Leakage-Resilient Cryptography from Puncturable Primitives and Obfuscation

Yu Chen[1] Yuyu Wang[2] Hong-Sheng Zhou[3]

[1]SKLOIS-IIE-CAS, UCAS

[2]Tokyo Institute of Technology, IOHK, AIST

[3]Virginia Commonwealth University

ASIACRYPT 2018

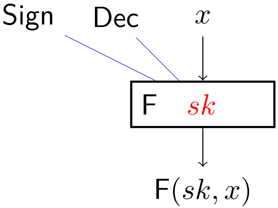
Dec. 5th 2018

Outline

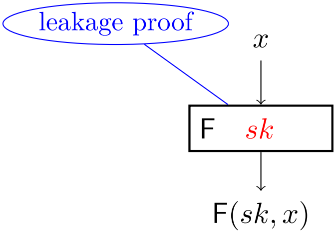
1. Background
2. Motivation
3. Primitives
4. Our Framework Towards Leakage-Resilience
   - Leakage-Resilient PKE
   - Leakage-Resilient SKE
   - Leakage-Resilient Signature
5. Achieving Optimal Leakage Rate

Outline

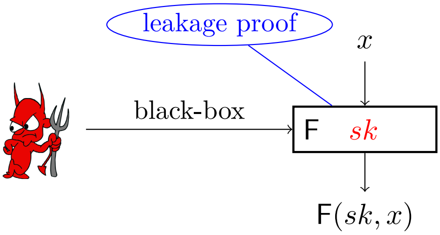
Leakage-Resilient Cryptography



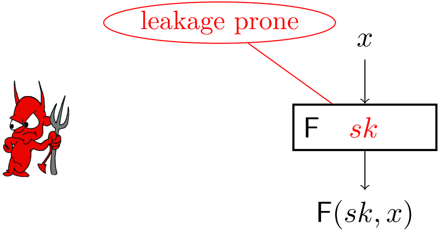
Leakage-Resilient Cryptography



Leakage-Resilient Cryptography

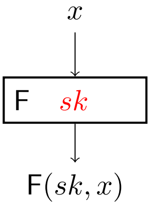


Leakage-Resilient Cryptography



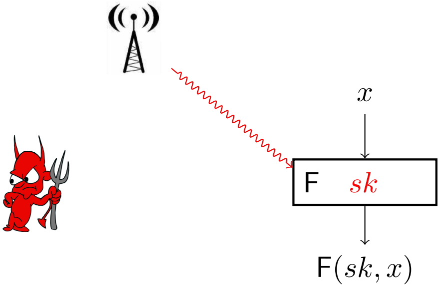
Leakage-Resilient Cryptography

leakage attacks (since 1996) invalidate this idealized assumption



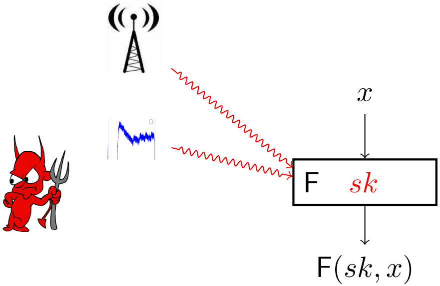
Leakage-Resilient Cryptography

leakage attacks (since 1996) invalidate this idealized assumption



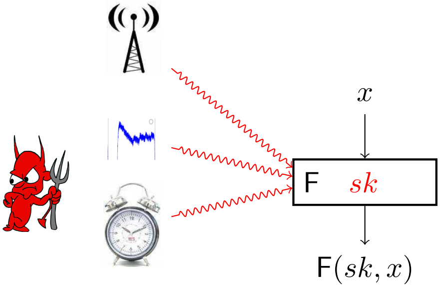
Leakage-Resilient Cryptography

leakage attacks (since 1996) invalidate this idealized assumption



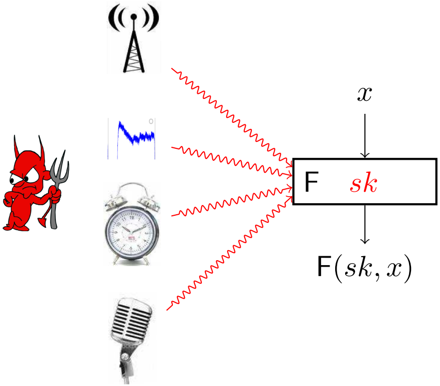
Leakage-Resilient Cryptography

leakage attacks (since 1996) invalidate this idealized assumption



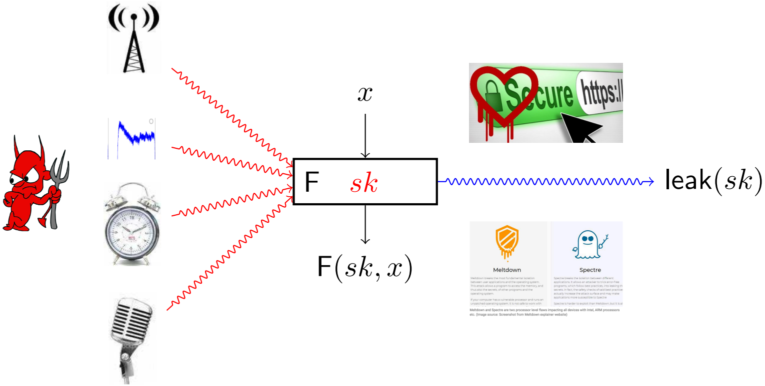
Leakage-Resilient Cryptography

leakage attacks (since 1996) invalidate this idealized assumption



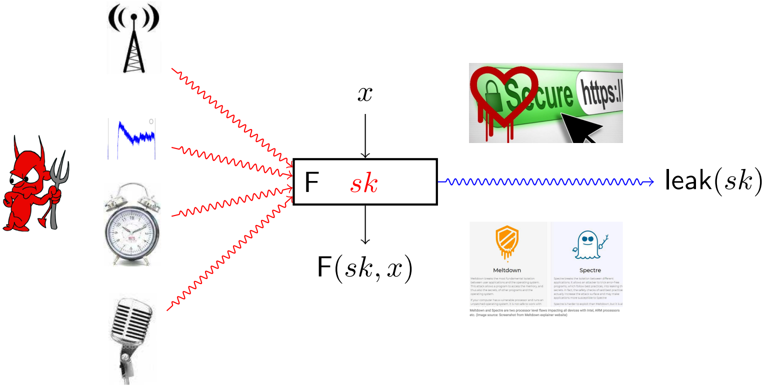
Leakage-Resilient Cryptography

leakage attacks (since 1996) invalidate this idealized assumption



