且 $i \neq$ “Bob”, 输出 sk_i; 否则, 输出 $\perp$.

如果任意的 PPT 敌手 $\mathcal{A}$ 在上述定义中的优势函数是可忽略的, 则称代理重加密方案 PRE 是主密钥安全的.

笔记 主密钥安全模型中的三个查询谕言机与前面的定义类似, 但是有较大的区别: 在主密钥安全模型中, 除了不能访问挑战用户 (授权人) Bob 的主密钥 sk_B 外, 敌手可以进行任意用户的主密钥和任意用户之间的代理密钥. 在实际应用中, 即使敌手可以获取一个合法用户的授权解密权限, 但是保护合法用户主密钥安全性依然是有意义的. 事实上, 一些代理重加密方案的密文可能有多种形式, 一种形式是允许代理密钥进行密文转化和授权解密的, 另一种形式是不允许进行密文转换的, 这类密文只有主密钥才能解密.

6.3.2 代理重加密的构造

代理重加密方案可以看作是标准公钥加密方案的一个延申, 除了增加密文转换功能外, 其他功能和标准公钥加密是一样的. 尽管如此, 由于代理密钥的引入, 使得不同代理重加密方案具有的性质也各不一样. 有些性质可以

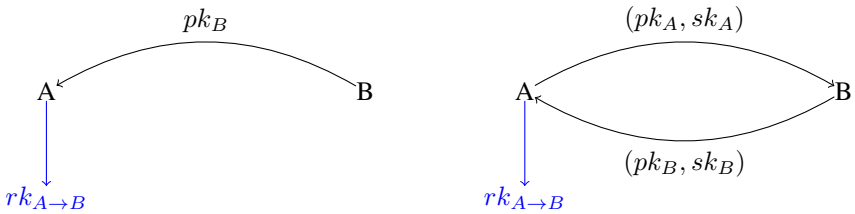
通过前面的安全模型来刻画, 而在效率或功能等方面的性质无法通过安全模型进行描述. 下面将代理重加密方案可能具有的几种典型性质总结如下:

1. **单向代理 vs 双向代理:** 一个代理密钥 $rk_{A\to B}$ 只能实现从用户 A 到用户 B 的密文转换, 那么此类方案称为单向代理重加密方案; 反之, 如果该密文也可以实现从用户 B 到用户 A 的密文转换, 则该方案是双向代理重加密方案.

单向代理重加密　　双向代理重加密

图 6.9: 单向代理重加密 vs 双向代理重加密

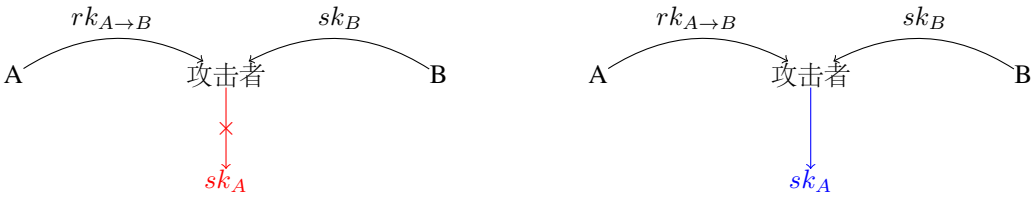
2. **非交互代理密钥 vs 交互式代理密钥:** 当 Alice 授权解密权限给 Bob 时, 如果生成代理密钥 $rk_{A\to B}$ 只需要 Bob 的公钥 pk_B, 则该代理密钥是非交互的; 如果生成代理密钥 $rk_{A\to B}$ 需要双方共同参与或者需要依赖一个可信第三方, 则该代理密钥是交互的.



非交互代理密钥　　交互式代理密钥

图 6.10: 非交互代理密钥 vs 交互代式理密钥

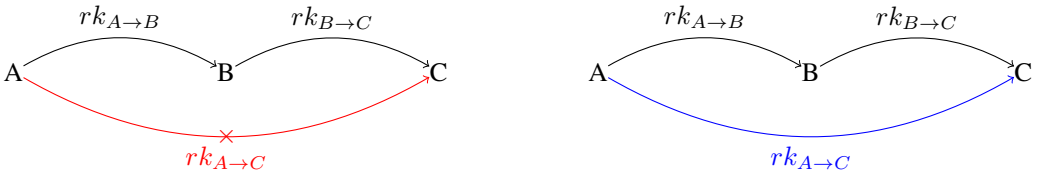
3. **抗合谋攻击 vs 合谋攻击:** 如果利用 Alice 到 Bob 的代理密钥 $rk_{A\to B}$ 和 Bob 的私钥 sk_B 不能恢复 Alice 的私钥 sk_A, 则代理重加密方案具有抗合谋攻击的性质; 否则, 不具有抗合谋攻击的性质. 实际上, 主密钥安全性的定义就是刻画此类合谋攻击的.



抗合谋攻击　　合谋攻击

图 6.11: 抗合谋攻击 vs 合谋攻击

4. **非传递性 vs 可传递性:** 如果利用从 Alice 到 Bob 的代理密钥 $rk_{A\to B}$ 和从 Bob 到 Carol 的代理密钥 $rk_{C\to C}$ 可以推导出从 Alice 到 Carol 的代理密钥 $rk_{A\to C}$, 则此类代理密钥具有可传递性, 否则具有不可传递性.
5. **代理不可见性 vs 代理可见性:** 所谓代理不可见性 (proxy invisibility), 有些文献也叫做代理透明性 (proxy transparency), 是指加密消息的发送者和任何被授权解密者都不必知道代理是否存在. 也就是说, 接收到的密文是发送者直接发送的还是通过代理服务器转换后的密文, 对于被授权解密者来说是不可区分的. 如果接收者需要知道是否是转换后的密文才能选择不同的方式进行解密, 那么代理必须是可见的. 代理是否可见对实际应用可能会有一定的影响. 例如错误地判断一个密文是原始密文还是转换而来的密文可能解密出错误的结果.



代理密钥不可传递　　代理密钥可传递

图 6.12: 非传递性 vs 可传递性

6. **单跳性 vs 多跳性性:** 代理重加密的单跳性是指一个通过密钥转换而来的密文不能被其他代理密钥再进行转换. 而多跳性则允许进一步被转换, 例如 Alice 的密文 c_A 经过代理密钥 $rk_{A\to B}$ 重加密后变为 Bob 的密文 c_B, 代理服务器利用代理密钥 $rk_{B\to C}$ 还可以进一步将密文 c_B 转换为 Carol 的密文 c_C. 利用多跳性, 代理服务器可能会做出未授权的密文转换, 这对于控制原始密文的访问权限是不利的.
7. **非转让性 vs 可转让性:** 非转让性是指解密权限不能进一步地被授权给其他用户. 在实际应用中, 可能存在部分恶意用户如 Bob, 在获取了 Alice 授权的代理密钥 $rk_{A\to B}$ 后, 是否可以将解密权限授权给 Carol, 例如生成代理密钥 $rk_{A\to C}$?

注记 6.9

非转让性与非传递性看上去很相似, 但是具有较大的区别. 在非传递性中, 敌手仅知道用户的公钥和一些 (可公开的) 代理密钥, 而在非转让性中, 敌手不仅知道一些公开的信息, 还知道恶意用户的私钥, 如果代理密钥是非交互的, 那么敌手是可以自己生成从 Bob 到 Carol 的代理密钥 $rk_{B\to C}$. 因此, 如果一个代理重加密方案的代理密钥具有可传递性, 那么该方案一定不具有非转让性. 非转让性和单跳性的概念也很类似, 但是单跳性主要从转换后的密文是否能代理密钥进行转换的角度考虑的, 而非转让性不仅仅如此. 非转让性是为了阻止解密权限的滥用, 不仅仅是阻止敌手生成新的代理密钥或者将重加密密文进一步转换. 因此, 要实现非转让性似乎比实现非传递性要更加困难. 例如, Bob 在获取了代理密钥 $rk_{A\to B}$ 后, 自然可以解密 Alice 的重加密密文, 进一步可以将恢复的消息授权给其他用户查看. 这似乎是无法避免的一个问题.

除了以上几种性质外, 每个用户的密钥数量也是非常重要的. 在一些方案中, 用户的密钥数量与授权人的数量线性相关, 导致用户的密钥存储和管理比较麻烦. 授权人是否能够解密最初发送给她的重加密后的密文也是一个比较重要的性质. 假设密文 $c = \mathsf{Encrypt}(pk_A, m)$ 是一个发送者发送给 Alice 的原始密文, 那么 Alice 具有访问该密文的权限. 那么, 如果该密文被重新加密为 Bob 的密文 $c' = \mathsf{ReEncrypt}(rk_{A\to B}, c)$, 那么重加密后的密文 Alice 是否还有访问权限? 如果简单地将原始密文作为重加密密文的一部分, 那么 Alice 仍然可以恢复原始消息. 这似乎与代理不可见性矛盾.

结合上述性质, 下面介绍几种典型的代理重加密方案的设计方法.

双向代理重加密方案

双向代理重加密可以通过一个标准的公钥加密方案来构造. 假设 $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ 是一个标准的公钥加密方案. 其构造思想如下如图 6.13所示:

PRE 的密钥生成算法是利用公钥加密方案的密钥生成算法 PKE.KeyGen 生成三组密钥 (pk_1, sk_1), (pk_2, sk_2) 和 (pk_3, sk_3). Alice 和 Bob 分别持有其中的两组密钥且仅有一组公共的密钥, 例如 Alice 持有密钥 (pk_1, sk_1) 和 (pk_2, sk_2), Bob 持有密钥 (pk_1, sk_1) 和 (pk_3, sk_3). 而重加密密钥由 Alice 和 Bob 非公共的密钥组成, 即, Proxy 持有的重加密钥为 $rk_{A\leftrightarrow B} = \{(pk_2, sk_2), (pk_3, sk_3)\}$. 为简化描述, 下面用 E_i 和 D_i 分别表示公钥加密方案在公钥 pk_i 下的加密算法和在私钥 sk_i 下的解密算法.

PRE 的加密算法是利用 PKE 的加密算法加密消息两次, 即 $c = \mathsf{E}_2(\mathsf{E}_1(m))$. 对于 Alice 来说, 利用自己的私钥 $sk_A = (sk_1, sk_2)$ 可以直接解密密文 $\mathsf{D}_1(\mathsf{D}_2(c)) = m$.

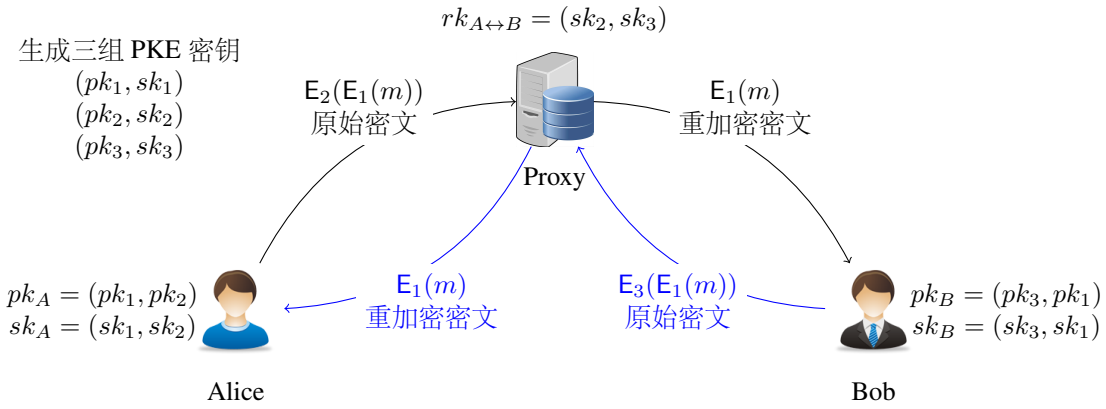


图 6.13: 基于标准公钥加密的双向代理重加密构造思想

PRE 的重加密算法是 Proxy 利用私钥 sk_2 将密文 c 部分解密为 $c'=\mathsf{D}_2(c)=\mathsf{E}_1(m)$. 利用密钥 sk_1, Bob 可以从重加密密文 c' 中恢复出消息 $m=\mathsf{D}_1(c')=\mathsf{D}_1(\mathsf{E}_1(m))$.

利用类似的方式可以将 Bob 的密文转化为 Alice 的密文, 因此这是一个双向代理重加密方案. 如果将重加密密文定义为 $c'=\mathsf{E}_3(\mathsf{D}_2(c))=\mathsf{E}_3(\mathsf{E}_1(m))$, 则重加密密文的形式同 Bob 的原始密文形式是完全一样的, 所以这个代理重加密方案具有代理不可见性.

上述方案的构造思想源于文献 [361], 如果所基于的公钥加密方案是 IND-CPA (或 IND-CCA) 安全的, 那么构造的双向代理重加密方案也是 IND-CPA (或 IND-CCA) 安全的.

由于通用构造方法需要双重加密和解密, 计算时间一般要比所基于的公钥加密方案多出一倍. 实际上, 基于具体的公钥加密算法可以设计出更高效的双向代理重加密方案. 下面以 BBS 方案 [359] 为例, 介绍一种基于 ElGamal 加密方案的双向代理重加密方案.

构造 6.9 (基于 ElGamal 的双向代理重加密方案)

- $\mathsf{Setup}(1^\kappa)$: 运行 $(\mathbb{G},q,g)\leftarrow\mathsf{GenGroup}(1^\kappa)$. 输出系统参数 $pp=(\mathbb{G},q,g)$.
- $\mathsf{KeyGen}(pp)$: 随机选择 $x\xleftarrow{\mathrm{R}}\mathbb{Z}_q$, 计算 $h=g^x$, 输出公钥 $pk=h$ 和私钥 $sk=x$.
- $\mathsf{ReKeyGen}(pk_A,sk_A^\dagger,pk_B,sk_B^*)$: 假设 $(pk_A,sk_A)=(h_A,x_A)$, $(pk_B,sk_B)=(h_B,x_B)$, 输出代理密钥 $rk_{A\to B}=x_A^{-1}x_B \bmod q$.
- $\mathsf{Encrypt}(pk,m)$: 随机选择 $r\xleftarrow{\mathrm{R}}\mathbb{Z}_q$, 计算 $c_1=h^r \bmod p$, $c_2=g^r m$. 输出密文 $c=(c_1,c_2)$.
- $\mathsf{Decrypt}(sk,c)$: 计算 $m=c_2/c_1^{1/x}$, 输出消息 m.
- $\mathsf{ReEncrypt}(rk_{A\to B},c_A)$: 假设 $c_A=(c_1,c_2)$, 计算 $c_1'=(c_1)^{rk_{A\to B}}$, 输出重加密密文 $c_B=(c_1',c_1)$. ♣

正确性. 构造 6.9的形式与标准的 ElGamal 加密方案类似, 但是在参数使用上稍有不同, 并且需要利用私钥的逆元来解密密文. 对于重加密密文, 由于

$$c_1'=(c_1)^{rk_{A\to B}}=(g^{x_A r})^{x_A^{-1}x_B}=(g^{x_B r})=(h_B)^r$$

所以, 重加密密文 c_B 是 Bob 的一个形式合法的密文, 因此 Bob 可以正常解密 c_B.

安全性. 下面针对两个用户的环境简要分析构造 6.9在 DDH 假设下满足 IND-CPA 安全性. 对于多用户环境, 需要考虑恶意用户的情况, 读者可尝试去分析. 假设 (g,g^a,g^b,T) 是一个 DDH 问题实例. 下面构造一个模拟者利用敌手区分 PRE 密文的能力来解决一个 DDH 问题. 模拟者随机选择 $z\xleftarrow{\mathrm{R}}\mathbb{Z}_q$, 令 Alice 的公钥为 $pk_A=g^a$, Bob 的公钥为 $pk_B=(g^b)^z$. 由于 a 和 z 都是随机选取的, 所以 Alice 和 Bob 的公钥与实际选取的分布是一样的. 当敌手查询 Alice 与 Bob 之间的代理密钥 $rk_{A\to B}$ 或 $rk_{B\to A}$ 时, 模拟者直接返回 z 或 $z^{-1} \bmod q$. 由于 $sk_A=a$, $sk_B=az \bmod q$, 所以 $rk_{A\to B}=sk_A^{-1}sk_B=z$. 模拟者定义挑战密文为:

$$c^*=(c_1,c_2)=(T,g^b m_\beta)$$

显然, 如果 $T = g^{ab}$, 则上述密文是一个合法的 PRE 密文; 如果 T 是随机选取的, g^b 与 T 完全独立且 y 是随机选取的, 此时密文 c^* 是 $\mathbb{G} \times \mathbb{G}$ 上的一个随机元素. 所以敌手能够区分挑战密文 c^* 等价于能够区分 DDH 问题.

注记 6.10

在上述通用构造方案中, Proxy 和 Bob 联合可以完全恢复 Alice 的私钥 $sk_A = (sk_1, sk_2)$. 而在基于 ElGamal 的构造中, Proxy 和 Bob 联合可以计算出 $sk_A = rk_{A\to B}^{-1} \cdot sk_B \bmod q$. 所以上述两种双向代理重加密方案都不能抵抗合谋攻击. 在多个用户环境下, 通用构造的用户密钥管理是比较复杂的, 每两个用户之间都要共享一对不同的密钥. 构造 6.9尽管可以避免用户密钥增长问题, 但是代理密钥具有传递性, 如: 已知 $rk_{A\to B} = x_A^{-1}x_B$ 和 $rk_{B\to C} = x_B^{-1}x_C$, 则

$$(x_A^{-1}x_B) \cdot (x_B^{-1}x_C) = x_A^{-1}x_C = rk_{A\to C}.$$

单向代理重加密方案

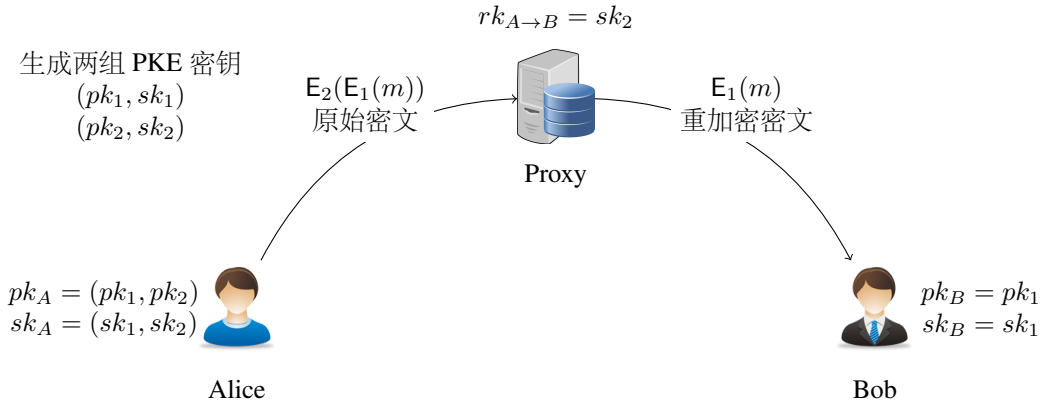


图 6.14: 基于标准公钥加密的单向代理重加密构造思想

Ivan 等在文献 [361] 中介绍了一种利用标准公钥加密构造单向代理重加密方案的方法. 该方法利用秘密共享的方式将授权人的私钥 sk 拆分为两部分 sk_1 和 sk_2 以实现密文转换. 如图 6.14所示. 其中 Alice 拥有两组 PKE 密钥, 而 Proxy 和 Bob 分别拥有其中的一组. 显然, 该方案不能抵抗合谋攻击, 达不到主密钥安全性. 当授权用户数量增长时, Bob 持有的密钥数量也随之增长. 下面介绍一种在效率和安全性方面更加完善的构造. 该构造由 Ateniese 等 [360] 设计, 可以看作是 BBS 方案在双线性配对上的实现.

