浅谈哈希证明系统

陈宇

2017 年 11 月 26 日

引言

DENG 神已经对零知识证明系统 (Zero Knowledeg Proof) 做了极为精彩的科普, 在这里我浅谈零知识证明系统的一个小弟–哈希证明系统进行, 回答以下三个问题:

- 从哪里来?
- 究竟是谁?
- 到哪里去?

零知识证明系统的一个重要性质是任何验证者均可快速验证一个证明是否有效, 即公开可验证性. 这一强性质恰是构造高效零知识证明系统的障碍之一. 一个自然的问题是能否弱化该性质使得高效的构造成为可能, 答案是肯定的.

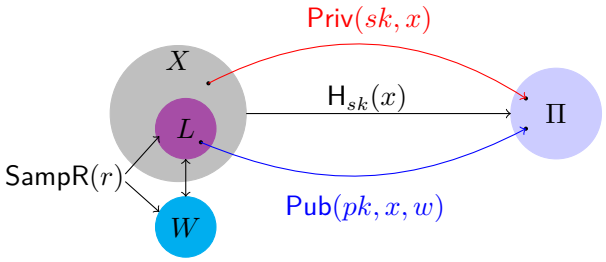
1 从哪里来

1998 年, Cramer 和 Shoup [CS98] 给出了首个标准模型下高效的选择密文安全的公钥加密方案, 简称 CS98 方案. 该方案的设计新颖独特, 结构精巧有如神来之笔, 大家看了均表示不明觉厉.

2002 年, Cramer 和 Shoup 将 CS98 方案的设计思想凝练抽象为哈希证明系统 (Hash Proof System, HPS) [CS02], 至此密码学的军火库又新添了一个强有力的武器, 其威力和影响远远超出最初的选择密文安全公钥加密.

2 究竟是谁

哈希证明系统的名字十分酷炫, 它到底是啥呢? 一句话: 是指定验证者的非交互式零知识证明系统, 因其证明为哈希值, 故名哈希证明系统. 看公式和符号容易头晕, 所以我把它们画成了下图:



首先介绍语言系统的概念. 令 X 是一个实例集合, L 是 X 中由某二元关系 R_L 定义的一个真子集, 即 $x \in L$ 当且仅当存在 $w \in W$ 使得 $(x, w) \in R_L$, 那么 (X, L, W, R_L) 构成了一个语言系统, L 为语言集合, W 为证据集合. L 中的实例常称为 Yes 实例, L 之外的实例称为 No 实例.

为了有密码学上的应用, 通常要求语言系统上存在所谓的子集成员不可区分问题 (SMP): 一个随机的 Yes 实例和一个随机的 No 实例是计算不可区分的.

举个例子, X 好比全体人类的集合, L 是其中所有好人的集合, 每个好人都有一张好人卡 (证据), $X \backslash L$ 是坏人集合, 坏人呢? 没有卡... 因为好人和坏人头上都没写字, 所以随便拉一人你很难区分他是好人还是坏人.

关于语言系统 (X, L, W, R_L) 的哈希证明系统由三个算法组成:

- 密钥生成算法 Gen: 生成一对密钥 (pk, sk), sk 与 pk 之间存在多对一的投射关系, 即 $\alpha(sk) = pk$. 每个 sk 都定义了一个哈希函数 $\mathsf{H}_{sk} : X \to \Pi$, 其中 X 是实例空间, Π 是证明空间.
- 私有求值算法 Priv: 拥有 sk 时可以高效计算 $\mathsf{H}_{sk}(x)$, 其中 x 是 X 中的任意实例.
- 公开求值算法 Pub: 当 x 属于语言 L 时, 可以在没有 sk 的情况下根据 pk 和 $x \in L$ 的证据 w 计算出哈希值 $\mathsf{H}_{sk}(x)$.

哈希函数 H_{sk} 的以下性质刻画了其在 X 上的截然不同的两种行为:

投射性: $\forall x \in L$, $\mathsf{H}_{sk}(x) = \mathsf{Pub}(pk, x, w)$, 其中 $pk = \alpha(sk)$. 该性质刻画的是当哈希函数的输入为语言中的元素时, 其输出由对应的公钥和元素本身惟一确定, 与私钥无关. 可以理解为好人太老实, 总是一根筋, 不管私钥 sk 是啥, 哈希值都不变.

均匀性: $\forall x \notin L$, $\mathsf{H}_{sk}(x)$ 在 Π 上均匀分布. 该性质刻画的是当哈希函数的输入不在语言中时, 其输出在 Π 中随机分布. 可以理解为坏人花样百出, 私钥 sk 不同, 哈希值也不同.