Leakage-Resilient Cryptography

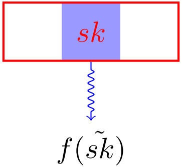
leakage attacks (since 1996) invalidate this idealized assumption



- Leakage-Resilient Cryptography: provably secure against *all* leakage attacks captured by leakage model.

Leakage Models

Various leakage models in the literature, differing in their specifications of leakage source/functions/behaviors:



- Only computation leaks model: [MR04]...
- Bounded leakage model: [AGV09, KV09, NS09, ADW09, ADN+10, QL13, CQX18]...
- Auxiliary input model: [DKL09, DGK+10]...
- Continual leakage model: [BKKV10, DHLW10]...

Bounded Leakage Model

In this work, we focus on the most basic bounded leakage model

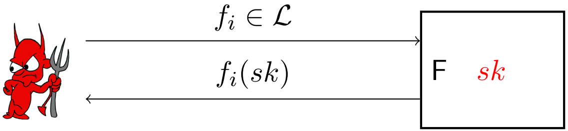
- conceptually simple yet general enough
- results in BLM used as building blocks for leakage-resilient schemes in more complex leakage models

Bounded Leakage Model

In this work, we focus on the most basic bounded leakage model

- conceptually simple yet general enough
- results in BLM used as building blocks for leakage-resilient schemes in more complex leakage models

A template of BLM



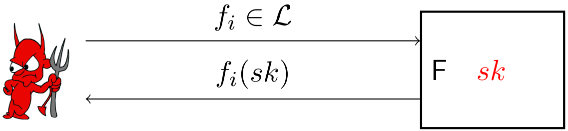
$$\sum |f_i(sk)| < \ell \leq |sk|$$

Bounded Leakage Model

In this work, we focus on the most basic bounded leakage model

- conceptually simple yet general enough
- results in BLM used as building blocks for leakage-resilient schemes in more complex leakage models

A template of BLM



$$\sum |f_i(sk)| < \ell \leq |sk|$$

- leakage ratio $\rho = \ell/|sk| \rightsquigarrow 1 - o(1)$ is optimal

Outline

Leakage-Resilient Workhorse Primitives

In the last two decades, a broad range of LR cryptographic schemes have been proposed.