构造 6.10 (基于双线性映射的单向代理重加密方案)

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{GenBLGroup}(1^\kappa)$ 生成一个 Type-I 双线性映射 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$. 令 $Z = e(g, g)$. 输出系统参数 $pp = (\mathbb{G}, \mathbb{G}_T, q, g, e, Z)$.
- $\mathsf{KeyGen}(pp)$: 随机选择 $x \xleftarrow{R} \mathbb{Z}_q$, 计算 $h = g^x$, 输出公钥 $pk = h$ 和私钥 $sk = x$.
- $\mathsf{ReKeyGen}(pk_A, sk_A, pk_B)$: 假设 $(pk_A, sk_A) = (h_A, x_A)$, $pk_B = h_B$, 输出代理密钥 $rk_{A\to B} = h_B^{1/x_A}$.
- 第一层次加密 $\mathsf{Encrypt}_1(pk, m)$: 对于消息 $m \in \mathbb{G}_T$, 随机选择 $r \xleftarrow{R} \mathbb{Z}_q$, 计算 $c_1 = Z^{xr}$, $c_2 = Z^r m$. 输出密文 $c = (c_1, c_2)$.
- 第二层次加密 $\mathsf{Encrypt}_2(pk, m)$: 对于消息 $m \in \mathbb{G}_T$, 随机选择 $r \xleftarrow{R} \mathbb{Z}_q$, 计算 $c_1 = g^{xr}$, $c_2 = Z^r m$. 输出密文 $c = (c_1, c_2)$.
- 第一层次密文解密算法 $\mathsf{Decrypt}_1(sk, c)$: 计算 $m = c_2/c_1^{1/x}$, 输出消息 m.
- 第二层次密文解密算法 $\mathsf{Decrypt}_2(sk, c)$: 计算 $m = c_2/e(c_1, g)^{1/x}$, 输出消息 m.
- $\mathsf{ReEncrypt}(rk_{A\to B}, c_A)$: 假设 $c_A = (c_1, c_2)$ 是第二层次密文, 计算 $c_1' = e(c_1, rk_{A\to B})$, 输出重加密密文 $c_B = (c_1', c_1)$.

正确性. 对于第一层次和第二层次的密文, 可以直接验证解密结果的正确. 对于重加密密文, 由于

$$c_1' = e(c_1, rk_{A\to B}) = e(g^{x_A r}, g^{x_B/x_A}) = e(g^{x_B r}, g) = Z^{x_B r}$$

所以, 重加密密文 c_B 是 Bob 的一个形式合法的第一层次密文, 因此 Bob 可以正常解密 c_B.

安全性. 构造 6.10的安全性可在选择明文不可区分安全模型及主密钥安全模型下分别讨论, 具体结论分别定理 6.8和定理 6.9. 定理的证明分别依赖双线性映射 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$ 上的如下两个假设:

- 假设 1: 已知 (g, g^a, g^b, T), 其中 $a, b \xleftarrow{R} \mathbb{Z}_q^2$, $g\mathbb{G}^n$, $T \in \mathbb{G}_T$, 判断 $T = e(g,g)^{a/b}$ 还是 $\mathbb{G}_T$ 上的一个随机元素是困难的.
- 假设 2: 已知 $(g, g^a, g^{1/a})$, 其中 $a \xleftarrow{R} \mathbb{Z}_q$, 计算 a 是困难的.

定理 6.8

如果假设 1 相对于 GenBLGroup 成立, 那么构造 6.10中的 PRE 是 IND-CPA 安全的. ♡

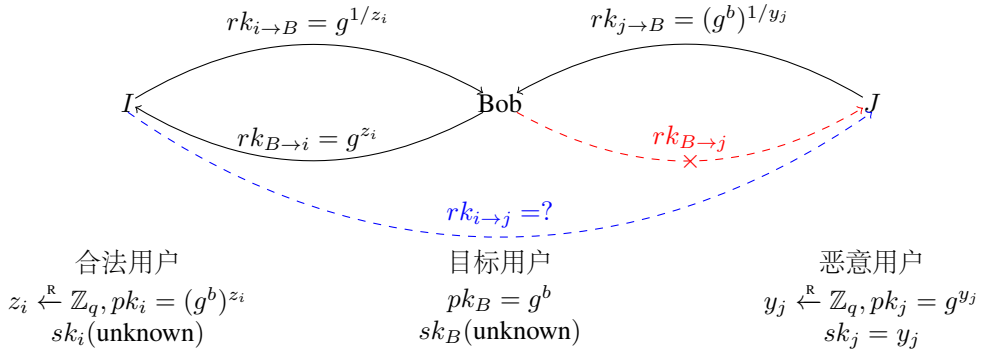


图 6.15: 模拟用户密钥及代理密钥示意图

定理 6.8的证明可以通过构造一个模拟算法, 将方案的选择明文攻击安全性归约到假设 1 上. 给定假设 1 的一个问题实例 (g, g^a, g^b, T), 令目标用户 Bob 的公钥为 $pk_B = g^b$. 考虑两类用户: 合法用户 (记作集合 I) 和恶意用户 (记作集合 J). 模拟算法能够回答敌手如图 6.15所示的不同情况的代理密钥查询. 由于敌手可能与恶意用户合谋, 模拟者必须能够向敌手提供恶意用户的私钥 (私钥查询), 所以恶意用户的公私钥对必须由模拟算法独立于假设 1 产生, 对于任意 $j \in J$, 随机选择 $y_j \xleftarrow{R} \mathbb{Z}_q$, 令 $(pk_j, sk_j) = (g^{y_j}, y_j)$. 对于合法集合 I 中的用户 i, 模拟算法随机选择 $z_i \xleftarrow{R} \mathbb{Z}_q$, 令 $pk_i = (g^b)^{z_i}$. 则模拟算法可以回答相关的代理密钥查询, 即: $rk_{B\to i} = g^{z_i}$, $rk_{i\to B} = g^{1/z_i}$ 和 $rk_{j\to B} = (g^b)^{1/y_j}$. 对于 Bob 到 J 的代理密钥是禁止查询的. 对于挑战密文, 模拟算法可令 $c^* = (g^a, T \cdot m_\beta)$. 当 $T = e(g,g)^{a/b}$ 时, 显然 $c^* = ((g^b)^{a/b}, e(g,g)^{a/b} \cdot m_\beta)$ 是一个合法的第二层次密文.

证明 假设 $\mathcal{A}$ 是一个攻击构造 6.10的 IND-CPA 安全性的敌手. 下面以归约的方式组织证明. 给定在双线性映射 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$ 上的一个问题实例 (g, g^a, g^b, T), 构造一个有效的模拟算法 $\mathcal{B}$, 利用 $\mathcal{A}$ 求解问题实例的解. $\mathcal{B}$ 模拟 $\mathcal{A}$ 的攻击环境如下:

- 初始化: $\mathcal{B}$ 根据问题实例计算 $Z = e(g,g)$, 设置系统参数为 $pp = (\mathbb{G}, \mathbb{G}_T, q, g, e, Z)$, 设置目标用户 Bob 的公钥为 $pk_B = g^b$, 并将 pp 和 pk_B 发送给 $\mathcal{A}$.
- 阶段 1 询问: $\mathcal{B}$ 按以下方式回答 $\mathcal{A}$ 的用户密钥查询和代理密钥查询:
  - 用户密钥查询: 当 $\mathcal{A}$ 查询合法用户集合 I 中用户 i 的公钥时, $\mathcal{B}$ 随机选择 $z_i \xleftarrow{R} \mathbb{Z}_q$, 返回 $pk_i = (g^b)^{z_i}$. 当 $\mathcal{A}$ 查询恶意用户集合 J 中用户 j 的公钥和私钥时, $\mathcal{B}$ 随机选择 $y_j \xleftarrow{R} \mathbb{Z}_q$, 返回 $(pk_j, sk_j) = (g^{y_j}, y_j)$.
  - 代理密钥查询: $\mathcal{B}$ 按以下几种情况回答:
    1. Bob 到 I 的代理密钥: $\mathcal{B}$ 返回 $rk_{B\to i} = g^{z_i}$.
    2. I 到 Bob 的代理密钥: $\mathcal{B}$ 返回 $rk_{i\to B} = g^{1/z_i}$.
    3. J 到 Bob 的代理密钥: $\mathcal{B}$ 返回 $rk_{j\to B} = (g^b)^{1/y_j}$.
- 挑战: 当阶段 1 询问结束时, $\mathcal{A}$ 选择两个挑战消息 $m_0, m_1 \in \mathbb{G}_T$ 发送给 $\mathcal{B}$. 算法 $\mathcal{B}$ 随机选择 $\beta \xleftarrow{R} \{0,1\}$, 返回挑战密文 $c^* = (g^a, T \cdot m_\beta)$.

- 阶段 2 询问: $\mathcal{A}$ 可以继续进行用户密钥查询和代理密钥查询:.
- 输出: 最终, $\mathcal{A}$ 将输出一猜测比特 $\beta' \in \{0,1\}$. 当 $\beta' = \beta$ 时, $\mathcal{B}$ 输出 1; 否则输出 0, 作为自己对假设 1 的问题实例的解.

至此, 完成了模拟算法 $\mathcal{B}$ 的描述. 由于问题实例中 a 和 b 都是随机选取的, z_i 和 y_j 由模拟算法随机选取的, 所以用户的公钥分布和实际选取的结果是一致的. 当 $T = e(g,g)^{a/b}$ 时, 挑战密文的分布和实际计算的结果也是一致的. 所以

$$\mathsf{Adv}_{\mathcal{A}}(\kappa) = \left|\Pr\left[\beta' = \beta | T = e(g,g)^{a/b}\right] - 1/2\right|$$

当 T 是 $\mathbb{G}_T$ 上的随机元素时, $\Pr\left[\beta' = \beta | T = e(g,g)^{a/b}\right] = 1/2$. 根据假设 1, 则

$$\left|\Pr[\mathcal{B}(T = e(g,g)^{a/b}) = 1] - \Pr[\mathcal{B}(T \xleftarrow{R} \mathbb{G}) = 1]\right| \leq \mathsf{negl}(\kappa)$$

由于

$$\begin{aligned}\Pr[\mathcal{B}(T = e(g,g)^{a/b}) = 1] &= \Pr\left[\beta' = \beta | T = e(g,g)^{a/b}\right] \\ \Pr[\mathcal{B}(T \xleftarrow{R} \mathbb{G}) = 1] &= \Pr\left[\beta' = \beta | T \xleftarrow{R} \mathbb{G}\right]\end{aligned}$$

所以 $\mathsf{Adv}_{\mathcal{A}}(\kappa)$ 是可忽略的.

定理 6.8证毕! □

注记 6.11

定理 6.8的证明实际上假设了合法用户和恶意用户是事先已知的, 这与实际环境稍有不同. 如何模拟自适应攻击的用户公钥值得进一步思考, 读者可以阅读文献 [362] 关于自适应攻击的解决方法. 此外, 在回答代理密钥查询时, 模拟算法还省略了非目标用户之间的代理密钥查询, 特别是如何模拟从合法用户到非法用户的代理密钥, 更是值得去思考.

定理 6.9

如果假设 2 相对于 GenBLGroup 成立, 那么构造 6.10中的 PRE 是主密钥安全的.

主密钥安全性的证明思路类似定理 6.8的证明. 根据假设 2 的问题实例 $(g, g^a, g^{1/a})$, 模拟算法 $\mathcal{B}$ 可以将目标用户 Bob 公钥设置为 $pk_B = g^a$, 而将其他用户 (可能都是恶意用户) 的公钥和私钥设置为 $(pk_i, sk_i) = (g^{x_i}, x_i)$, 其中 $x_i \xleftarrow{R} \mathbb{Z}_q$ 是由模拟算法随机选取的. 由于模拟算法知道每个用户的私钥 sk_i 以及 $g^{1/a}$, 所以模拟算法可以回答任意代理密钥查询. 下面给出定理 6.9的完整证明过程:

证明 假设 $\mathcal{A}$ 是一个攻击构造 6.10主密钥安全性的敌手. 下面以归约的方式组织证明. 给定在双线性映射 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$ 上的一个问题实例 $(g, g^a, g^{1/a})$, 构造一个有效的模拟算法 $\mathcal{B}$, 利用 $\mathcal{A}$ 求解问题实例的解. $\mathcal{B}$ 模拟 $\mathcal{A}$ 的攻击环境如下:

- 初始化: $\mathcal{B}$ 根据问题实例计算 $Z = e(g,g)$, 设置系统参数为 $pp = (\mathbb{G}, \mathbb{G}_T, q, g, e, Z)$, 设置目标用户 Bob 的公钥为 $pk_B = g^a$, 并将 pp 和 pk_B 发送给 $\mathcal{A}$.
- 询问: $\mathcal{B}$ 按以下方式回答 $\mathcal{A}$ 的用户密钥查询和代理密钥查询:
  - 用户密钥查询: 当 $\mathcal{A}$ 查询 i 公钥或私钥时, $\mathcal{B}$ 随机选择 $x_i \xleftarrow{R} \mathbb{Z}_q$, 返回公钥 $pk_i = g^{x_i}$ 或私钥 $sk_i = x_i$.
  - 代理密钥查询: $\mathcal{B}$ 按以下几种情况回答:
    1. Bob 到 i 的代理密钥: $\mathcal{B}$ 返回 $rk_{B\to i} = (g^{1/a})^{x_i}$.
    2. i 到 Bob 的代理密钥: $\mathcal{B}$ 返回 $rk_{i\to B} = (g^a)^{1/x_i}$.
- 输出: 当询问结束时, $\mathcal{A}$ 输出 Bob 的一个猜测私钥 sk'_B. 如果 $g^{sk'_B} = g^a$, 则 $\mathcal{B}$ 输出 sk'_B 作为自己对假设 2 的问题实例的解; 否则, 输出 $\mathbb{Z}_q$ 上的一个随机元素.

由于问题实例中 a, x_i 都是随机选取的, 所以 Bob 的公钥和其他用户的公钥的分布与实际选取的结果是一致的. 由于 $\mathcal{B}$ 知道 x_i 和 $g^{1/a}$, 所以模拟算法回答的代理密钥查询结果也是正确的. 因此, 如果 $\mathcal{A}$ 能够以不可忽略的概率输出 Bob 的私钥, 那么 $\mathcal{B}$ 就能够以相同的优势求解问题实例, 这与假设 2 相矛盾.

定理 6.9证毕! □

注记 6.12

定理 6.9说明了即使敌手与若干用户合谋, 并获取了从目标用户 Bob 授权的若干代理密钥, 也无法恢复 Bob 的私钥 sk_B. 这对于保护 Bob 的第一层次密文是有帮助的, 因为敌手仅能通过代理密钥访问 Bob 的第二层次密文, 而第一层次密文必须使用 Bob 的私钥才能解密. 那么, 主密钥安全性能否保障第一层次密文的语义安全性? 读者可以思考在主密钥攻击下第一层次密文的语义安全性模型, 并思考在该模型下如何设计一个安全的代理重加密方案.

通过前面两个定理的分析, 可以看出构造 6.10满足单向代理、非交互代理密钥、抗合谋攻击、非传递性、单跳性和代理不可见等优良的性质. 此外, 用户密钥也是紧致的, 与授权用户数量无关. 但是在安全性方面, 依赖的问题假设不够标准, 其困难性还是需要进一步论证. 依赖更加标准的问题假设构造代理重加密方案可以进一步阅读 Ateniese 等的改进方案 [360].

前面介绍的 PRE 方案都是选择明文不可区分的, 下面介绍一种 IND-CCA 安全的单向代理重加密方案 [363].