语言系统上的困难问题告诉我们 L 中的随机元素和 L 外的随机元素是计算不可区分的, 这一点是联系两大性质的纽带.

从上面的定义似乎看不出证明系统的影子, 看完下图就了然了.

$$\mathsf{Gen}(\lambda) \to (pk, sk)$$
$$\downarrow$$
$$\mathsf{Gen}(\lambda) \to (crs, td)$$

$$\mathsf{Prover}(x, w) \xrightarrow{x,\ \pi \leftarrow \mathsf{H}_{sk}(x) = \mathsf{Pub}(pk, x, w)} \mathsf{Verifier}(sk)$$
$$\pi \stackrel{?}{=} \mathsf{H}_{sk}(x) \text{ via } \mathsf{Priv}(sk, x)$$

1. 可信第三方运行密钥生成算法, 生成一对公私钥 (pk, sk), 将 pk 作为公共参考串 crs, sk 作为陷门 td 发送给验证者 V
2. 证明者 P 如何向 V 证明 $x \in L$(x 是一个好人) 呢? 运行公开求值算法 $\mathsf{Pub}(pk, x, w)$ 计算哈希值 $\pi \leftarrow \mathsf{H}_{sk}(x)$ 并发送给 V, 证明本身就是实例的哈希值!
3. V 如何验证呢? 利用从可信第三方处获得的 sk 计算哈希值并与 P 发送过来的进行比对, 相等就接受, 否则拒绝.

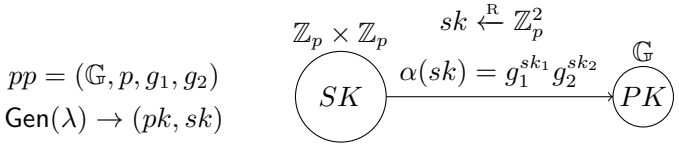
上述协议显然是非交互的, 下面逐一验证证明系统的性质:

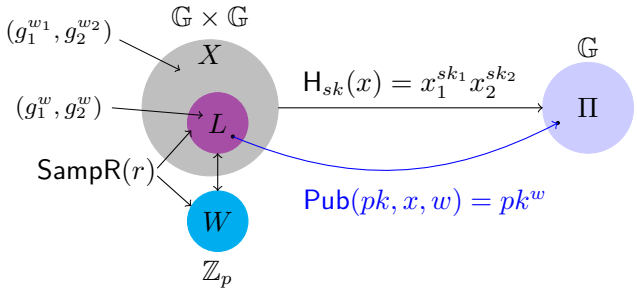
- 完备性: 当 $x \in L$ 时, 哈希函数的投射性保证了 P 给出的证明 V 都会接受.
- 合理性: 当 $x \notin L$ 时, 哈希函数的均匀性保证了即使 P 拥有无穷的计算能力, 输出正确哈希值的概率也是可忽略的.

- (指定验证者) 零知识性: 由于 V 本身拥有 sk, 它从 P 处得到的证明自己就能够独立计算出来, 证明的零知识性是显然成立的. 此外, 只有拥有 sk 才有能力验证一个证明的真伪.

以上的条分缕析说明哈希证明系统就一个指定验证者的非交互式零知识证明系统.

为了让大家有更直观的认识, 以下给出一个最简单的哈希证明系统的实例 (基于判定性 Diffie—Hellman 假设的 universal HPS, 也称为 Cramer-Shoup Lite HPS).





3 到哪里去

花了很长的篇幅说了哈希证明系统究竟是啥, 可能很多小伙伴都晕了, 其实以上仅仅是极其精要的概述, 很多细节已经略去了, 有兴趣大家可以读原文 [CS02].

当今干啥都讲究有用, 那下面就简单说说哈希证明系统有哪些神奇的、不可替代的应用. 在此之前, 首先给出哈希证明系统的两个局限和两点观察:

两个局限

1. 证明不具有公开可验证性

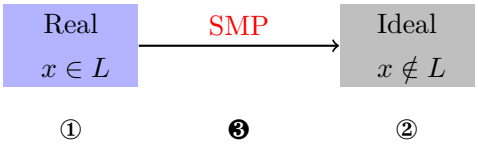
2. 证明的表达能力有限, 目前仅能对证明群中的子群成员归属问题, 尚未知能否延伸到任意的 $\mathcal{NP}$ 语言. 在很多具体的零知识证明应用场合, 公开验证性和强大的表达能力均不是必须, 因此用标准的零知识证明系统就是高射炮打蚊子——大材小用, 哈希证明系统可以做的更快更好! (效率的优势恰恰源自局限)

两点观察

1. 哈希证明系统中的证明者是高效的, 即证明者在拥有相应的证据 w(好人卡) 时, 可以快速计算出证明. 这一点是不仅是哈希证明系统, 也是所有证明系统有用的前提条件.