But, several interesting problems are still open around *lower-level, workhorse* primitives, such as SKE, PKE and Signature

Leakage-Resilient SKE

LR SKE can be reduced to constructing LR wPRF

- Pietrzak [Pie09], Dodis and Yu [DY13]: any PRF is already leakage-resilient against $\ell = O(\log \lambda)$-bit leakage
- Hazay et al. [HLWW13]: OWF $\Rightarrow$ LR wPRF with leakage rate $O(\log \lambda)/|sk|$

Is there a generic construction of LR wPRF with optimal leakage rate?

Leakage-Resilient PKE

Existing LR PKE are based on either specific assumptions such as LWE [AGV09] and QR [BG10], or more generally the hash proof system [NS09]

Whether the classic construction of PKE based on TDF/TDR can be made LR? Is there a generic construction of LR PKE?

CCA security vs. leakage-resilience (dual)

- CCA: $\mathcal{A}$ learns sk via a specific family of functions (tie to $\mathsf{Dec}(sk, \cdot)$) with unbounded output length
- LR: $\mathcal{A}$ learns sk via arbitrary functions with bounded output length

Is there a connection between CCA security and LR?

Leakage-Resilient Signature

Challenging problem: fully leakage-resilience – EUF-CMA remains in the presence of both secret key and random coins leakage

- when Sign is deterministic or public-coin: standard LR $\Rightarrow$ FLR

All the known FLR Sigs [BSW11, MTVY11, LLW11, GJS11] are randomized and secret-coin.

Boyle et al. [BSW11] left the open problem

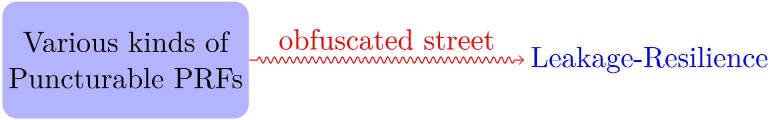
Do there exist deterministic or public-coin LR signatures?

Bonus: such kind of Sig remain secure even all the random coins are revealed

This Work

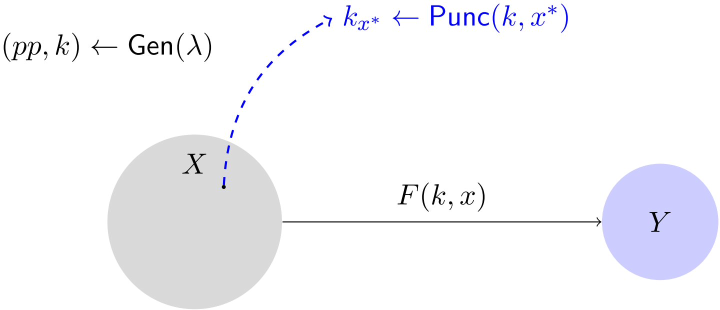
Our goal: Generic constructions of LR encryption and signature with optimal leakage rate (in the bounded leakage model)

Our major insight



Outline

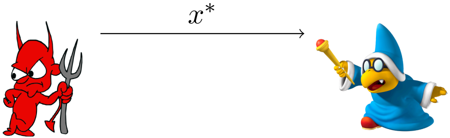
Puncturable PRF [SW14]



$$\mathsf{Eval}(k_{x^*}, x) = F(k, x) \text{ for } x \neq x^*$$

Selective Puncturable PRF

Selective Puncturable PRF



Selective Puncturable PRF

x^* $\longrightarrow$

$$(pp, k) \leftarrow \mathsf{Gen}(\lambda)$$
$$k_{x^*} \leftarrow \mathsf{Punc}(k, x^*)$$
$$y_0^* \leftarrow F(k, x^*)$$
$$y_1^* \xleftarrow{\text{R}} Y$$

Selective Puncturable PRF

$\beta \xleftarrow{R} \{0,1\}$

x^* $\longrightarrow$

x^*, k_{x^*}, y^*_β $\longleftarrow$

$$(pp, k) \leftarrow \mathsf{Gen}(\lambda)$$
$$k_{x^*} \leftarrow \mathsf{Punc}(k, x^*)$$
$$y^*_0 \leftarrow F(k, x^*)$$
$$y^*_1 \xleftarrow{R} Y$$