构造 6.11 (IND-CCA 安全的单向代理重加密方案)

- $\mathsf{Setup}(1^\kappa)$: 运行 $\mathsf{GenBLGroup}(1^\kappa)$ 生成 Type-I 双线性映射 $(\mathbb{G}, \mathbb{G}_T, q, g, e)$. 随机选择群元素 $h_1, h_2, u, v, w \xleftarrow{R} \mathbb{G}$. 选择 3 个密码学哈希函数, 分别是抗碰撞哈希函数 $\mathsf{H}_0 : \mathbb{G} \times \{0,1\}^l \to \mathbb{Z}_q$, 抗碰撞且单向哈希函数 $\mathsf{H}_1 : \mathbb{G}_T \to \{0,1\}^{l_1}$ 和一致哈希函数 $\mathsf{H}_2 : \mathbb{G} \to \{0,1\}^{l_2}$, 且满足 $l = l_1 + l_2 < \log q$. 令 $Z = e(h_1, h_2)$. 输出系统参数 $pp = (\mathbb{G}, \mathbb{G}_T, q, g, e, h_1, h_2, u, v, w, Z, \mathsf{H}_0, \mathsf{H}_1, \mathsf{H}_2)$.
- $\mathsf{KeyGen}(pp)$: 随机选择 $x_i, y_i \xleftarrow{R} \mathbb{Z}_q$, 计算公钥 $pk_i = (pk_{i,1} = h_1^{x_i}, pk_{i,2} = g^{y_i})$ 和私钥 $sk_i = (sk_{i,1} = x_i, sk_{i,2} = y_i)$. 输出用户 i 的公钥 pk_i 和私钥 sk_i.
- $\mathsf{ReKeyGen}(pk_A, sk_A, pk_B)$: 假设 $pk_A = (pk_{A,1}, pk_{A,2})$, $sk_A = (sk_{A,1}, sk_{A,2})$, $pk_B = (pk_{B,1}, pk_{B,2})$, 输出代理密钥 $rk_{A\to B} = (h_2 \cdot pk_{B,2})^{1/sk_{A,1}}$.
- 第一层次加密 $\mathsf{Encrypt}_1(pk_i, m)$: 对于消息 $m \in \{0,1\}^{l_2}$, 随机选择 $r, s \xleftarrow{R} \mathbb{Z}_q$, 计算 $c_0 = g^r$, $c_1 = Z^r \cdot e(h_1, pk_{i,2})^r$, $c_2 = \mathsf{H}_1(Z^r) || \mathsf{H}_2(Z^r) \oplus m$, $c_3 = (u^t v^s w)^r$, 其中 $t = \mathsf{H}_0(c_0, c_2)$. 输出密文 $c = (s, c_0, c_1, c_2, c_3)$.
- 第二层次加密 $\mathsf{Encrypt}_2(pk_i, m)$: 对于消息 $m \in \{0,1\}^{l_2}$, 随机选择 $r, s \xleftarrow{R} \mathbb{Z}_q$, 计算 $c_0 = g^r$, $c_1 = pk_{i,1}^r$, $c_2 = \mathsf{H}_1(Z^r) || \mathsf{H}_2(Z^r) \oplus m$, $c_3 = (u^t v^s w)^r$, 其中 $t = \mathsf{H}_0(c_0, c_2)$. 输出密文 $c = (s, c_0, c_1, c_2, c_3)$.
- 第一层次密文解密算法 $\mathsf{Decrypt}_1(sk_i, c_i)$: 假设 $sk_i = (sk_{i,1}, sk_{i,2})$ 和 $c_i = (s, c_0, c_1, c_2, c_3)$, 按以下步骤解密第一层次密文:
  1. 计算 $t = \mathsf{H}_0(c_0, c_2)$;
  2. 如果 $e(c_3, g) \neq e(u^t v^s w, c_0)$, 输出 $\perp$; 否则,
     (a). 拆分 $c_2 = \tau_1 || \tau_2$;
     (b). 计算 $K = c_1 / e(h_1^{sk_{i,2}}, c_0)$;
     (c). 如果 $\tau_1 = \mathsf{H}_1(K)$, 输出消息 $m = \tau_2 \oplus \mathsf{H}_2(K)$; 否则, 输出 $\perp$.
- 第二层次密文解密算法 $\mathsf{Decrypt}_2(sk, c)$: 假设 $sk_i = (sk_{i,1}, sk_{i,2})$ 和 $c_i = (s, c_0, c_1, c_2, c_3)$, 按以下步骤解密第二层次密文:
  1. 计算 $t = \mathsf{H}_0(c_0, c_2)$;
  2. 如果 $e(c_3, g) \neq e(u^t v^s w, c_0)$ 或 $e(c_1, g) \neq e(pk_{A,1}, c_0)$, 则输出 $\perp$; 否则,
     (a). 拆分 $c_2 = \tau_1 || \tau_2$;
     (b). 计算 $K = e(c_1, h_2^{1/sk_{i,1}})$;
     (c). 如果 $\tau_1 = \mathsf{H}_1(K)$, 输出消息 $m = \tau_2 \oplus \mathsf{H}_2(K)$; 否则, 输出 $\perp$.
- $\mathsf{ReEncrypt}(rk_{A\to B}, c_A)$: 假设 $c_A = (s, c_0, c_1, c_2, c_3)$ 是第二层次密文, 按以下步骤计算重加密密文:
  1. 计算 $t = \mathsf{H}_0(c_0, c_2)$;
  2. 如果 $e(c_3, g) \neq e(u^t v^s w, c_0)$ 或 $e(c_1, g) \neq e(pk_{A,1}, c_0)$, 则输出 $\perp$; 否则,
     (a). 计算 $c_1' = e(c_1, rk_{A\to B}) = Z^r \cdot e(h_1, pk_{B,2})^r$;

(b). 输出 $c_B = (s, c_0, c_1', c_2, c_3)$.

笔记 在构造 6.11 中, 无论是解密算法还是重加密算法, 都需要通过 $e(c_3, g) \stackrel{?}{=} e(u^t v^s w, c_0)$ 或 $e(c_1, g) = e(pk_{A,1}, c_0)$ 检验密文的完整性. 通过检验后, 不同层次密文的解密算法都可以正确地计算 $K = Z^r$, 继而恢复原始消息 m. 因此, 方案的正确性可以得到保证.

一般地, 抗选择密文攻击的方案必须具有密文不可延展性, 否则敌手可以对挑战密文进行篡改, 通过解密询问获取与原始消息相关的解密结果. 在上述构造中, 无论是第一层次密文还是第二层次密文, 如果 c_0 和 c_2 发生了变化, 那么直接导致 $t = \mathsf{H}_0(c_0, c_2)$ 发生改变. 实际上, (c_0, c_3) 可以看作是一个支持完整性公开验证 (双线性映射计算) 的可提取哈希证明系统, 不仅用于密文的完整性验证, 而且在安全性证明中还用于模拟解密查询. 对于第二层次密文 c_1 的完整性通过双线性映射计算来检验, 从而保持与 c_0 具有相同的随机数 r. 对于第一层次密文, 如果 c_1 发生了改变, 那么解密算法恢复的 K 等于 Z^r 的概率是可忽略的, 从而无法通过 $\tau_1 = \mathsf{H}_1(K)$ 的检验.

构造 6.11 的抗选择密文攻击安全性可通过下面的定理 6.10 保证, 其详细的安全性分析请参考文献 [363].

定理 6.10

如果 DBDH 假设成立, 密码学哈希函数 H_0 是抗碰撞的, H_1 是抗碰撞且单向的, H_2 满足一致性, 则构造 6.11 是一个 IND-CCA 安全的单向代理重加密方案.

文献 [363] 中描述的代理重加密方案的 IND-CCA 安全性与本书定义的 IND-CCA 安全性有所不同. 文献 [363] 采用 CK (chosen key) 安全模型. 该模型允许敌手替换用户的公钥 (可能是敌手选择的公钥, 也可能是其他用户的公钥). 而本书定义的 IND-CCA 安全性采用的是 KOSK (knowledge of secret key) 安全模型. 该模型要求挑战者生成所有用户的公钥并且必须知道恶意用户的私钥. 实际上, 许多 PRE 方案都是在 KOSK 模型中设计的. 读者可思考这两种模型之间的本质区别. 此外, 读者还可以进一步思考, 如果用户 A 的第二层次密文经过重加密后转化为用户 B 的一个合法的第一层次密文, 那么, 如何防止敌手询问重加密密文的解密谕言机. 关于代理重加密的选择密文攻击安全性和自适应安全性可以进一步阅读文献 [364, 365, 362]. 关于代理重加密的其他性质, 如抗量子安全性、细粒度访问控制等, 可以阅读文献 [366, 367, 368].

第七章 标准化及工程实践

章前概述

内容提要

- 公钥加密的标准化工作
- 公钥加密的工程实践

本章开始介绍公钥加密的标准化与工程实践. 7.1节简介与公钥加密有关的标准化组织和标准化进展, 7.2节介绍了公钥加密在工程实践方面需要注意的事项.

7.1 公钥加密的标准化

今天下, 车同轨, 书同文, 行同伦.

—《礼记 · 中庸》

标准化对密码技术的实际落地应用具有重要意义, 否则, 即使是同一密码方案/协议也可能由于参数选取、接口设计缺乏统一的规范而无法互联互通. 以下首先简要介绍与密码领域相关较为密切的国内外标准化组织.

7.1.1 国内外标准化组织简介

1. 国际标准化组织与国际电工委员会
   国际标准化组织 (ISO) 与国际电工委员会 (IEC) 联合成立了名为 ISO/IEC JTC 1 的委员会, 重点关注信息技术领域的标准化, 联合制定了一系列 ISO/IEC 标准, 其中的 ISO/IEC 18033 系列标准规定了加密算法、密码协议和密钥管理技术. ISO/IEC 标准通常由世界各地的成员国共同参与制定, 涉及多轮草案和投票, 标准化过程严格, 需经过彻底的审查和意见反馈与修订. 正因如此, ISO/IEC 标准具有广泛的国际认可度, 在实施全球化技术和安全政策方面均具有强大的影响力, 可确保来自不同供应商的产品和服务可以安全有效地协同工作. 符合 ISO/IEC 标准的密码产品通常质量和安全方面具有较高的置信度, 这对于金融交易、医疗保健和国家安全等关键应用至关重要.
2. 互联网工程任务组
   互联网工程任务组 (IETF) 是一个开放的标准组织, 负责开发和推广互联网标准, 特别是维护 TCP/IP 协议族的标准. 与 ISO/IEC 不同, 它不依赖于任何特定国家或管理机构, 没有正式的会员资格或会员资格要求. IETF 将其技术文档发布为征求意见稿 RFC (Requests for Comments), IETF 制定的 RFC 全方位涵盖了计算机网络体系, 在安全性与隐私方面, IETF 制定的技术标准和实践文档致力于抵御已知和新出现的威胁, 为互联网的安全和隐私提供了重要的基础要素. IETF 针对安全方面正在进行的一些工作包括: 最新版本的传输层安全协议 TLS 1.3、自动证书管理环境协议 (最近发布为 RFC 8555) 和消息传递分层安全协议等. IETF 标准具有高度的包容性, 任何人均可参与到标准制定的过程中, 且标准制定更多的基于实施和部署规范方面的实际经验, 强调实用性和执行性. IETF 标准在万物互联互通中起到了至关重要的作用, 标准化的通信协议确保不同的系统可以无缝地协同工作, SSL/TLS 等就是 IETF 标准的典范工作.
3. 美国电气电子工程师学会
   美国电气电子工程师学会 (Institute of Electrical and Electronics Engineers, IEEE) 的标准化组织为推出了公钥密码学标准 IEEE P1363. 该标准包括传统公钥密码学 (IEEE Std 1363-2000 and 1363a-2004)、格基公钥密码学 (IEEE Std 1363.1-2008)、口令基公钥密码学 (IEEE Std 1363.2-2008)、使用双线性映射的公钥密码学 (IEEE Std 1363.3-2013).
4. 中国国家标准局
   中国国家标准局现称为中国国家标准化管理委员会 (SAC), 是中国国务院直属的政府机构, 它负责起草和管理国家标准, 并代表中国加入 ISO/IEC 等国际标准组织. 中国国家标准局一直积极制定信息安全国家标准, 包括密码算法和协议. SAC 的标准对国内行业具有重大影响, 并且经常被用作中国境内法规的基础. 随着中国在全球贸易中的地位不断提升, SAC 标准在国际上的影响力也越来越高.
5. 美国国家标准局
   美国国家标准与技术研究院 (National Institute of Standards and Technology, NIST) 成立于 1901 年, 现隶属于美国商务部. NIST 是美国最古老的物理科学实验室之一, 成立之初的目的是消除当时美国工业在测量基础设施方面的短板. 当前, 从智能电网和电子健康记录到原子钟、先进纳米材料和计算机芯片, 无数产品和服务都在某种程度上依赖于 NIST 提供的技术、测量和标准. NIST 致力于制定与信息技术各个方面相关的标准和指南, 还专门为联邦机构和广大公众制定密码标准和指南, 包括哈希算法、随机数生成算法、加密方案、签名方案和后量子密码方案等. 尽管 NIST 是美国机构, 但具有国际影响力, 其标准和指南不仅被美国联邦机构广泛采用, 还被私营部门组织和全球其他政府广泛采用.

6. 美国国家标准学会
   美国国家标准学会 (American National Standards Institute, ANSI) 成立于 1918 年, 是美国非盈利民间标准化团体. 作为自愿性标准体系中的协调中心, ANSI 的主要职能是协调国内各机构团体的标准化活动、审核批准美国国家标准、代表美国参加国际标准化活动、提供标准信息咨询服务等. 在密码学领域, 该组织制定了基于椭圆曲线的公钥密码学标准 ANSI X9.63.
7. RSA 公司
   1990 年起, RSA 公司发布了一系列公钥密码技术标准 (Public Key Cryptography Standards, PKCS), 旨在推广公司拥有专利的密码算法, 如 RSA 加密算法与签名、Schnorr 签名等. 尽管 PKCS 系列不是工业标准, 但其中的部分算法已经在纳入若干标准化组织 (如 IETF 和 PKIX 工作组) 的正式标准进程中.

7.1.2 公钥加密标准方案

选择密文安全常简称为 CCA 安全, 自上世纪 90 年代起即成为公钥加密的事实标准 (de facto standard), 正因如此, 绝大多数标准化组织制定的公钥加密标准均具备选择密文安全. 以下首先介绍基于数论类假设的公钥加密标准方案.

- 基于整数分解类困难问题的公钥加密方案
  PKCS#1 [369] 是 PKCS 系列标准中最早也应用最广泛的一个, 制定了 RSA 加密和签名标准, 最新的版本号为 v2.2. PKCS#1 中定义了 RSA 公钥和私钥应如何表示和存储, 规定了基本的 RSA 操作, 包括加密和解密、签名和验证. 特别的, 标准中为 RSA 加密方案引入填充机制 OAEP (Optimal Asymmetric Encryption Padding), 得到可证明 IND-CCA 安全的 RSA-OAEP, 解决了早期版本中存在的安全问题, 如针对 PKCS#1 v1.5 填充的自适应选择明文攻击和 Bleichenbacher 攻击.
- 基于离散对数类困难问题的公钥加密方案
  离散对数类困难问题根据代数结构的不同, 划分为数域和椭圆曲线两个子类, 在同样的安全级别下, 后者的参数规模更为紧致, 因此构建于其上的密码方案相比前者具有显著的性能优势, 但是由于数学结构复杂, 工程实践的难度也更大. DHIES (Diffe-Hellman Integrated Encryption Scheme, DHIES) 是 DHAES [370] 的标准化方案, 采用混合加密方式, 密钥封装机制基于数域循环群上的 ElGamal PKE 和哈希函数构造, 数据封装机制基于消息验证码和堆成加密方案构造. DHIES 整体方案在随机谕言机模型中基于 CDH 假设具备可证明的选择密文安全. ECIES (Elliptic Curve Encryption Scheme, ECIES) 是 DHIES 在椭圆曲线循环群上的对于版本. DHIES 和 ECIES 被纳入 IEEE 1363a、ANSI X9.63 和 ISO/IEC 18033-2 标准. ECIES 还被椭圆曲线密码标准组 (Standards for Efficient Cryptography Group, SECG) 纳入到椭圆曲线密码学标准 SEC 1 [371] 中.
  NIST 在联邦信息处理标准 (Federal Information Processing Standards Publication)FIPS 186-5 [372]、SECG 在 SEC 2 [373] 和 ECC Brainpool 在 RFC 5639 [374] 中分别给出了推荐的椭圆曲线参数选择. 中国密码管理局为满足国内电子认证服务系统等应用需求, 于 2010 年 12 月 17 日发布了《SM2 椭圆曲线公钥密码算法》[375], 2016 年成为中国国家密码标准. SM2 标准中包括推荐椭圆曲线参数和包括公钥加密方案在内的各种类型公钥密码方案.
- 基于格类困难问题的公钥加密方案
  Shor 算法的出现意味着在后量子时代基于数论类困难问题的密码方案将不再安全, 因此设计能够抵抗量子攻击的密码方案成为当前密码学的前沿热点, 其中格基方案是后量子安全密码学中的主流. NIST 自 2016 年开始了后量子密码学标准方案的征集. 经过最新一轮的评审, NIST 于 2023 年 8 月 24 号发布了 3 个 FIPS 草案拟定了后量子密码系列方案, 其中 FIPS 203 [376] 定义了基于 LWE 困难问题的公钥加密方案 CRYSTALS-KYBER [377].

如前所述, 绝大多数标准中的公钥加密方案都满足 IND-CCA 安全. 然而, IND-CCA 安全与同态性无法共存, 在分布式计算环境和大数据应用等密态数据的可操作性比机密性保护更重要的场景中, 迫切需要标准化的同态公钥加密方案.

- 部分同态加密方案标准

ISO/IEC 18033-6 [378] 标准中定义了 Exponential ElGamal 和 Paillier [281] 两个加法同态加密方案.

- 全同态加密方案标准

  全同态加密尚处于飞速发展阶段, 然而工业界的应用需求更为迫切. 2017 年, 来自 IBM、Microsoft、Intel 和 NIST 和其它开放组织的研究人员共同成立了全同态标准化联盟 (Homomorphic Encryption Standardization Consortium), 并发布了同态加密标准文档 [379]. 该文档涵盖了适用于整数运算的 Brakerski-Gentry-Vaikuntanathan (BGV) [380] 和 Brakerski/Fan-Vercauteren (BFV) [381, 382]、适用于浮点数运算的 Cheon-Kim-Kim-Song (CKKS) [383] 以及适用于 Boolean 电路求值的 Ducas-Micciancio (FHEW) [384] 和 Chillotti-Gama-Georgieva-Izabachene (TFHE) [385]. 该文档尽管不是官方标准, 但基本可以看成事实上的标准.

7.2 公钥加密的工程实践

纸上得来终觉浅, 绝知此事要躬行.

— 宋 · 陆游《冬夜读书示子聿》

实现密码算法对程序员的素质要求较高, 既需要专业的密码知识以确保实现的忠实性和安全性, 也需要精湛的编程技术以确保实现的效率. 在一般情况下, 不建议非专业程序员自行从底层起构建密码算法, 如此不仅可省去重复制造轮子的无用功, 更能避免造出方形轮子的错误.

7.2.1 重要方案的优秀开源实现

工程实践中经常需要使用已有的公钥加密方案, 以下推荐部分常用方案的优秀开源实现供一线程序员按图索骥.

标准公钥加密方案

- RSA-OAEP: OpenSSL 库 [386] 中提供了 C 语言版本的实现.
- Paillier: mpc4j 库 [387] 提供了 Java 语言的实现.
- ElGamal: Kunlun 库 [388] 中给出了 ElGamal PKE 及其多个衍生方案的 C++ 实现, 同时给出了配套的零知识证明实现, 可直接部署应用于密态计算场景.

属性加密

- FAME (Fast Attribute-based Message Encryption) [389]: 首个基于标准假设完全安全的密文策略和密钥策略 ABE 方案 (对策略类型或属性没有任何限制), 构建于 Type-III 双线性映射上. 相应的开源实现可参考: https://github.com/sagrawal87/ABE

全同态加密: Microsoft 的 SEAL (Simple Encrypted Arithmetic Library) 库 [390] 给出了 BGV、BFV 和 CKKS 方案的优秀实现, PALISADE 的后继者 OpenFHE [391] 则包含了所有主流全同态加密方案的实现.

7.2.2 重要的开源密码库

下面的内容适用于程序员在实现自研公钥加密方案时, 为如何选择合适的密码算法库做出参考.