2. 私钥与实例的无关性: 即使验证者拥有一个私钥, 也无法判定给定的实例是否属于 L. 该性质可以理解为一个人是好人还是坏人和判定的方法无关. 该性质尤为重要.

所有基于哈希证明系统的密码方案的安全性建立均是三板斧套路:



1. 真实的方案设计选取语言中的实例 $x \in L$

2. 想象中的方案设计选取语言外的实例 $x \notin L$

我们利用哈希函数的均匀性说明证明中方案的安全性, 再利用语言系统的不可区分性说明敌手无法察觉实例的真伪切换, 最终证明真实的方案是安全的. 特别值得指出的是, 基于哈希证明系统的方案证明集中体现了混合论证的证明方法 (hybrid argument)、计算意义下的归约和信息论意义下的证明, 是可证明安全从入门到精通的最佳切入点.

在设计可证明安全的密码方案特别是公钥加密方案中的一个主要技术难点是: 私钥往往嵌入在底层的困难问题中, 因此归约算法通常并不掌握私钥, 但归约算法又必须能够回答敌手关于私钥的询问. 一个很好的例子就是难以证明 ElGamal 类型加密满足选择密文安全性, 因为私钥嵌入在底层 Diffie-Hellman 困难问题中.

哈希证明系统另辟蹊径, 直接绕过该难点, 关窍就在于私钥与实例无关, 归约算法始终拥有私钥，因此可以轻松回答关于私钥的任意询问.

这一特性使得哈希证明系统成为构造以下密码方案的利器:

- 高安全性公钥加密, 包括选择密文安全、密钥泄漏安全、密钥篡改安全、选择打开攻击安全, 特别的, 它几乎是构造抗泄漏公钥加密方案的不二法门
- 密钥交换协议: 包括基于口令的密钥交换和认证密钥交换
- 不经意传输协议

哈希证明系统还有诸多变体, 如:

- 基于身份的哈希证明系统 [CZLC12]: 用于构造基于身份的加密
- 同态的哈希证明系统 [Wee16]: 用于构造消息依赖密钥的公钥加密方案
- 可延展的哈希证明系统 [CMY+16]: 密码逆向防火墙
- 可更新的哈希证明系统 [YXZ+15]: 抗持续泄漏的公钥加密
- 对偶的哈希证明系统 [Wee12]: 这个厉害了, 直接蕴含另一核武器—有损陷门函数

4 总结

哈希证明系统是零知识证明系统一个小弟, 胜在结构简洁优美, 实例切换加信息论意义下论证的三板斧套路极其有用, 从铜锣湾砍到尖沙咀. 低配版的小弟都这么厉害, 那大哥就猛得没边了.

参考文献

[CMY+16] Rongmao Chen, Yi Mu, Guomin Yang, Willy Susilo, Fuchun Guo, and Mingwu Zhang. Cryptographic reverse firewall via malleable smooth projective hash functions. In *Advances in Cryptology - ASIACRYPT 2016*, volume 10031 of *Lecture Notes in Computer Science*, pages 844–876, 2016.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Advances in Cryptology - CRYPTO 1998*, volume 1462 of *LNCS*, pages 13–25. Springer, 1998.

[CS02] 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*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.

[CZLC12] Yu Chen, Zongyang Zhang, Dongdai Lin, and Zhenfu Cao. Anonymous identity-based hash proof system and its applications. In *Provable Security - 6th International Conference, ProvSec 2012*, volume 7496 of *LNCS*, pages 143–160. Springer, 2012.

[Wee12] Hoeteck Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 246–262. Springer, 2012.

[Wee16] Hoeteck Wee. Kdm-security via homomorphic smooth projective hashing. In *Public-Key Cryptography - PKC 2016*, volume 9615 of *LNCS*, pages 159–179. Springer, 2016.

[YXZ+15] Rupeng Yang, Qiuliang Xu, Yongbin Zhou, Rui Zhang, Chengyu Hu, and Zuoxia Yu. Updatable hash proof system and its applications. In *Computer Security - ESORICS 2015*, volume 9326 of *Lecture Notes in Computer Science*, pages 266–285. Springer, 2015.