表 7.1: 常用开源密码算法库

库名	编程语言	支持算子类型				易用性	实时性	国密算法支持
		对称密码	大整数运算	椭圆曲线	双线性映射			
OpenSSL	C	✓	✓	✓	✗	★★★★★	★★★★★	SM2/SM3/SM4
tongsuo	C	✓	✓	✓	✗	★★★★★	★★★★★	SM2/SM3/SM4
gmSSL	C	✓	✓	✓	✗	★★★	★★★	SM2/SM3/SM4/SM9/ZUC
mcl	C/C++	✓	✓	✓	✓	★★★	★★★★★	—
MIRACL	C/C++	✓	✓	✓	✓	★★★★★	★★★	—
NTL	C++	✗	✓	✗	✗	★★★★★	★★★★★	—
Bouncy Castle	Java/C#	✓	✓	✓	✓	★★★★★	★★★★★	SM2/SM3/SM4
Crypto++	C++	✓	✓	✓	✓	★	★★★★	SM3/SM4
Botan	C++	✓	✓	✓	✗	★★★★★	★★★★★	SM2/SM3/SM4
libsodium	C	✗	✓	✓	✗	★★★	★★	—
libgcrypt	C	✗	✓	✓	✗	★★★★	★★★★	SM2/SM3/SM4

7.3 公钥加密的工程实践经验

工程实现是公钥加密应用中至关重要的一环, 需要同时关注效率与安全两个维度.

效率方面: 公钥加密方案涉及的运算往往较为复杂, 针对不同实现平台的优化策略也大相径庭.

- 软件实现中, 开发者需要充分利用现代处理器提供的指令集以高效实现各类公钥密码方案, 技术难点在于如何基于有限的指令来完成复杂多样的公钥操作, 如大数模乘、模幂、椭圆曲线运算、NTT/FFT 等.
- 硬件实现中, 开发者需要使用足够少资源 (如电路门、存储单元等) 来实现各类公钥密码方案, 并且达到相应的技术指标 (如吞吐量、延迟、功耗等), 技术难点在于如何在运行效率和资源开销之间取得最佳平衡.

安全方面: 需要结合应用场景和部署环境避免以下问题

- 密码误用: 开发者没有正确实现或者调用密码算法产生的安全漏洞统称为密码误用, 这也是最为常见的应用错误. 实现中容易忽视的细节包括:
  - nonce 的使用: 顾名思义, nonce 仅可使用一次, 若使用超过一次则方案不再安全.
  - 随机数的使用: 随机数是现代密码学的核心要素. 若随机数被泄漏或者分布不均匀, 密码方案/协议的安全性将不再成立. 在密码方案/协议的理论设计过程中, 密码学家通常假设真随机数的存在且易获得. 在工程实现中, 则需基于可靠的熵源调用安全的真或伪随机数发生器 (Random Number Generator, RNG) 产生随机数, 切不可使用不可靠的熵源或者使用非密码学意义的随机数发生器, 也不可随意复用随机数.
  - 哈希函数的实例化: 切不可使用非密码学哈希函数代替密码学哈希函数, 在需要将哈希函数建模为随机谕言机时 (如 hash-to-EC-point 函数), 也务必确保哈希函数的像不泄漏原像的代数结构.
  - 密码算法的选择: 切勿使用不再安全的密码算法.
  - 密码算法的设计: 缺乏专业技能的密码开发者尽量避免自行设计密码算法. 一个常见的错误认知是若各密码组件安全, 则任意组合的综合方案也安全. 事实并非如此, 一个典型的例子便是密钥对复用与分离策略. 在使用多个带密钥密码组件构建密码方案/协议时, 默认的原则密钥分离, 即各组件独立生成各自的密钥. 然而若未经专门的设计, 多密码组件重用密钥可能会导致方案/协议不再安全, 如复用 RSA 加密方案和签名方案的密钥对将导致联合方案不再安全.

  应对密码误用可从两方面入手, 一是可使用相关工具检测常见的密码误用, 二是加强对密码开发者的密码专业知识培训来杜绝密码误用.
- 侧信道攻击: 传统安全模型假设密码算法的软硬件实现完美黑盒, 即敌手仅能以黑盒的方式通过预定义的接口收集信息以分析密码算法, 而无法通过探测或篡改等侧信道攻击方式获取内部秘密信息 (如私钥、随机数). 然而密码的工程实现无法达到完美黑盒, 敌手可实施种类繁多的被动或主动侧信道攻击. 可证明安全的抗侧信道攻击密码方案通常效率低下, 无法满足生产实际的效率要求. 广泛部署应用的仍是普通方案, 因此在工程实现中需要充分考虑侧信道防御机制.
  - 被动式侧信道攻击防护: 被动式侧信道攻击指敌手通过分析运行时间/能耗/电磁辐射、冷启动读取内存等被动方法获取侧信道信息, 对目标密码算法实施泄漏攻击. 抗泄漏攻击的主要防御手段是掩码和隐藏技术. 掩码技术的思想是把运行中涉及的所有秘密信息随机分片, 使得敌手必须获得足够的分片才能恢复秘密信息. 掩码技术实现开销较大, 但可基于特定物理假设可证明安全. 然而, 目前的掩码技术主要用于对称密码算法的防护, 并不能很好地适用于运算更为复杂的公钥密码算法. 例如基于格的密码算法中往往涉及到运算较为复杂的高斯采样, 如何在掩码分片上高效地执行高斯采样目前仍然是一个挑战. 隐藏技术的思想是通过随机延迟、乱序执行、双轨逻辑等方法隐藏秘密信息通过侧信道的泄漏, 本质上是在一定程度上增加了泄漏的噪声. 隐藏技术大致分为算法防护和硬件体系防护, 前者主要是修改密码算法的执行流程, 如随机延迟或打乱执行顺序等; 后者主要是在硬件层面上的措施, 往往对软件实现人员透明, 典型的方法是设计新低泄漏的电路逻辑器件或进入随机噪声源等. 隐藏技术只能在一定程度上增加敌手的分析难度, 但具有设计灵活、实现开销较小的特点, 是目前业界针对公钥密码算法的主流选择.
  - 主动式侧信道攻击防护: 主动式侧信道攻击指敌手通过故障植入、截断线路、加热冷冻、射线照射等主动方法获取侧信道信息, 对目标密码算法实施篡改攻击. 抗篡改攻击的实际防御手段包括环境监测、电路隐藏、混淆冗余、多次运行、故障修复码等方法. 前两种方法属于硬件层面的防护策略, 其中环境监测通过加入各类传感器从而主动发现故障注入, 电路隐藏则通过利用特殊的布局布线策略 (如使用

3D 芯片或多层 PCB 技术隐藏密码算法执行的硬件功能模块), 使攻击者很难精确注入有效的故障. 后面三类方法属于软件层面的防护策略. 与被动式攻击防护不同的是, 目前主动式攻击在软件层面上的防护远少于在硬件层面上的防护, 主要原因是故障攻击的实施难度较大.

应对侧信道攻击的方法是优先选择高质量的开源代码实现, 自行开发时需精心实现、严格评估, 同时应根据具体应用场景综合采用软硬结合的实现策略.

参考文献

[1] Claude E Shannon. “A mathematical theory of cryptography”. In: *Mathematical Theory of Cryptography* (1945).

[2] Horst Feistel. “Cryptography and computer privacy”. In: *Scientific american* 228.5 (1973), pp. 15–23.

[3] Whitefield Diffie and Martin E. Hellman. “New Directions in Cryptograpgy”. In: *IEEE Transactions on Infomation Theory* 22(6) (1976), pp. 644–654.

[4] Ron Rivest, Adi Shamir, and Leonard Adleman. “A Method for Obtaining Digital Signatures and Public Key Cryptosystems”. In: *Communications of the ACM* 21(2) (February 1978), pp. 120–126.

[5] Shafi Goldwasser and Silvio Micali. “Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information”. In: *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, STOC 1982*. ACM, 1982, pp. 365–377.

[6] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. “The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)”. In: *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, STOC 1985*. ACM, 1985, pp. 291–304.

[7] Andrew Chi-Chih Yao. “How to Generate and Exchange Secrets (Extended Abstract)”. In: *27th Annual Symposium on Foundations of Computer Science, FOCS 1986*. IEEE Computer Society, 1986, pp. 162–167.

[8] Shengli Liu. “公钥加密系统的可证明安全——新挑战新方法”. In: *密码学报* 1.6 (2014), pp. 537–550.

[9] Moni Naor and Moti Yung. “Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks”. In: *Proceedings of the 22th Annual ACM Symposium on Theory of Computing, STOC 1990*. ACM, 1990, pp. 427–437.

[10] Daniel Bleichenbacher. “Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1”. In: *Advances in Cryptology - CRYPTO 1998*. Vol. 1462. Lecture Notes in Computer Science. Springer, 1998, pp. 1–12.

[11] Mihir Bellare and Phillip Rogaway. “Random Oracles are Practical: A Paradigm for Designing Efficient Protocols”. In: *1st ACM Conference on Computer and Communications Security*. 1993, pp. 62–73.

[12] Mihir Bellare and Phillip Rogaway. “Optimal Asymmetric Encryption - How to Encrypt with RSA”. In: *Advances in Cryptology - EUROCRYPT 1995*. Vol. 950. LNCS. Springer, 1995, pp. 92–111.

[13] Victor Shoup. “OAEP Reconsidered”. In: *Advances in Cryptology - CRYPTO 2001*. Vol. 2139. Lecture Notes in Computer Science. Springer, 2001, pp. 239–259.

[14] Eiichiro Fujisaki et al. “RSA-OAEP Is Secure under the RSA Assumption”. In: *Advances in Cryptology - CRYPTO 2001*. Vol. 2139. Lecture Notes in Computer Science. Springer, 2001, pp. 260–274.

[15] Eiichiro Fujisaki and Tatsuaki Okamoto. “How to Enhance the Security of Public-Key Encryption at Minimum Cost”. In: *Public Key Cryptography - PKC 1999*. Vol. 1560. LNCS. 1999, pp. 53–68.

[16] Eiichiro Fujisaki and Tatsuaki Okamoto. “Secure Integration of Asymmetric and Symmetric Encryption Schemes”. In: *Advances in Cryptology - CRYPTO 1999*. Vol. 1666. LNCS. 1999, pp. 537–554.

[17] Eiichiro Fujisaki and Tatsuaki Okamoto. “Secure Integration of Asymmetric and Symmetric Encryption Schemes”. In: *J. Cryptol.* 26.1 (2013), pp. 80–101.

[18] Tatsuaki Okamoto and David Pointcheval. “REACT: Rapid Enhanced-Security Asymmetric Cryptosystem Transform”. In: *Topics in Cryptology - CT-RSA 2001*. Vol. 2020. Lecture Notes in Computer Science. Springer, 2001, pp. 159–175.

[19] Haodong Jiang et al. “IND-CCA-Secure Key Encapsulation Mechanism in the Quantum Random Oracle Model, Revisited”. In: *Advances in Cryptology - CRYPTO 2018*. Vol. 10993. Lecture Notes in Computer Science. Springer, 2018, pp. 96–125.

[20] Nina Bindel et al. “Tighter Proofs of CCA Security in the Quantum Random Oracle Model”. In: *Theory of Cryptography - 17th International Conference, TCC 2019*. Vol. 11892. Lecture Notes in Computer Science. Springer, 2019, pp. 61–90.

[21] Loïs Huguenin-Dumittan and Serge Vaudenay. “On IND-qCCA Security in the ROM and Its Applications - CPA Security Is Sufficient for TLS 1.3”. In: *Advances in Cryptology - EUROCRYPT 2022*. Vol. 13277. Lecture Notes in Computer Science. Springer, 2022, pp. 613–642.

[22] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. “A Modular Analysis of the Fujisaki-Okamoto Transformation”. In: *Theory of Cryptography - 15th International Conference, TCC 2017*. Vol. 10677. Lecture Notes in Computer Science. Springer, 2017, pp. 341–371.

[23] Danny Dolev, Cynthia Dwork, and Moni Naor. “Non-Malleable Cryptography (Extended Abstract)”. In: *STOC*. ACM, 1991, pp. 542–552.

[24] Amit Sahai. “Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security”. In: *FOCS 1999*. ACM, 1999, pp. 543–553.

[25] Silvio Biagioni, Daniel Masny, and Daniele Venturi. “Naor-Yung Paradigm with Shared Randomness and Applications”. In: *Security and Cryptography for Networks - 10th International Conference, SCN 2016*. Vol. 9841. Lecture Notes in Computer Science. Springer, 2016, pp. 62–80.

[26] Ronald Cramer, Dennis Hofheinz, and Eike Kiltz. “A Twist on the Naor-Yung Paradigm and Its Application to Efficient CCA-Secure Encryption from Hard Search Problems”. In: *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*. Vol. 5978. LNCS. Springer, 2010, pp. 146–164.

[27] Ronald Cramer and Victor Shoup. “Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption”. In: *Advances in Cryptology - EUROCRYPT 2002*. 2002, pp. 45–64.

[28] Hoeteck Wee. “Efficient Chosen-Ciphertext Security via Extractable Hash Proofs”. In: *Advances in Cryptology - CRYPTO 2010*. Vol. 6223. 2010, pp. 314–332.

[29] Dan Boneh et al. “Chosen-Ciphertext Security from Identity-Based Encryption”. In: *SIAM Journal on Computation* 36.5 (2007), pp. 1301–1328.

[30] Eike Kiltz. “On the Limitations of the Spread of an IBE-to-PKE Transformation”. In: *Public Key Cryptography - PKC 2006*. Vol. 3958. LNCS. Springer, 2006, pp. 274–289.

[31] Chris Peikert and Brent Waters. “Lossy trapdoor functions and their applications”. In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008*. 2008, pp. 187–196.

[32] Alon Rosen and Gil Segev. “Chosen-Ciphertext Security via Correlated Products”. In: *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*. Vol. 5444. LNCS. Springer, 2009, pp. 419–436.

[33] Eike Kiltz, Payman Mohassel, and Adam O’Neill. “Adaptive Trapdoor Functions and Chosen-Ciphertext Security”. In: *Advances in Cryptology - EUROCRYPT 2010*. 2010, pp. 673–692.

[34] Susan Hohenberger, Venkata Koppula, and Brent Waters. “Chosen Ciphertext Security from Injective Trapdoor Functions”. In: *Advances in Cryptology - CRYPTO 2020*. Vol. 12170. Lecture Notes in Computer Science. Springer, 2020, pp. 836–866.

[35] Amit Sahai and Brent Waters. “How to use indistinguishability obfuscation: deniable encryption, and more”. In: *Symposium on Theory of Computing, STOC 2014*. ACM, 2014, pp. 475–484.

[36] Yu Chen and Zongyang Zhang. “Publicly evaluable pseudorandom functions and their applications”. In: *Journal of Computer Security* 24.2 (2016), pp. 289–320.

[37] Taher ElGamal. “A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms”. In: *IEEE Transactions on Information Theory* 31 (1985), pp. 469–472.

[38] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. “Possibility and Impossibility Results for Encryption and Commitment Secure under Selective Opening”. In: *Advances in Cryptology - EUROCRYPT 2009*. Vol. 5479. LNCS. Springer, 2009, pp. 1–35.

[39] Mihir Bellare et al. “Standard Security Does Not Imply Security against Selective-Opening”. In: *Advances in Cryptology - EUROCRYPT 2012*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 645–662.

[40] Brett Hemenway et al. “Lossy Encryption: Constructions from General Assumptions and Efficient Selective Opening Chosen Ciphertext Security”. In: *Advances in Cryptology - ASIACRYPT 2011*. Vol. 7073. LNCS. Springer, 2011, pp. 70–88.

[41] Dennis Hofheinz. “All-But-Many Lossy Trapdoor Functions”. In: *Advances in Cryptology - EUROCRYPT 2012*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 209–227.

[42] Serge Fehr et al. “Encryption Schemes Secure against Chosen-Ciphertext Selective Opening Attacks”. In: *Advances in Cryptology - EUROCRYPT 2010*. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 381–402.

[43] Zhengan Huang, Shengli Liu, and Baodong Qin. “Sender-Equivocable Encryption Schemes Secure against Chosen-Ciphertext Attacks Revisited”. In: *Public-Key Cryptography - PKC 2013*. Vol. 7778. Lecture Notes in Computer Science. Springer, 2013, pp. 369–385.

[44] Xavier Boyen and Qinyi Li. “All-But-Many Lossy Trapdoor Functions from Lattices and Applications”. In: *Advances in Cryptology - CRYPTO 2017*. Vol. 10403. Lecture Notes in Computer Science. Springer, 2017, pp. 298–331.

[45] Benoît Libert et al. “All-But-Many Lossy Trapdoor Functions and Selective Opening Chosen-Ciphertext Security from LWE”. In: *Advances in Cryptology - CRYPTO 2017*. Vol. 10403. Lecture Notes in Computer Science. Springer, 2017, pp. 332–364.

[46] Carmit Hazay, Arpita Patra, and Bogdan Warinschi. “Selective Opening Security for Receivers”. In: *Advances in Cryptology - ASIACRYPT 2015*. Vol. 9452. Lecture Notes in Computer Science. Springer, 2015, pp. 443–469.

[47] Dingding Jia, Xianhui Lu, and Bao Li. “Constructions Secure Against Receiver Selective Opening and Chosen Ciphertext Attacks”. In: *Topics in Cryptology - CT-RSA 2017*. Vol. 10159. Lecture Notes in Computer Science. Springer, 2017, pp. 417–431.

[48] Keisuke Hara et al. “Simulation-Based Receiver Selective Opening CCA Secure PKE from Standard Computational Assumptions”. In: *Security and Cryptography for Networks - 11th International Conference, SCN 2018*. Ed. by Dario Catalano and Roberto De Prisco. Vol. 11035. Lecture Notes in Computer Science. Springer, 2018, pp. 140–159.

[49] Junzuo Lai et al. “Simulation-Based Bi-Selective Opening Security for Public Key Encryption”. In: *Advances in Cryptology - ASIACRYPT 2021*. Vol. 13091. Lecture Notes in Computer Science. Springer, 2021, pp. 456–482.

[50] Mihir Bellare and Scott Yilek. “Encryption Schemes Secure under Selective Opening Attack”. In: *IACR Cryptol. ePrint Arch.* (2009), p. 101. URL: http://eprint.iacr.org/2009/101.

[51] Dan Boneh et al. “Circular-Secure Encryption from Decision Diffie-Hellman”. In: *Advances in Cryptology - CRYPTO 2008*. Vol. 5157. LNCS. Springer, 2008, pp. 108–125.

[52] Jan Camenisch and Anna Lysyanskaya. “An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation”. In: *Advances in Cryptology - EUROCRYPT 2001*. Springer, 2001, pp. 93–118.

[53] John Black, Phillip Rogaway, and Thomas Shrimpton. “Encryption-Scheme Security in the Presence of Key-Dependent Messages”. In: *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002*. Vol. 2595. LNCS. Springer, 2002, pp. 62–75.

[54] Benny Applebaum et al. “Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems”. In: *Advances in Cryptology - CRYPTO 2009*. Vol. 5677. LNCS. Springer, 2009, pp. 595–618.

[55] Tal Malkin, Isamu Teranishi, and Moti Yung. “Efficient Circuit-Size Independent Public Key Encryption with KDM Security”. In: *Advances in Cryptology - EUROCRYPT 2011*. Vol. 6632. LNCS. Springer, 2011, pp. 507–526.

[56] Hoeteck Wee. “KDM-Security via Homomorphic Smooth Projective Hashing”. In: *Public-Key Cryptography - PKC 2016*. Vol. 9615. LNCS. Springer, 2016, pp. 159–179.

[57] Zvika Brakerski, Shafi Goldwasser, and Yael Tauman Kalai. “Black-Box Circular-Secure Encryption beyond Affine Functions”. In: *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*. Vol. 6597. LNCS. Springer, 2011, pp. 201–218.

[58] Jan Camenisch, Nishanth Chandran, and Victor Shoup. “A Public Key Encryption Scheme Secure against Key Dependent Chosen Plaintext and Adaptive Chosen Ciphertext Attacks”. In: *Advances in Cryptology - EUROCRYPT 2009*. Vol. 5479. LNCS. Springer, 2009, pp. 351–368.

[59] Shuai Han, Shengli Liu, and Lin Lyu. “Efficient KDM-CCA Secure Public-Key Encryption for Polynomial Functions”. In: *Advances in Cryptology - ASIACRYPT 2016*. Vol. 10032. LNCS. Springer, 2016, pp. 307–338.

[60] Fuyuki Kitagawa and Keisuke Tanaka. “A Framework for Achieving KDM-CCA Secure Public-Key Encryption”. In: *Advances in Cryptology - ASIACRYPT 2018*. Vol. 11273. Lecture Notes in Computer Science. Springer, 2018, pp. 127–157.

[61] Fuyuki Kitagawa, Takahiro Matsuda, and Keisuke Tanaka. “Simple and Efficient KDM-CCA Secure Public Key Encryption”. In: *Advances in Cryptology - ASIACRYPT 2019*. Vol. 11923. Lecture Notes in Computer Science. Springer, 2019, pp. 97–127.

[62] Fuyuki Kitagawa and Takahiro Matsuda. “CPA-to-CCA Transformation for KDM Security”. In: *Theory of Cryptography - TCC 2019*. Vol. 11892. Lecture Notes in Computer Science. Springer, 2019, pp. 118–148.

[63] Brent Waters and Daniel Wichs. “Universal Amplification of KDM Security: From 1-Key Circular to Multi-Key KDM”. In: *Advances in Cryptology - CRYPTO 2023*. Vol. 14082. Lecture Notes in Computer Science. Springer, 2023, pp. 674–693.

[64] Silvio Micali and Leonid Reyzin. “Physically Observable Cryptography (Extended Abstract)”. In: *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*. 2004, pp. 278–296.

[65] Stefan Dziembowski and Krzysztof Pietrzak. “Leakage-Resilient Cryptography”. In: *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008*. 2008, pp. 293–302.

[66] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. “Simultaneous Hardcore Bits and Cryptography against Memory Attacks”. In: *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*. Vol. 5444. LNCS. Springer, 2009, pp. 474–495.

[67] Moni Naor and Gil Segev. “Public-Key Cryptosystems Resilient to Key Leakage”. In: *Advances in Cryptology - CRYPTO 2009*. Vol. 5677. LNCS. Springer, 2009, pp. 18–35.

[68] Yevgeniy Dodis et al. “Efficient Public-Key Cryptography in the Presence of Key Leakage”. In: *Advances in Cryptology - ASIACRYPT 2010*. 2010, pp. 613–631.

[69] Shengli Liu, Jian Weng, and Yunlei Zhao. “Efficient Public Key Cryptosystem Resilient to Key Leakage Chosen Ciphertext Attacks”. In: *Topics in Cryptology - CT-RSA 2013*. Vol. 7779. LNCS. Springer, 2013, pp. 84–100.

[70] Baodong Qin and Shengli Liu. “Leakage-Resilient Chosen-Ciphertext Secure Public-Key Encryption from Hash Proof System and One-Time Lossy Filter”. In: *Advances in Cryptology - ASIACRYPT 2013*. Vol. 8270. LNCS. Springer, 2013, pp. 381–400.

[71] Baodong Qin and Shengli Liu. “Leakage-Flexible CCA-secure Public-Key Encryption: Simple Construction and Free of Pairing”. In: *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography*. Vol. 8383. LNCS. Springer, 2014, pp. 19–36.

[72] Yu Chen, Baodong Qin, and Haiyang Xue. “Regularly Lossy Functions and Their Applications”. In: *Topics in Cryptology - CT-RSA 2018*. 2018, pp. 491–511.

[73] Yu Chen, Yuyu Wang, and Hong-Sheng Zhou. “Leakage-Resilient Cryptography from Puncturable Primitives and Obfuscation”. In: *Advances in Cryptology - ASIACRYPT 2018*. 2018, pp. 575–606.

[74] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. “Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model”. In: *Advances in Cryptology - CRYPTO 2009*. Vol. 5677. LNCS. Springer, 2009, pp. 36–54.

[75] Joël Alwen et al. “Public-Key Encryption in the Bounded-Retrieval Model”. In: *Advances in Cryptology - EUROCRYPT 2010*. Vol. 6110. LNCS. Springer, 2010, pp. 113–134.

[76] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. “On cryptography with auxiliary input”. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*. ACM, 2009, pp. 621–630.

[77] Yevgeniy Dodis et al. “Public-Key Encryption Schemes with Auxiliary Inputs”. In: *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*. Vol. 5978. LNCS. Springer, 2010, pp. 361–381.

[78] Zvika Brakerski et al. “Overcoming the Hole in the Bucket: Public-Key Cryptography Resilient to Continual Memory Leakage”. In: *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010*. IEEE Computer Society, 2010, pp. 501–510.

[79] Allison B. Lewko, Yannis Rouselakis, and Brent Waters. “Achieving Leakage Resilience through Dual System Encryption”. In: *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*. Vol. 6597. LNCS. Springer, 2011, pp. 70–88.

[80] Allison B. Lewko, Mark Lewko, and Brent Waters. “How to leak on key updates”. In: *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011*. ACM, 2011, pp. 725–734.

[81] Tsz Hon Yuen et al. “Identity-Based Encryption Resilient to Continual Auxiliary Leakage”. In: *Advances in Cryptology - EUROCRYPT 2012*. Vol. 7237. Lncs. Springer, 2012, pp. 117–134.

[82] Dana Dachman-Soled et al. “Leakage-Resilient Public-Key Encryption from Obfuscation”. In: *Public-Key Cryptography - PKC 2016*. 2016, pp. 101–128.

[83] Rosario Gennaro et al. “Algorithmic Tamper-Proof (ATP) Security: Theoretical Foundations for Security against Hardware Tampering”. In: *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*. 2004, pp. 258–277.

[84] Yuval Ishai et al. “Private Circuits II: Keeping Secrets in Tamperable Circuits”. In: *Advances in Cryptology - EUROCRYPT 2006*. Vol. 4004. Lecture Notes in Computer Science. Springer, 2006, pp. 308–327.

[85] Sebastian Faust, Krzysztof Pietrzak, and Daniele Venturi. "Tamper-Proof Circuits: How to Trade Leakage for Tamper-Resilience". In: *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011*. Vol. 6755. Lecture Notes in Computer Science. Springer, 2011, pp. 391–402.

[86] Dana Dachman-Soled and Yael Tauman Kalai. "Securing Circuits against Constant-Rate Tampering". In: *Advances in Cryptology - CRYPTO 2012*. Vol. 7417. Springer, 2012, pp. 533–551.

[87] Dana Dachman-Soled and Yael Tauman Kalai. "Securing Circuits and Protocols against 1/poly(k) Tampering Rate". In: *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014*. Vol. 8349. Lecture Notes in Computer Science. Springer, 2014, pp. 540–565.

[88] Per Austrin et al. "On the Impossibility of Cryptography with Tamperable Randomness". In: *Advances in Cryptology - CRYPTO 2014*. Vol. 8616. Lecture Notes in Computer Science. Springer, 2014, pp. 462–479.

[89] Mihir Bellare, David Cash, and Rachel Miller. "Cryptography Secure against Related-Key Attacks and Tampering". In: *Advances in Cryptology - ASIACRYPT 2011*. Vol. 7073. LNCS. Springer, 2011, pp. 486–503.

[90] Hoeteck Wee. "Public Key Encryption against Related Key Attacks". In: *Public Key Cryptography - PKC 2012*. 2012, pp. 262–279.

[91] Baodong Qin et al. "Continuous Non-malleable Key Derivation and Its Application to Related-Key Security". In: *Public-Key Cryptography - PKC 2015*. Vol. 9020. LNCS. Springer, 2015, pp. 557–578.

[92] Sebastian Faust et al. "Efficient Non-malleable Codes and Key-Derivation for Poly-size Tampering Circuits". In: *Advances in Cryptology - EUROCRYPT 2014*. Vol. 8441. LNCS. Springer, 2014, pp. 111–128.

[93] Yu Chen et al. "Non-Malleable Functions and Their Applications". In: *Public-Key Cryptography - PKC 2016*. Full version to appear at JoC 2022. 2016, pp. 386–416.

[94] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. "Non-Malleable Codes". In: *Innovations in Computer Science - ICS 2010*. Tsinghua University Press, 2010, pp. 434–452.

[95] Sebastian Faust et al. "Non-Malleable Codes for Space-Bounded Tampering". In: *Advances in Cryptology - CRYPTO 2017*. Vol. 10402. Lecture Notes in Computer Science. Springer, 2017, pp. 95–126.

[96] Gianluca Brian et al. "Continuously Non-malleable Codes Against Bounded-Depth Tampering". In: *Advances in Cryptology - ASIACRYPT 2022*. Vol. 13794. Lecture Notes in Computer Science. Springer, 2022, pp. 384–413.

[97] Yuyu Wang et al. "Impossibility on Tamper-Resilient Cryptography with Uniqueness Properties". In: *Public-Key Cryptography - PKC 2021*. Ed. by Juan A. Garay. Vol. 12710. Lecture Notes in Computer Science. Springer, 2021, pp. 389–420.

[98] Brent Waters. "Efficient Identity-Based Encryption Without Random Oracles". In: *Advances in Cryptology - EUROCRYPT 2005*. Vol. 3494. LNCS. Springer, 2005, pp. 114–127.

[99] Olivier Blazy, Eike Kiltz, and Jiaxin Pan. "(Hierarchical) Identity-Based Encryption from Affine Message Authentication". In: *Advances in Cryptology - CRYPTO 2014*. Springer, 2014, pp. 408–425.

[100] Shuai Han, Shengli Liu, and Dawu Gu. "Almost Tight Multi-user Security Under Adaptive Corruptions & Leakages in the Standard Model". In: *Advances in Cryptology - EUROCRYPT 2023*. Springer, 2023, pp. 132–162.

[101] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. "Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements". In: *Advances in Cryptology - EUROCRYPT 2000*. Vol. 1807. LNCS. Springer, 2000, pp. 259–274.

[102] Dennis Hofheinz and Tibor Jager. "Tightly Secure Signatures and Public-Key Encryption". In: *Advances in Cryptology - CRYPTO 2012*. Vol. 7417. Springer, 2012, pp. 590–607.

[103] Jens Groth and Amit Sahai. “Efficient Non-interactive Proof Systems for Bilinear Groups”. In: *Advances in Cryptology - EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. Lecture Notes in Computer Science. Springer, 2008, pp. 415–432.

[104] Masayuki Abe et al. “Tagged One-Time Signatures: Tight Security and Optimal Tag Size”. In: *Public-Key Cryptography - PKC 2013*. Vol. 7778. Springer, 2013, pp. 312–331.

[105] Benoît Libert et al. “Non-malleability from Malleability: Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures”. In: *Advances in Cryptology - EUROCRYPT 2014*. Vol. 8441. Lecture Notes in Computer Science. Springer, 2014, pp. 514–532.

[106] Benoît Libert et al. “Compactly Hiding Linear Spans - Tightly Secure Constant-Size Simulation-Sound QA-NIZK Proofs and Applications”. In: *Advances in Cryptology - ASIACRYPT 2015*. Vol. 9452. Lecture Notes in Computer Science. Springer, 2015, pp. 681–707.

[107] Romain Gay et al. “More Efficient (Almost) Tightly Secure Structure-Preserving Signatures”. In: *Advances in Cryptology - EUROCRYPT 2018*. Vol. 10821. Lecture Notes in Computer Science. Springer, 2018, pp. 230–258.

[108] Masayuki Abe et al. “Shorter QA-NIZK and SPS with Tighter Security”. In: *Advances in Cryptology - ASIACRYPT 2019*. Vol. 11923. Lecture Notes in Computer Science. Springer, 2019, pp. 669–699.

[109] Ronald Cramer and Victor Shoup. “A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack”. In: *Advances in Cryptology - CRYPTO 1998*. 1998, pp. 13–25.

[110] Romain Gay et al. “Tightly CCA-Secure Encryption Without Pairings”. In: *Advances in Cryptology - EUROCRYPT 2016*. Vol. 9665. Lecture Notes in Computer Science. Springer, 2016, pp. 1–27.

[111] Romain Gay, Dennis Hofheinz, and Lisa Kohl. “Kurosawa-Desmedt Meets Tight Security”. In: *Advances in Cryptology - CRYPTO 2017*. Vol. 10403. Lecture Notes in Computer Science. Springer, 2017, pp. 133–160.

[112] Lin Lyu et al. “Tightly SIM-SO-CCA Secure Public Key Encryption from Standard Assumptions”. In: *Public-Key Cryptography - PKC 2018*. Vol. 10769. Lecture Notes in Computer Science. Springer, 2018, pp. 62–92.

[113] Shuai Han et al. “Tight Leakage-Resilient CCA-Security from Quasi-Adaptive Hash Proof System”. In: *Advances in Cryptology - CRYPTO 2019*. Vol. 11693. Lecture Notes in Computer Science. Springer, 2019, pp. 417–447.

[114] Dennis Hofheinz, Jessica Koch, and Christoph Striecks. “Identity-Based Encryption with (Almost) Tight Security in the Multi-instance, Multi-ciphertext Setting”. In: *Public-Key Cryptography - PKC 2015*. Vol. 9020. Lecture Notes in Computer Science. Springer, 2015, pp. 799–822.

[115] Jie Chen, Junqing Gong, and Jian Weng. “Tightly Secure IBE Under Constant-Size Master Public Key”. In: *Public-Key Cryptography - PKC 2017*. Vol. 10174. Lecture Notes in Computer Science. Springer, 2017, pp. 207–231.

[116] Adi Shamir. “Identity-Based Cryptosystems and Signatures Schemes”. In: *Advances in Cryptology - CRYPTO 1984*. 1984, pp. 47–53.

[117] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. “Cryptosystems based on pairing”. In: *The 2000 Symposium on Cryptography and Information Security, Japan* 45 (2000), pp. 26–28.

[118] Dan Boneh and Matthew Franklin. “Identity-Based Encryption from the Weil Pairing”. In: *Advances in Cryptology - CRYPTO 2001*. Vol. 2139. LNCS. Springer, 2001, pp. 213–229.

[119] Clifford Cocks. “An Indentity Based Encryption Scheme Based on Quadratic Residues”. In: *Cryptography and Coding, 8th IMA International Conference*. Vol. 2260. LNCS. Springer, 2001, pp. 360–363.

[120] Craig Gentry and Alice Silverberg. “Hierarchical ID-Based Cryptography”. In: *Advances in Cryptology - ASIACRYPT 2002*. Vol. 2501. LNCS. Springer, 2002, pp. 548–566.

[121] Jeremy Horwitz and Ben Lynn. “Toward Hierarchical Identity-Based Encryption”. In: *Advances in Cryptology - EUROCRYPT 2002*. Vol. 2322. LNCS. Springer, 2002, pp. 466–481.

[122] Ran Canetti, Shai Halevi, and Jonathan Katz. “Chosen-ciphertext security from identity based encryption”. In: *Advances in Cryptology - EUROCRYPT 2004*. Vol. 3027. LNCS. Springer, 2004, pp. 207–222.

[123] Dan Boneh and Xavier Boyen. “Efficient Selective-ID Secure Identity Based Encryption without Random Oracles”. In: *Advances in Cryptology - EUROCRYPT 2004*. Vol. 3027. LNCS. Springer, 2004, pp. 223–238.

[124] Dan Boneh and Xavier Boyen. “Secure Identity Based Encryption Without Random Oracles”. In: *Advances in Cryptology - CRYPTO 2004*. Vol. 3152. LNCS. Springer, 2004, pp. 443–459.

[125] Susan Hohenberger, Amit Sahai, and Brent Waters. “Replacing a Random Oracle: Full Domain Hash from Indistinguishability Obfuscation”. In: *Advances in Cryptology - EUROCRYPT 2014*. Vol. 8441. LNCS. Springer, 2014, pp. 201–220.

[126] Craig Gentry. “Practical Identity-Based Encryption Without Random Oracles”. In: *Advances in Cryptology - EUROCRYPT 2006*. Vol. 4004. LNCS. Springer, 2006, pp. 445–464.

[127] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. “Trapdoors for hard lattices and new cryptographic constructions”. In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008*. ACM, 2008, pp. 197–206.

[128] Shweta Agrawal, Dan Boneh, and Xavier Boyen. “Efficient Lattice (H)IBE in the Standard Model”. In: *Advances in Cryptology - EUROCRYPT 2010*. Vol. 6110. LNCS. Springer, 2010, pp. 553–572.

[129] David Cash et al. “Bonsai Trees, or How to Delegate a Lattice Basis”. In: *Advances in Cryptology - EUROCRYPT 2010*. Vol. 6110. LNCS. Springer, 2010, pp. 523–552.

[130] Shota Yamada. “Adaptively Secure Identity-Based Encryption from Lattices with Asymptotically Shorter Public Parameters”. In: *Advances in Cryptology - EUROCRYPT 2016*. Vol. 9666. Lecture Notes in Computer Science. Springer, 2016, pp. 32–62.

[131] Xavier Boyen and Qinyi Li. “Towards Tightly Secure Lattice Short Signature and Id-Based Encryption”. In: *Advances in Cryptology - ASIACRYPT 2016*. Vol. 10032. Lecture Notes in Computer Science. 2016, pp. 404–434.

[132] Shuichi Katsumata and Shota Yamada. “Partitioning via Non-linear Polynomial Functions: More Compact IBEs from Ideal Lattices and Bilinear Maps”. In: *Advances in Cryptology - ASIACRYPT 2016*. Vol. 10032. Lecture Notes in Computer Science. 2016, pp. 682–712.

[133] Daniel Apon, Xiong Fan, and Feng-Hao Liu. *Compact Identity Based Encryption from LWE*. Cryptology ePrint Archive, Paper 2016/125. https://eprint.iacr.org/2016/125. 2016. URL: https://eprint.iacr.org/2016/125.

[134] Jiang Zhang, Yu Chen, and Zhenfeng Zhang. “Programmable Hash Functions from Lattices: Short Signatures and IBEs with Small Key Sizes”. In: *Advances in Cryptology - CRYPTO 2016*. 2016, pp. 303–332.

[135] Parhat Abla et al. “Ring-Based Identity Based Encryption - Asymptotically Shorter MPK and Tighter Security”. In: *Theory of Cryptography - 19th International Conference, TCC 2021*. Vol. 13044. Lecture Notes in Computer Science. Springer, 2021, pp. 157–187.

[136] Shota Yamada. “Asymptotically Compact Adaptively Secure Lattice IBEs and Verifiable Random Functions via Generalized Partitioning Techniques”. In: *Advances in Cryptology - CRYPTO 2017*. Vol. 10403. Lecture Notes in Computer Science. Springer, 2017, pp. 161–193.

[137] Dan Boneh et al. “On the Impossibility of Basing Identity Based Encryption on Trapdoor Permutations”. In: *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008*. IEEE Computer Society, 2008, pp. 283–292.

[138] Periklis A. Papakonstantinou, Charles Rackoff, and Yevgeniy Vahlis. “How powerful are the DDH hard groups?” In: *IACR Cryptol. ePrint Arch.* (2012), p. 653.

[139] Nico Döttling and Sanjam Garg. “Identity-Based Encryption from the Diffie-Hellman Assumption”. In: *Advances in Cryptology - CRYPTO 2017*. Vol. 10401. Lecture Notes in Computer Science. Springer, 2017, pp. 537–569.

[140] Nico Döttling and Sanjam Garg. “From Selective IBE to Full IBE and Selective HIBE”. In: *TCC 2017*. Vol. 10677. Lecture Notes in Computer Science. Springer, 2017, pp. 372–408.

[141] Dennis Hofheinz and Eike Kiltz. “Programmable Hash Functions and Their Applications”. In: *Advances in Cryptology - CRYPTO 2008*. 2008, pp. 21–38.

[142] Yu Chen et al. “Anonymous Identity-Based Hash Proof System and Its Applications”. In: *Provable Security - 6th International Conference, ProvSec 2012*. 2012, pp. 143–160.

[143] Dan Boneh, Craig Gentry, and Michael Hamburg. “Space-Efficient Identity Based Encryption Without Pairings”. In: *48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007*. IEEE Computer Society, 2007, pp. 647–657.

[144] Jean-Sébastien Coron. “A variant of Boneh-Franklin IBE with a tight reduction in the random oracle model”. In: *Des. Codes Cryptography* 50.1 (2009), pp. 115–133.

[145] Yu Chen et al. “Identity-Based Extractable Hash Proofs and Their Applications”. In: *International Conference on Applied Cryptography and Network Security - ACNS 2012*. 2012, pp. 153–170.

[146] Eike Kiltz and David Galindo. “Direct Chosen-Ciphertext Secure Identity-Based Key Encapsulation Without Random Oracles”. In: *Information Security and Privacy, 11th Australasian Conference, ACISP 2006*. Vol. 4058. LNCS. Springer, 2006, pp. 336–347.

[147] Eike Kiltz and Yevgeniy Vahlis. “CCA2 Secure IBE: Standard Model Efficiency through Authenticated Symmetric Encryption”. In: *CT-RSA*. Vol. 4964. LNCS. Springer, 2008, pp. 221–238.

[148] Kristiyan Haralambiev et al. “Simple and Efficient Public-Key Encryption from Computational Diffie-Hellman in the Standard Model”. In: *Public Key Cryptography - PKC 2010*. 2010, pp. 1–18.

[149] David Galindo. “Chosen-Ciphertext Secure Identity-Based Encryption from Computational Bilinear Diffie-Hellman”. In: *Pairing-Based Cryptography - Pairing 2010*. Vol. 6487. LNCS. Springer, 2010, pp. 367–376.

[150] Yu Chen, Liqun Chen, and Zongyang Zhang. “CCA Secure IB-KEM from the Computational Bilinear Diffie-Hellman Assumption in the Standard Model”. In: *Information Security and Cryptology - 14th International Conference, ICISC 2011*. 2011, pp. 275–301.

[151] Alexandra Boldyreva, Vipul Goyal, and Virendra Kumar. “Identity-based encryption with efficient revocation”. In: *Proceedings of the 2008 ACM Conference on Computer and Communications Security, CCS 2008*. ACM, 2008, pp. 417–426.

[152] Jae Hong Seo and Keita Emura. “Revocable Identity-Based Encryption Revisited: Security Model and Construction”. In: *Public-Key Cryptography - PKC 2013*. Vol. 7778. Lecture Notes in Computer Science. Springer, 2013, pp. 216–234.

[153] Jae Hong Seo and Keita Emura. “Revocable hierarchical identity-based encryption via history-free approach”. In: *Theor. Comput. Sci.* 615 (2016), pp. 45–60.

[154] Kwangsu Lee, Dong Hoon Lee, and Jong Hwan Park. “Efficient revocable identity-based encryption via subset difference methods”. In: *Des. Codes Cryptogr.* 85.1 (2017), pp. 39–76.

[155] Shuichi Katsumata, Takahiro Matsuda, and Atsushi Takayasu. “Lattice-Based Revocable (Hierarchical) IBE with Decryption Key Exposure Resistance”. In: *Public-Key Cryptography - PKC 2019*. Vol. 11443. Lecture Notes in Computer Science. Springer, 2019, pp. 441–471.

[156] Fuchun Guo, Yi Mu, and Zhide Chen. “Identity-Based Online/Offline Encryption”. In: *Financial Cryptography and Data Security, 12th International Conference, FC 2008*. Vol. 5143. Lecture Notes in Computer Science. Springer, 2008, pp. 247–261.

[157] Shweta Agrawal et al. “Functional Encryption for Threshold Functions (or Fuzzy IBE) from Lattices”. In: *Public Key Cryptography - PKC 2012*. Vol. 7293. LNCS. Springer, 2012, pp. 280–297.

[158] Jie Chen et al. “Identity-Based Matchmaking Encryption from Standard Assumptions”. In: *Advances in Cryptology - ASIACRYPT 2022*. Vol. 13793. Lecture Notes in Computer Science. Springer, 2022, pp. 394–422.

[159] Amit Sahai and Brent Waters. “Fuzzy Identity Based Encryption”. In: *Advances in Cryptology - EUROCRYPT 2005*. Vol. 3494. LNCS. Springer, 2005, pp. 457–473.

[160] Vipul Goyal et al. “Attribute-based encryption for fine-grained access control of encrypted data”. In: *ACM Conference on Computer and Communications Security, CCS 2006*. ACM, 2006, pp. 89–98.

[161] John Bethencourt, Amit Sahai, and Brent Waters. “Ciphertext-Policy Attribute-Based Encryption”. In: *IEEE Symposium on Security and Privacy 2007 (SP’ 2007)*. IEEE Computer Society, 2007, pp. 321–334.

[162] Ling Cheung and Calvin C. Newport. “Provably secure ciphertext policy ABE”. In: *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*. ACM, 2007, pp. 456–465.

[163] Brent Waters. “Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization”. In: *Public Key Cryptography - PKC 2011*. Vol. 6571. Lecture Notes in Computer Science. Springer, 2011, pp. 53–70.

[164] Rafail Ostrovsky, Amit Sahai, and Brent Waters. “Attribute-based encryption with non-monotonic access structures”. In: *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*. ACM, 2007, pp. 195–203.

[165] Tatsuaki Okamoto and Katsuyuki Takashima. “Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption”. In: *Advances in Cryptology - CRYPTO 2010*. Vol. 6223. Springer, 2010, pp. 191–208.

[166] Allison B. Lewko et al. “Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption”. In: *Advances in Cryptology - EUROCRYPT 2010*. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 62–91.

[167] Jiang Zhang, Zhenfeng Zhang, and Aijun Ge. “Ciphertext policy attribute-based encryption from lattices”. In: *7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS 2012*. ACM, 2012, pp. 16–17.

[168] Xavier Boyen. “Attribute-Based Functional Encryption on Lattices”. In: *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013*. Vol. 7785. Lecture Notes in Computer Science. Springer, 2013, pp. 122–142.

[169] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. “Attribute-based encryption for circuits”. In: *Symposium on Theory of Computing Conference, STOC’13*. ACM, 2013, pp. 545–554.

[170] Dan Boneh et al. “Fully Key-Homomorphic Encryption, Arithmetic Circuit ABE and Compact Garbled Circuits”. In: *Advances in Cryptology - EUROCRYPT 2014*. Vol. 8441. Lecture Notes in Computer Science. Springer, 2014, pp. 533–556.

[171] Craig Gentry, Amit Sahai, and Brent Waters. “Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based”. In: *Advances in Cryptology - CRYPTO 2013*. Vol. 8042. LNCS. Springer, 2013, pp. 75–92.

[172] Pratish Datta, Ilan Komargodski, and Brent Waters. “Decentralized Multi-authority ABE for DNFs from LWE”. In: *Advances in Cryptology - EUROCRYPT 2021*. Vol. 12696. Lecture Notes in Computer Science. Springer, 2021, pp. 177–209.

[173] Jonathan Katz, Amit Sahai, and Brent Waters. “Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products”. In: *Advances in Cryptology - EUROCRYPT 2008*. Vol. 4965. LNCS. Springer, 2008, pp. 146–162.

[174] Tatsuaki Okamoto and Katsuyuki Takashima. “Fully Secure Unbounded Inner-Product and Attribute-Based Encryption”. In: *Advances in Cryptology - ASIACRYPT 2012*. Vol. 7658. Lecture Notes in Computer Science. Springer, 2012, pp. 349–366.

[175] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. “Predicate Encryption for Circuits from LWE”. In: *Advances in Cryptology - CRYPTO 2015*. Vol. 9216. Lecture Notes in Computer Science. Springer, 2015, pp. 503–523.

[176] Nir Bitansky and Vinod Vaikuntanathan. “Indistinguishability Obfuscation from Functional Encryption”. In: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015*. IEEE Computer Society, 2015, pp. 171–190.

[177] Hoeteck Wee. “Attribute-Hiding Predicate Encryption in Bilinear Groups, Revisited”. In: *Theory of Cryptography - 15th International Conference, TCC 2017*. Vol. 10677. Lecture Notes in Computer Science. Springer, 2017, pp. 206–233.

[178] Pratish Datta, Tatsuaki Okamoto, and Katsuyuki Takashima. “Adaptively Simulation-Secure Attribute-Hiding Predicate Encryption”. In: *Advances in Cryptology - ASIACRYPT 2018*. Vol. 11273. Lecture Notes in Computer Science. Springer, 2018, pp. 640–672.

[179] Brent Waters, Hoeteck Wee, and David J. Wu. “Multi-authority ABE from Lattices Without Random Oracles”. In: *Theory of Cryptography - 20th International Conference, TCC 2022*. Vol. 13747. Lecture Notes in Computer Science. Springer, 2022, pp. 651–679.

[180] Pratish Datta, Ilan Komargodski, and Brent Waters. “Fully Adaptive Decentralized Multi-Authority ABE”. In: *Advances in Cryptology - EUROCRYPT 2023*. Vol. 14006. Lecture Notes in Computer Science. Springer, 2023, pp. 447–478.

[181] Susan Hohenberger et al. “Registered Attribute-Based Encryption”. In: *Advances in Cryptology - EUROCRYPT 2023*. Vol. 14006. Lecture Notes in Computer Science. Springer, 2023, pp. 511–542.

[182] Cody Freitag, Brent Waters, and David J. Wu. “How to Use (Plain) Witness Encryption: Registered ABE, Flexible Broadcast, and More”. In: *Advances in Cryptology - CRYPTO 2023*. Vol. 14084. Lecture Notes in Computer Science. Springer, 2023, pp. 498–531.

[183] Adam O’Neill. *Definitional Issues in Functional Encryption*. IACR Cryptology ePrint Archive, Report 2010/556. http://eprint.iacr.org/2010/556. 2010.

[184] Dan Boneh, Amit Sahai, and Brent Waters. “Functional Encryption: Definitions and Challenges”. In: *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*. Vol. 6597. LNCS. Springer, 2011, pp. 253–273.

[185] Dan Boneh et al. “Public Key Encryption with Keyword Search”. In: *Advances in Cryptology - EUROCRYPT 2004*. Vol. 3621. LNCS. Springer, 2004, pp. 506–522.

[186] Allison B. Lewko et al. “Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption”. In: *Advances in Cryptology - EUROCRYPT 2010*. Vol. 6110. LNCS. Springer, 2010, pp. 62–91.

[187] Dan Boneh and Brent Waters. “Conjunctive, Subset, and Range Queries on Encrypted Data”. In: *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007*. Vol. 4392. LNCS. Springer, 2007, pp. 535–554.

[188] Amit Sahai and Hakan Seyalioglu. “Worry-free encryption: functional encryption with public keys”. In: *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010*. ACM, 2010, pp. 463–472.

[189] Sanjam Garg et al. “Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits”. In: *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*. IEEE Computer Society, 2013, pp. 40–49.

[190] Elette Boyle, Kai-Min Chung, and Rafael Pass. “On Extractability Obfuscation”. In: *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014*. Vol. 8349. LNCS. Springer, 2014, pp. 52–73.

[191] Brent Waters. “A Punctured Programming Approach to Adaptively Secure Functional Encryption”. In: *Advances in Cryptology - CRYPTO 2015*. Vol. 9216. Lecture Notes in Computer Science. Springer, 2015, pp. 678–697.

[192] Sanjam Garg et al. “Functional Encryption Without Obfuscation”. In: *Theory of Cryptography - 13th International Conference, TCC 2016-A*. Vol. 9563. Lecture Notes in Computer Science. Springer, 2016, pp. 480–511.

[193] Michel Abdalla et al. “Simple Functional Encryption Schemes for Inner Products”. In: *Public-Key Cryptography - PKC 2015*. Vol. 9020. Lecture Notes in Computer Science. Springer, 2015, pp. 733–751.

[194] Shweta Agrawal, Benoît Libert, and Damien Stehlé. “Fully Secure Functional Encryption for Inner Products, from Standard Assumptions”. In: *Advances in Cryptology - CRYPTO 2016*. Vol. 9816. Lecture Notes in Computer Science. Springer, 2016, pp. 333–362.

[195] Shweta Agrawal et al. “Adaptive Simulation Security for Inner Product Functional Encryption”. In: *Public-Key Cryptography - PKC 2020*. Vol. 12110. Lecture Notes in Computer Science. Springer, 2020, pp. 34–64.

[196] Carmen Elisabetta Zaira Baltico et al. “Practical Functional Encryption for Quadratic Functions with Applications to Predicate Encryption”. In: *Advances in Cryptology - CRYPTO 2017*. Vol. 10401. Lecture Notes in Computer Science. Springer, 2017, pp. 67–98.

[197] Romain Gay. “A New Paradigm for Public-Key Functional Encryption for Degree-2 Polynomials”. In: *Public-Key Cryptography - PKC 2020*. Vol. 12110. Lecture Notes in Computer Science. Springer, 2020, pp. 95–120.

[198] Junqing Gong and Haifeng Qian. “Simple and efficient FE for quadratic functions”. In: *Des. Codes Cryptogr.* 89.8 (2021), pp. 1757–1786.

[199] Hoeteck Wee. “Functional Encryption for Quadratic Functions from k-Lin, Revisited”. In: *Theory of Cryptography - 18th International Conference, TCC 2020*. Vol. 12550. Lecture Notes in Computer Science. Springer, 2020, pp. 210–228.

[200] Shafi Goldwasser et al. “Multi-input Functional Encryption”. In: *Advances in Cryptology - EUROCRYPT 2014*. Vol. 8441. Lecture Notes in Computer Science. Springer, 2014, pp. 578–602.

[201] Michel Abdalla et al. “Multi-input Inner-Product Functional Encryption from Pairings”. In: *Advances in Cryptology - EUROCRYPT 2017*. Vol. 10210. Lecture Notes in Computer Science. 2017, pp. 601–626.

[202] Michel Abdalla et al. “Multi-Input Functional Encryption for Inner Products: Function-Hiding Realizations and Constructions Without Pairings”. In: *Advances in Cryptology - CRYPTO 2018*. Vol. 10991. Lecture Notes in Computer Science. Springer, 2018, pp. 597–627.

[203] Junichi Tomida. “Tightly Secure Inner Product Functional Encryption: Multi-input and Function-Hiding Constructions”. In: *Advances in Cryptology - ASIACRYPT 2019*. Vol. 11923. Lecture Notes in Computer Science. Springer, 2019, pp. 459–488.

[204] Shweta Agrawal, Rishab Goyal, and Junichi Tomida. “Multi-input Quadratic Functional Encryption from Pairings”. In: *Advances in Cryptology - CRYPTO 2021*. Vol. 12828. Lecture Notes in Computer Science. Springer, 2021, pp. 208–238.

[205] Shweta Agrawal, Rishab Goyal, and Junichi Tomida. “Multi-Input Quadratic Functional Encryption: Stronger Security, Broader Functionality”. In: *Theory of Cryptography - 20th International Conference, TCC 2022*. Vol. 13747. Lecture Notes in Computer Science. Springer, 2022, pp. 711–740.

[206] Pratish Datta, Tatsuaki Okamoto, and Junichi Tomida. “Full-Hiding (Unbounded) Multi-input Inner Product Functional Encryption from the k-Linear Assumption”. In: *Public-Key Cryptography - PKC 2018*. Vol. 10770. Lecture Notes in Computer Science. Springer, 2018, pp. 245–277.

[207] Junichi Tomida and Katsuyuki Takashima. “Unbounded Inner Product Functional Encryption from Bilinear Maps”. In: *Advances in Cryptology - ASIACRYPT 2018*. Vol. 11273. Lecture Notes in Computer Science. Springer, 2018, pp. 609–639.

[208] Junichi Tomida. “Unbounded Quadratic Functional Encryption and More from Pairings”. In: *Advances in Cryptology - EUROCRYPT 2023*. Vol. 14006. Lecture Notes in Computer Science. Springer, 2023, pp. 543–572.

[209] Miguel Ambrona, Dario Fiore, and Claudio Soriente. “Controlled Functional Encryption Revisited: Multi-Authority Extensions and Efficient Schemes for Quadratic Functions”. In: *Proc. Priv. Enhancing Technol.* 2021.1 (2021), pp. 21–42.

[210] Jérémy Chotard et al. “Decentralized Multi-Client Functional Encryption for Inner Product”. In: *Advances in Cryptology - ASIACRYPT 2018*. Vol. 11273. Lecture Notes in Computer Science. Springer, 2018, pp. 703–732.

[211] Jérémy Chotard et al. “Dynamic Decentralized Functional Encryption”. In: *Advances in Cryptology - CRYPTO 2020*. Vol. 12170. Lecture Notes in Computer Science. Springer, 2020, pp. 747–775.

[212] Zvika Brakerski et al. “Hierarchical Functional Encryption”. In: *8th Innovations in Theoretical Computer Science Conference, ITCS 2017*. Vol. 67. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 8:1–8:27.

[213] Dawn Xiaodong Song, David Wagner, and Adrian Perrig. “Practical Techniques for Searches on Encrypted Data”. In: *2000 IEEE Symposium on Security and Privacy*. 2000, pp. 44–55.

[214] Michel Abdalla et al. “Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions”. In: *Advances in Cryptology - CRYPTO 2005*. Vol. 3621. LNCS. Springer, 2005, pp. 205–222.

[215] Philippe Golle, Jessica Staddon, and Brent R. Waters. “Secure Conjunctive Keyword Search over Encrypted Data”. In: *Applied Cryptography and Network Security, Second International Conference, ACNS 2004*. Vol. 3089. Lecture Notes in Computer Science. Springer, 2004, pp. 31–45.

[216] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. “On the Integration of Public Key Data Encryption and Public Key Encryption with Keyword Search”. In: *Information Security, 9th International Conference, ISC 2006*. Vol. 4176. LNCS. Springer, 2006, pp. 217–232.

[217] Rui Zhang and Hideki Imai. “Generic Combination of Public Key Encryption with Keyword Search and Public Key Encryption”. In: *Cryptology and Network Security, 6th International Conference, CANS 2007*. Vol. 4856. LNCS. Springer, 2007, pp. 159–174.

[218] Michel Abdalla, Mihir Bellare, and Gregory Neven. “Robust Encryption”. In: *TCC 2010*. Vol. 5978. LNCS. Springer, 2010, pp. 480–497.

[219] Yu Chen et al. “Generic constructions of integrated PKE and PEKS”. In: *Des. Codes Cryptography* 78.2 (2016), pp. 493–526.

[220] Qiuxiang Dong et al. “Fuzzy Keyword Search over Encrypted Data in the Public Key Setting”. In: *Web-Age Information Management - 14th International Conference, WAIM 2013*. Vol. 7923. Lecture Notes in Computer Science. Springer, 2013, pp. 729–740.

[221] Bing Wang et al. “Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud”. In: *2014 IEEE Conference on Computer Communications, INFOCOM 2014*. IEEE, 2014, pp. 2112–2120.

[222] Guofu Yu, Xinrui Ge, and Jia Yu. “支持数据去重的可验证模糊多关键词搜索方案”. In: 密码学报 5 (2019), p. 12.

[223] Jiaxun Hua et al. “An enhanced wildcard-based fuzzy searching scheme in encrypted databases”. In: *World Wide Web* 23.3 (2020), pp. 2185–2214.

[224] Yong Ho Hwang and Pil Joong Lee. “Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-user System”. In: *Pairing-Based Cryptography - Pairing 2007*. Vol. 4575. LNCS. Springer, 2007, pp. 2–22.

[225] Rongmao Chen et al. “Dual-Server Public-Key Encryption With Keyword Search for Secure Cloud Storage”. In: *IEEE Trans. Inf. Forensics Secur.* 11.4 (2016), pp. 789–798.

[226] Mihir Bellare, Alexandra Boldyreva, and Adam O’Neill. “Deterministic and Efficiently Searchable Encryption”. In: *Advances in Cryptology - CRYPTO 2007*. Vol. 4622. LNCS. Springer, 2007, pp. 535–552.

[227] Alexandra Boldyreva, Serge Fehr, and Adam O’Neill. “On Notions of Security for Deterministic Encryption, and Efficient Constructions without Random Oracles”. In: *Advances in Cryptology - CRYPTO 2008*. Vol. 5157. LNCS. Springer, 2008, pp. 335–359.

[228] Mihir Bellare et al. “Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles”. In: *Advances in Cryptology - CRYPTO 2008*. Vol. 5157. LNCS. Springer, 2008, pp. 360–378.

[229] Zvika Brakerski and Gil Segev. “Better Security for Deterministic Public-Key Encryption: The Auxiliary-Input Setting”. In: *Advances in Cryptology - CRYPTO 2011*. Vol. 6841. LNCS. Springer, 2011, pp. 543–560.

[230] Ilya Mironov et al. “Incremental Deterministic Public-Key Encryption”. In: *Advances in Cryptology - EUROCRYPT 2012*. Vol. 7237. LNCS. Springer, 2012, pp. 628–644.

[231] Ananth Raghunathan, Gil Segev, and Salil P. Vadhan. “Deterministic Public-Key Encryption for Adaptively Chosen Plaintext Distributions”. In: *Advances in Cryptology - EUROCRYPT 2013*. Vol. 7881. Springer, 2013, pp. 93–110.

[232] Mihir Bellare, Rafael Dowsley, and Sriram Keelveedhi. “How Secure is Deterministic Encryption?” In: *Public-Key Cryptography - PKC 2015*. Vol. 9020. Lecture Notes in Computer Science. Springer, 2015, pp. 52–73.

[233] Mark Zhandry. “On ELFs, Deterministic Encryption, and Correlated-Input Security”. In: *Advances in Cryptology - EUROCRYPT 2019*. Vol. 11478. Springer, 2019, pp. 3–32.

[234] Mihir Bellare, Wei Dai, and Lucy Li. “The Local Forking Lemma and Its Application to Deterministic Encryption”. In: *Advances in Cryptology - ASIACRYPT 2019*. Vol. 11923. Lecture Notes in Computer Science. Springer, 2019, pp. 607–636.

[235] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. “On data banks and privacy homomorphisms”. In: *Foundations of secure computation* 4.11 (1978), pp. 169–180.

[236] Shafi Goldwasser and Silvio Micali. “Probabilistic Encryption”. In: *J. Comput. Syst. Sci.* 28.2 (1984), pp. 270–299.

[237] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. “Evaluating 2-DNF Formulas on Ciphertexts”. In: *TCC 2005*. Vol. 3378. Lecture Notes in Computer Science. Springer, 2005, pp. 325–341.

[238] Tomas Sander, Adam L. Young, and Moti Yung. “Non-Interactive CryptoComputing For NC^1”. In: *FOCS 1999*. IEEE Computer Society, 1999, pp. 554–567.

[239] Craig Gentry. “Fully homomorphic encryption using ideal lattices”. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*. ACM, 2009, pp. 169–178.

[240] Craig Gentry and Shai Halevi. “Implementing Gentry’s Fully-Homomorphic Encryption Scheme”. In: *Advances in Cryptology - EUROCRYPT 2011*. Vol. 6632. Lecture Notes in Computer Science. Springer, 2011, pp. 129–148.

[241] Marten van Dijk et al. “Fully Homomorphic Encryption over the Integers”. In: *Advances in Cryptology - EUROCRYPT 2010*. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 24–43.

[242] Zvika Brakerski and Vinod Vaikuntanathan. “Efficient Fully Homomorphic Encryption from (Standard) LWE”. In: *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS*. IEEE Computer Society, 2011, pp. 97–106.

[243] Zvika Brakerski and Vinod Vaikuntanathan. “Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages”. In: *Advances in Cryptology - CRYPTO 2011*. Vol. 6841. LNCS. Springer, 2011, pp. 505–524.

[244] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. “(Leveled) fully homomorphic encryption without bootstrapping”. In: *Innovations in Theoretical Computer Science 2012*. ACM, 2012, pp. 309–325.

[245] Nigel P. Smart and Frederik Vercauteren. “Fully homomorphic SIMD operations”. In: *Des. Codes Cryptogr.* 71.1 (2014), pp. 57–81.

[246] Craig Gentry, Shai Halevi, and Nigel P. Smart. “Fully Homomorphic Encryption with Polylog Overhead”. In: *Advances in Cryptology - EUROCRYPT 2012*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 465–482.

[247] Hao Chen and Kyoohyung Han. “Homomorphic Lower Digits Removal and Improved FHE Bootstrapping”. In: *Advances in Cryptology - EUROCRYPT 2018*. Vol. 10820. Lecture Notes in Computer Science. Springer, 2018, pp. 315–337.

[248] Shai Halevi and Victor Shoup. “Faster Homomorphic Linear Transformations in HElib”. In: *Advances in Cryptology - CRYPTO 2018*. Vol. 10991. Lecture Notes in Computer Science. Springer, 2018, pp. 93–120.

[249] Zvika Brakerski and Vinod Vaikuntanathan. “Lattice-based FHE as secure as PKE”. In: *Innovations in Theoretical Computer Science, ITCS’14*. ACM, 2014, pp. 1–12.

[250] Jacob Alperin-Sheriff and Chris Peikert. “Faster Bootstrapping with Polynomial Error”. In: *Advances in Cryptology - CRYPTO 2014*. Vol. 8616. Lecture Notes in Computer Science. Springer, 2014, pp. 297–314.

[251] Ilaria Chillotti et al. “TFHE: Fast Fully Homomorphic Encryption Over the Torus”. In: *J. Cryptol.* 33.1 (2020), pp. 34–91.

[252] Nicolas Gama et al. “Structural Lattice Reduction: Generalized Worst-Case to Average-Case Reductions and Homomorphic Cryptosystems”. In: *Advances in Cryptology - EUROCRYPT 2016*. Vol. 9666. Springer, 2016, pp. 528–558.

[253] Jung Hee Cheon et al. “Homomorphic Encryption for Arithmetic of Approximate Numbers”. In: *Advances in Cryptology - ASIACRYPT 2017*. Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 409–437.

[254] Jung Hee Cheon et al. “Bootstrapping for Approximate Homomorphic Encryption”. In: *Advances in Cryptology - EUROCRYPT 2018*. Vol. 10820. Lecture Notes in Computer Science. Springer, 2018, pp. 360–384.

[255] Jung Hee Cheon et al. “A Full RNS Variant of Approximate Homomorphic Encryption”. In: *Selected Areas in Cryptography - SAC 2018*. Vol. 11349. Lecture Notes in Computer Science. Springer, 2018, pp. 347–368.

[256] Fabian Boemer et al. “nGraph-HE2: A High-Throughput Framework for Neural Network Inference on Encrypted Data”. In: *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography, WAHC 2019*. ACM, 2019, pp. 45–56.

[257] Baiyu Li and Daniele Micciancio. “On the Security of Homomorphic Encryption on Approximate Numbers”. In: *Advances in Cryptology - EUROCRYPT 2021*. Vol. 12696. Lecture Notes in Computer Science. Springer, 2021, pp. 648–677.

[258] Jung Hee Cheon, Seungwan Hong, and Duhyeong Kim. “Remark on the Security of CKKS Scheme in Practice”. In: *IACR Cryptol. ePrint Arch.* (2020), p. 1581. URL: https://eprint.iacr.org/2020/1581.

[259] Wonkyung Jung et al. “Over 100x Faster Bootstrapping in Fully Homomorphic Encryption through Memory-centric Optimization with GPUs”. In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2021.4 (2021), pp. 114–148.

[260] Yi Deng. “Magic Adversaries Versus Individual Reduction: Science Wins Either Way”. In: *Advances in Cryptology - EUROCRYPT 2017*. Vol. 10211. LNCS. 2017, pp. 351–377.

[261] Victor Shoup. *Sequences of games: a tool for taming complexity in security proofs*. IACR Cryptology ePrint Archive. http://eprint.iacr.org/2004/332. 2004.

[262] Joseph A. Akinyele, Christina Garman, and Susan Hohenberger. “Automating Fast and Secure Translations from Type-I to Type-III Pairing Schemes”. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015*. ACM, 2015, pp. 1370–1381.

[263] Peter W. Shor. “Algorithms for Quantum Computation: Discrete Logarithms and Factoring”. In: *35th Annual Symposium on Foundations of Computer Science, FOCS 1994*. 1994, pp. 124–134.

[264] Miklós Ajtai. “Generating Hard Instances of Lattice Problems (Extended Abstract)”. In: *STOC 1996*. ACM, 1996, pp. 99–108.

[265] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, STOC 2005*. ACM, 2005, pp. 84–93.

[266] Yevgeniy Dodis et al. “Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data”. In: *SIAM J. Comput.* 38.1 (2008), pp. 97–139.

[267] Dan Boneh and Matthew K. Franklin. “Identity-Based Encryption from the Weil Pairing”. In: *SIAM Journal on Computation* 32 (2003), pp. 586–615.

[268] Antoine Joux. “A One Round Protocol for Tripartite Diffie-Hellman”. In: *J. Cryptology* 17.4 (2004), pp. 263–276.

[269] Dan Boneh and Alice Silverberg. “Applications of Multilinear Forms to Cryptography”. In: (2002). http://eprint.iacr.org/2002/080.

[270] Dan Boneh and Mark Zhandry. “Multiparty Key Exchange, Efficient Traitor Tracing, and More from Indistinguishability Obfuscation”. In: *Advances in Cryptology - CRYPTO 2014*. 2014, pp. 480–499.

[271] Navid Alamati et al. “Minicrypt Primitives with Algebraic Structure and Applications”. In: *Advances in Cryptology - EUROCRYPT 2019*. Vol. 11477. Lecture Notes in Computer Science. Springer, 2019, pp. 55–82.

[272] David Cash, Eike Kiltz, and Victor Shoup. “The Twin Diffie-Hellman Problem and Applications”. In: *Advances in Cryptology - EUROCRYPT 2008*. Vol. 4965. LNCS. Springer, 2008, pp. 127–145.

[273] Eduarda S. V. Freire et al. “Non-Interactive Key Exchange”. In: *16th International Conference on Practice and Theory in Public-Key Cryptography - PKC 2013*. Vol. 7778. LNCS. Springer, 2013, pp. 254–271.

[274] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. “How to construct random functions”. In: *J. ACM* 33.4 (1986), pp. 792–807.

[275] Dan Boneh and Brent Waters. “Constrained Pseudorandom Functions and Their Applications”. In: *Advances in Cryptology - ASIACRYPT 2013*. Vol. 8270. LNCS. Springer, 2013, pp. 280–300.

[276] Aggelos Kiayias et al. “Delegatable pseudorandom functions and applications”. In: *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013*. ACM, 2013, pp. 669–684.

[277] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. “Functional Signatures and Pseudorandom Functions”. In: *17th International Conference on Practice and Theory in Public-Key Cryptography, PKC 2014*. Vol. 8383. LNCS. Springer, 2014, pp. 501–519.

[278] Yevgeniy Dodis and Matthias Ruhl. *GM-Security and Semantic Security Revisited*. http://people.csail.mit.edu/ruhl/papers/drafts/semantic.html. 1999.

[279] .

[280] Josh Benaloh. In: 1994, pp. 120–128.

[281] Pascal Paillier. “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes”. In: *Advances in Cryptology - EUROCRYPT 1999*. 1999, pp. 223–238.

[282] Yuval Ishai and Anat Paskin. “Evaluating Branching Programs on Encrypted Data”. In: *TCC 2007*. Vol. 4392. Lecture Notes in Computer Science. Springer, 2007, pp. 575–594.

[283] R. Rivest, L. Adleman, and M. Dertouzos. “On Data Banks and Privacy Homomorphisms”. In: *Foundations of Secure Computation* (1978), pp. 169–179.

[284] Shai Halevi. “Homomorphic Encryption”. In: *Tutorials on the Foundations of Cryptography*. Springer International Publishing, 2017, pp. 219–276.

[285] Ronald Cramer and Victor Shoup. “Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack”. In: *SIAM Journal on Computing* 33 (2003), pp. 167–226.

[286] Kaoru Kurosawa and Yvo Desmedt. “A New Paradigm of Hybrid Encryption Scheme”. In: *Advances in Cryptology - CRYPTO 2004*. 2004, pp. 426–442.

[287] Michael Rabin. “Digitalized Signatures and Public-Key Functions as Intractable as Factorization”. In: *MIT Laboratory for Computer Science, Technical Report TR-212* (1979).

[288] Benedikt Bünz et al. “Bulletproofs: Short Proofs for Confidential Transactions and More”. In: *2018 IEEE Symposium on Security and Privacy, SP 2018*. 2018, pp. 315–334.

[289] Prastudy Fauzi et al. “Quisquis: A New Design for Anonymous Cryptocurrencies”. In: *Advances in Cryptology - ASIACRYPT 2019*. Vol. 11921. Lecture Notes in Computer Science. Springer, 2019, pp. 649–678.

[290] Benedikt Bünz et al. “Zether: Towards Privacy in a Smart Contract World”. In: *Financial Cryptography and Data Security - FC 2020*. Vol. 12059. Springer, 2020, pp. 423–443.

[291] Yu Chen et al. “PGC: Pretty Good Confidential Transaction System with Auditability”. In: *The 25th European Symposium on Research in Computer Security, ESORICS 2020*. https://eprint.iacr.org/2019/319. 2020, pp. 591–610.

[292] Daniele Micciancio. “Duality in Lattice Cryptography (invited talk)”. In: *Public Key Cryptography - PKC 2010*. Vol. 6056. Lecture Notes in Computer Science. Springer, 2010.

[293] Ilan Komargodski. “Leakage Resilient One-Way Functions: The Auxiliary-Input Setting”. In: *Theory of Cryptography - 14th International Conference, TCC 2016-B*. Vol. 9985. LNCS. Springer, 2016, pp. 139–158.

[294] Mark Zhandry. “The Magic of ELFs”. In: *Advances in Cryptology - CRYPTO 2016*. Vol. 9814. LNCS. Springer, 2016, pp. 479–508.

[295] URL: https://zhuanlan.zhihu.com/p/109773214.

[296] Charles Rackoff and Daniel R. Simon. "Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack". In: *Advances in Cryptology - CRYPTO 1991*. Vol. 576. LNCS. 1991, pp. 433–444.

[297] Eike Kiltz. "Chosen-Ciphertext Secure Key-Encapsulation Based on Gap Hashed Diffie-Hellman". In: *Public Key Cryptography - PKC 2007*. Vol. 4450. LNCS. Full version is avaiable at ePrint Archive: Report 2007/036. Springer, 2007, pp. 282–297.

[298] Dennis Hofheinz and Eike Kiltz. "Practical Chosen Ciphertext Secure Encryption from Factoring". In: *Advances in Cryptology - EUROCRYPT 2009*. Vol. 5479. LNCS. Springer, 2009, pp. 313–332.

[299] Boaz Barak et al. "On the (Im)possibility of Obfuscating Programs". In: *Advances in Cryptology - CRYPTO 2001*. Vol. 2139. LNCS. Springer, 2001, pp. 1–18.

[300] Mihir Bellare, Igors Stepanovs, and Brent Waters. "New Negative Results on Differing-Inputs Obfuscation". In: *Advances in Cryptology - EUROCRYPT 2016*. Vol. 9666. LNCS. Springer, 2016, pp. 792–821.

[301] Yu Chen and Zongyang Zhang. "Publicly Evaluable Pseudorandom Functions and Their Applications". In: *9th International Conference on Security and Cryptography for Networks, SCN 2014*. 2014, pp. 115–134.

[302] Moni Naor and Omer Reingold. "Number-theoretic constructions of efficient pseudo-random functions". In: *J. ACM* 51.2 (2004), pp. 231–262.

[303] Paul C. Kocher. "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems". In: *Advances in Cryptology - CRYPTO 1996*. 1996, pp. 104–113.

[304] Karine Gandolfi, Christophe Mourtel, and Francis Olivier. "Electromagnetic Analysis: Concrete Results". In: *CHES 2001*. Generators. 2001, pp. 251–261.

[305] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. "Differential Power Analysis". In: *Advances in Cryptology - CRYPTO 1999*. 1999, pp. 388–397.

[306] J. Alex Halderman et al. "Lest We Remember: Cold Boot Attacks on Encryption Keys". In: *Proceedings of the 17th USENIX Security Symposium*. 2008, pp. 45–60.

[307] Henri Cohen et al., eds. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman and Hall/CRC, 2005. ISBN: 978-1-58488-518-4. DOI: 10.1201/9781420034981. URL: https://doi.org/10.1201/9781420034981.

[308] Giovanni Di Crescenzo, Richard J. Lipton, and Shabsi Walfish. "Perfectly Secure Password Protocols in the Bounded Retrieval Model". In: *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*. Ed. by Shai Halevi and Tal Rabin. Vol. 3876. Lecture Notes in Computer Science. Springer, 2006, pp. 225–244.

[309] Stefan Dziembowski. "Intrusion-Resilience Via the Bounded-Storage Model". In: *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006*. Vol. 3876. LNCS. Springer, 2006, pp. 207–224.

[310] Shai Halevi and Huijia Lin. "After-the-Fact Leakage in Public-Key Encryption". In: *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*. Ed. by Yuval Ishai. Vol. 6597. Lecture Notes in Computer Science. Springer, 2011, pp. 107–124.

[311] Shengli Liu, Jian Weng, and Yunlei Zhao. "Efficient Public Key Cryptosystem Resilient to Key Leakage Chosen Ciphertext Attacks". In: *Topics in Cryptology - CT-RSA 2013*. Vol. 7779. LNCS. Springer, 2013, pp. 84–100.

[312] Baodong Qin, Shengli Liu, and Kefei Chen. "Efficient chosen-ciphertext secure public-key encryption scheme with high leakage-resilience". In: *IET Inf. Secur.* 9.1 (2015), pp. 32–42.

[313] Yu Chen, Baodong Qin, and Haiyang Xue. “Regular lossy functions and their applications in leakage-resilient cryptography”. In: *Theor. Comput. Sci.* 739 (2018), pp. 13–38.

[314] Dennis Hofheinz. “Circular Chosen-Ciphertext Security with Compact Ciphertexts”. In: *Advances in Cryptology - EUROCRYPT 2013*. Vol. 7881. LNCS. Springer, 2013, pp. 520–536.

[315] Eike Kiltz, Krzysztof Pietrzak, and Mario Szegedy. “Digital Signatures with Minimal Overhead from Indifferentiable Random Invertible Functions”. In: *Advances in Cryptology - CRYPTO 2013*. Vol. 8042. Lecture Notes in Computer Science. Springer, 2013, pp. 571–588.

[316] Yannick Seurin. “On the Lossiness of the Rabin Trapdoor Function”. In: *Public-Key Cryptography - PKC 2014*. Vol. 8383. Lecture Notes in Computer Science. Springer, 2014, pp. 380–398.

[317] Brett Hemenway and Rafail Ostrovsky. “Extended-DDH and Lossy Trapdoor Functions”. In: *Public Key Cryptography - PKC 2012*. Vol. 7293. LNCS. Springer, 2012, pp. 627–643.

[318] Hoeteck Wee. “Dual Projective Hashing and Its Applications - Lossy Trapdoor Functions and More”. In: *Advances in Cryptology - EUROCRYPT 2012*. Vol. 7237. LNCS. Springer, 2012, pp. 246–262.

[319] Zvika Brakerski and Shafi Goldwasser. “Circular and Leakage Resilient Public-Key Encryption under Subgroup Indistinguishability - (or: Quadratic Residuosity Strikes Back)”. In: *Advances in Cryptology - CRYPTO 2010*. Vol. 6223. LNCS. Springer, 2010, pp. 1–20.

[320] Eli Biham. “New Types of Cryptanalytic Attacks Using Related Keys”. In: *J. Cryptology* 7.4 (1994), pp. 229–246.

[321] Lars R. Knudsen. “Cryptanalysis of LOKI91”. In: *Advances in Cryptology - AUSCRYPT ’92, Workshop on the Theory and Application of Cryptographic Techniques, Gold Coast, Queensland, Australia, December 13-16, 1992, Proceedings*. Ed. by Jennifer Seberry and Yuliang Zheng. Vol. 718. Lecture Notes in Computer Science. Springer, 1992, pp. 196–208.

[322] Mihir Bellare and Tadayoshi Kohno. “A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications”. In: *Advances in Cryptology - EUROCRYPT 2003*. Vol. 2656. LNCS. Springer, 2003, pp. 491–506.

[323] Mihir Bellare and David Cash. “Pseudorandom Functions and Permutations Provably Secure against Related-Key Attacks”. In: *Advances in Cryptology - CRYPTO 2010*. 2010, pp. 666–684.

[324] Benny Applebaum, Danny Harnik, and Yuval Ishai. “Semantic Security under Related-Key Attacks and Applications”. In: *Innovations in Computer Science - ICS 2010*. 2011, pp. 45–60.

[325] Yu Chen et al. “Non-Malleable Functions and their Applications”. In: *J. Cryptol.* 35.2 (2022), p. 11.

[326] Fuyuki Kitagawa, Takahiro Matsuda, and Keisuke Tanaka. “CCA Security and Trapdoor Functions via Key-Dependent-Message Security”. In: *J. Cryptol.* 35.2 (2022), p. 9.

[327] Boaz Barak et al. “Bounded Key-Dependent Message Security”. In: *Advances in Cryptology - EUROCRYPT 2010*. Vol. 6110. LNCS. Springer, 2010, pp. 423–444.

[328] Benny Applebaum. “Key-Dependent Message Security: Generic Amplification and Completeness”. In: *Advances in Cryptology - EUROCRYPT 2011*. Vol. 6632. LNCS. Springer, 2011, pp. 527–546.

[329] Baodong Qin, Shengli Liu, and Zhengan Huang. “Key-Dependent Message Chosen-Ciphertext Security of the Cramer-Shoup Cryptosystem”. In: *Information Security and Privacy - 18th Australasian Conference, ACISP 2013*. Vol. 7959. LNCS. Springer, 2013, pp. 136–151.

[330] Sanjam Garg, Romain Gay, and Mohammad Hajiabadi. “Master-Key KDM-Secure IBE from Pairings”. In: *Public Key Cryptography (1)*. Vol. 12110. Lecture Notes in Computer Science. Springer, 2020, pp. 123–152.

[331] Shengyuan Feng, Junqing Gong, and Jie Chen. “Master-Key KDM-Secure ABE via Predicate Encoding”. In: *Public Key Cryptography (1)*. Vol. 12710. Lecture Notes in Computer Science. Springer, 2021, pp. 543–572.

[332] Jiaxin Pan, Chen Qian, and Benedikt Wagner. “Generic constructions of master-key KDM secure attribute-based encryption”. In: *Des. Codes Cryptogr.* 92.1 (2024), pp. 51–92.

[333] David Cash, Matthew Green, and Susan Hohenberger. “New Definitions and Separations for Circular Security”. In: *Public Key Cryptography - PKC 2012*. Vol. 7293. LNCS. Springer, 2012, pp. 540–557.

[334] Yu Chen et al. “KDM security for identity-based encryption: Constructions and separations”. In: *Inf. Sci.* 486 (2019), pp. 450–473.

[335] David Kolevski et al. “Cloud computing data breaches: A review of U.S. regulation and data breach notification literature”. In: *IEEE International Symposium on Technology and Society, ISTAS 2021, Waterloo, ON, Canada, October 28-31, 2021*. IEEE, 2021, pp. 1–7.

[336] Jin Wook Byun et al. “Off-Line Keyword Guessing Attacks on Recent Keyword Search Schemes over Encrypted Data”. In: *Secure Data Management, Third VLDB Workshop, SDM 2006*. Vol. 4165. LNCS. Springer, 2006, pp. 75–83.

[337] Ik Rae Jeong et al. “Constructing PEKS schemes secure against keyword guessing attacks is possible?” In: *Computer Communications* 32.2 (2009).

[338] Dennis Hofheinz and Enav Weinreb. *Searchable encryption with decryption in the standard model*. IACR Cryptology ePrint Archive, Report 2008/423. http://eprint.iacr.org/2008/423. 2008.

[339] Qiang Tang and Liqun Chen. “Public-Key Encryption with Registered Keyword Search”. In: *Public Key Infrastructures, Services and Applications - 6th European Workshop, EuroPKI 2009, Pisa, Italy, September 10-11, 2009, Revised Selected Papers*. Ed. by Fabio Martinelli and Bart Preneel. Vol. 6391. Lecture Notes in Computer Science. Springer, 2009, pp. 163–178.

[340] Rongmao Chen et al. “Server-Aided Public Key Encryption With Keyword Search”. In: *IEEE Trans. Inf. Forensics Secur.* 11.12 (2016), pp. 2833–2842.

[341] Qiong Huang and Hongbo Li. “An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks”. In: *Inf. Sci.* 403 (2017), pp. 1–14.

[342] Joonsang Baek, Reihaneh Safavi-Naini, and Willy Susilo. “Public Key Encryption with Keyword Search Revisited”. In: *Computational Science and Its Applications - ICCSA 2008*. Vol. 5072. LNCS. Springer, 2008, pp. 1249–1259.

[343] Hyun Sook Rhee et al. “Improved searchable public key encryption with designated tester”. In: *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2009*. ACM, 2009, pp. 376–379.

[344] Wei-Chuen Yau et al. “Keyword guessing attacks on secure searchable public key encryption schemes with a designated tester”. In: *Int. J. Comput. Math.* 90.12 (2013), pp. 2581–2587.

[345] Mahnaz Noroozi, Iman Karoubi, and Ziba Eslami. “Designing a secure designated server identity-based encryption with keyword search scheme: still unsolved”. In: *Ann. des Télécommunications* 73.11-12 (2018), pp. 769–776.

[346] Debiao He et al. “Certificateless Public Key Authenticated Encryption With Keyword Search for Industrial Internet of Things”. In: *IEEE Trans. Ind. Informatics* 14.8 (2018), pp. 3618–3627.

[347] Xueqiao Liu et al. “Towards Enhanced Security for Certificateless Public-Key Authenticated Encryption with Keyword Search”. In: *Provable Security - 13th International Conference, ProvSec 2019, Cairns, QLD, Australia, October 1-4, 2019, Proceedings*. Ed. by Ron Steinfeld and Tsz Hon Yuen. Vol. 11821. Lecture Notes in Computer Science. Springer, 2019, pp. 113–129.

[348] Biwen Chen et al. “Dual-Server Public-Key Authenticated Encryption with Keyword Search”. In: *IEEE Trans. Cloud Comput.* 10.1 (2022), pp. 322–333.

[349] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007. ISBN: 1584885513.

[350] Baodong Qin et al. “Public-key authenticated encryption with keyword search revisited: Security model and constructions”. In: *Inf. Sci.* 516 (2020), pp. 515–528.

[351] Mahnaz Noroozi and Ziba Eslami. “Public key authenticated encryption with keyword search: revisited”. In: *IET Information Security* 13.4 (2019), pp. 336–342.

[352] Baodong Qin et al. “Improved Security Model for Public-Key Authenticated Encryption with Keyword Search”. In: *Provable and Practical Security - 15th International Conference, ProvSec 2021, Guangzhou, China, November 5-8, 2021, Proceedings*. Ed. by Qiong Huang and Yu Yu. Vol. 13059. Lecture Notes in Computer Science. Springer, 2021, pp. 19–38.

[353] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. “The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES”. In: *Topics in Cryptology - CT-RSA 2001*. Vol. 2020. LNCS. Springer, 2001, pp. 143–158.

[354] Yu Chen, Qiang Tang, and Yuyu Wang. “Hierarchical Integrated Signature and Encryption (or Key Separation vs. Key Reuse: Enjoy the Best of Both Worlds)”. In: *Advances in Cryptology - ASIACRYPT 2021*. 2021, pp. 575–606.

[355] Adam L. Young and Moti Yung. “Auto-Recoverable Auto-Certifiable Cryptosystems”. In: *Advances in Cryptology - EUROCRYPT 1998*. Vol. 1403. Lecture Notes in Computer Science. Springer, 1998, pp. 17–31.

[356] Adam L. Young and Moti Yung. “Auto-Recoverable Cryptosystems with Faster Initialization and the Escrow Hierarchy”. In: *Public Key Cryptography, Second International Workshop on Practice and Theory in Public Key Cryptography, PKC 1999*. Vol. 1560. Lecture Notes in Computer Science. Springer, 1999, pp. 306–314.

[357] Pascal Paillier and Moti Yung. “Self-Escrowed Public-Key Infrastructures”. In: *Information Security and Cryptology - ICISC 1999*. Vol. 1787. Lecture Notes in Computer Science. Springer, 1999, pp. 257–268.

[358] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. “Pairings for cryptographers”. In: *Discret. Appl. Math.* 156.16 (2008), pp. 3113–3121.

[359] Matt Blaze, Gerrit Bleumer, and Martin Strauss. “Divertible Protocols and Atomic Proxy Cryptography”. In: *Advances in Cryptology - EUROCRYPT 1998*. Vol. 1403. Lecture Notes in Computer Science. Springer, 1998, pp. 127–144.

[360] Giuseppe Ateniese et al. “Improved proxy re-encryption schemes with applications to secure distributed storage”. In: *ACM Trans. Inf. Syst. Secur.* 9.1 (2006), pp. 1–30.

[361] Anca-Andreea Ivan and Yevgeniy Dodis. “Proxy Cryptography Revisited”. In: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2003*. The Internet Society, 2003.

[362] Georg Fuchsbauer et al. “Adaptively Secure Proxy Re-encryption”. In: *Public-Key Cryptography - PKC 2019, Proceedings, Part II*. Ed. by Dongdai Lin and Kazue Sako. Vol. 11443. Lecture Notes in Computer Science. Springer, 2019, pp. 317–346.

[363] Jiang Zhang, Zhenfeng Zhang, and Yu Chen. “PRE: Stronger security notions and efficient construction with non-interactive opening”. In: *Theor. Comput. Sci.* 542 (2014), pp. 1–16.

[364] Benoît Libert and Damien Vergnaud. “Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption”. In: *Public Key Cryptography - PKC 2008*. Ed. by Ronald Cramer. Vol. 4939. Lecture Notes in Computer Science. Springer, 2008, pp. 360–379.

[365] Jun Shao and Zhenfu Cao. “CCA-Secure Proxy Re-encryption without Pairings”. In: *Public Key Cryptography - PKC 2009*. Ed. by Stanislaw Jarecki and Gene Tsudik. Vol. 5443. Lecture Notes in Computer Science. Springer, 2009, pp. 357–376.

[366] Elena Kirshanova. “Proxy Re-encryption from Lattices”. In: *Public-Key Cryptography - PKC 2014*. Ed. by Hugo Krawczyk. Vol. 8383. Lecture Notes in Computer Science. Springer, 2014, pp. 77–94.

[367] Xiong Fan and Feng-Hao Liu. “Proxy Re-Encryption and Re-Signatures from Lattices”. In: *Applied Cryptography and Network Security - 17th International Conference, ACNS 2019*. Ed. by Robert H. Deng et al. Vol. 11464. Lecture Notes in Computer Science. Springer, 2019, pp. 363–382.

[368] Yunxiao Zhou et al. “Fine-Grained Proxy Re-encryption: Definitions and Constructions from LWE”. In: *Advances in Cryptology - ASIACRYPT 2023, Proceedings, Part VI*. Ed. by Jian Guo and Ron Steinfeld. Vol. 14443. Lecture Notes in Computer Science. Springer, 2023, pp. 199–231.

[369] *PKCS#1: RSA Cryptography Specifications Version 2.2*. https://www.rfc-editor.org/rfc/rfc8017.html.

[370] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. “DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem”. In: (1999). https://eprint.iacr.org/1999/007.

[371] *SEC 1: Elliptic Curve Cryptography Ver. 1.0*. https://www.secg.org/SEC1-Ver-1.0.pdf.

[372] *Digital Signature Standard (DSS)*. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf.

[373] *SEC 2: Recommended Elliptic Curve Domain Parameters*. https://www.secg.org/sec2-v2.pdf.

[374] *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation*. https://datatracker.ietf.org/doc/html/rfc5639.

[375] *SM2 椭圆曲线公钥密码算法*. https://www.rfc-editor.org/rfc/rfc8017.html.

[376] *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. https://csrc.nist.gov/pubs/fips/203/ipd.

[377] Joppe W. Bos et al. “CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM”. In: *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018*. IEEE, 2018, pp. 353–367.

[378] *IT Security Techniques –Encryption algorithms - Part 6: Homomorphic Encryption*. https://www.iso.org/standard/67740.html.

[379] Martin Albrecht et al. *Homomorphic Encryption Security Standard*. Tech. rep. https://homomorphicencryption.org/wp-content/uploads/2018/11/HomomorphicEncryptionStandardv1.1.pdf. HomomorphicEncryption.org, 2018.

[380] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. “(Leveled) Fully Homomorphic Encryption without Bootstrapping”. In: *ACM Trans. Comput. Theory* 6.3 (2014), 13:1–13:36.

[381] Zvika Brakerski. “Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP”. In: *Advances in Cryptology - CRYPTO 2012*. Vol. 7417. Lecture Notes in Computer Science. Springer, 2012, pp. 868–886.

[382] Junfeng Fan and Frederik Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. IACR Cryptol. ePrint Arch. http://eprint.iacr.org/2012/144. 2012.

[383] Jung Hee Cheon et al. “Homomorphic Encryption for Arithmetic of Approximate Numbers”. In: *Advances in Cryptology - ASIACRYPT 2017*. Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 409–437.

[384] Léo Ducas and Daniele Micciancio. “FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second”. In: *Advances in Cryptology - EUROCRYPT 2015*. Vol. 9056. Lecture Notes in Computer Science. Springer, 2015, pp. 617–640.

[385] Ilaria Chillotti et al. “TFHE: Fast Fully Homomorphic Encryption Over the Torus”. In: *J. Cryptol.* 33.1 (2020), pp. 34–91.

[386] https://github.com/openssl.

[387] https://github.com/alibaba-edu/mpc4j/.

[388] *Kunlun*. https://github.com/yuchen1024/Kunlun.

[389] Shashank Agrawal and Melissa Chase. “FAME: Fast Attribute-based Message Encryption”. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017*. ACM, 2017, pp. 665–682.

[390] *SEAL*. https://github.com/microsoft/SEAL.

[391] *OpenFHE*. https://github.com/openfheorg/openfhe-development.