Selective Puncturable PRF

$\beta \xleftarrow{R} \{0,1\}$

x^* $\longrightarrow$

$\longleftarrow$ x^*, k_{x^*}, y^*_β

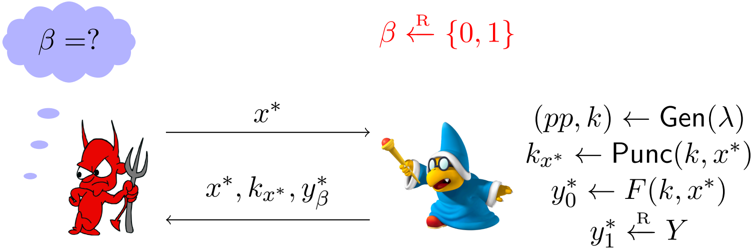
$(pp, k) \leftarrow \mathsf{Gen}(\lambda)$

$k_{x^*} \leftarrow \mathsf{Punc}(k, x^*)$

$y^*_0 \leftarrow F(k, x^*)$

$y^*_1 \xleftarrow{R} Y$

Selective Puncturable PRF



- directly implied by GGM-PRF $\Leftarrow$ OWF

Weak Puncturable PRF

Weak Puncturable PRF

$$(pp, k) \leftarrow \mathsf{Gen}(\lambda)$$
$$x^* \xleftarrow{\text{R}} X$$
$$k_{x^*} \leftarrow \mathsf{Punc}(k, x^*)$$
$$y_0^* \leftarrow F(k, x^*)$$
$$y_1^* \xleftarrow{\text{R}} Y$$

Weak Puncturable PRF

$$\beta \xleftarrow{R} \{0,1\}$$

$$\xleftarrow{\quad pp,\, x^*,\, k_{x^*},\, y^*_\beta \quad}$$

$$(pp, k) \leftarrow \mathsf{Gen}(\lambda)$$
$$x^* \xleftarrow{R} X$$
$$k_{x^*} \leftarrow \mathsf{Punc}(k, x^*)$$
$$y^*_0 \leftarrow F(k, x^*)$$
$$y^*_1 \xleftarrow{R} Y$$

Weak Puncturable PRF

$\beta \xleftarrow{R} \{0,1\}$

$pp, x^*, k_{x^*}, y^*_\beta$ ←

$$(pp, k) \leftarrow \mathsf{Gen}(\lambda)$$
$$x^* \xleftarrow{R} X$$
$$k_{x^*} \leftarrow \mathsf{Punc}(k, x^*)$$
$$y^*_0 \leftarrow F(k, x^*)$$
$$y^*_1 \xleftarrow{R} Y$$

Weak Puncturable PRF

$\beta =?$

$\beta \xleftarrow{R} \{0,1\}$

$pp, x^*, k_{x^*}, y^*_\beta$

$\longleftarrow$

$$(pp, k) \leftarrow \mathsf{Gen}(\lambda)$$
$$x^* \xleftarrow{R} X$$
$$k_{x^*} \leftarrow \mathsf{Punc}(k, x^*)$$
$$y^*_0 \leftarrow F(k, x^*)$$
$$y^*_1 \xleftarrow{R} Y$$

Theorem: $sPPRF \Leftrightarrow wPPRF$

Indistinguishability Obfuscation [BGI+12]

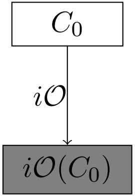
A uniform PPT machine $i\mathcal{O}$ is called an indistinguishability obfuscator if:

Indistinguishability Obfuscation [BGI+12]

A uniform PPT machine $i\mathcal{O}$ is called an indistinguishability obfuscator if:

- Preserving Functionality: $\forall C \in \mathcal{C}_\lambda$, $\forall x \in \{0,1\}^*$

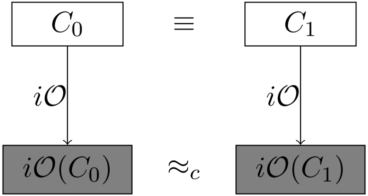
$$\Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(C)] = 1$$



Indistinguishability Obfuscation [BGI+12]

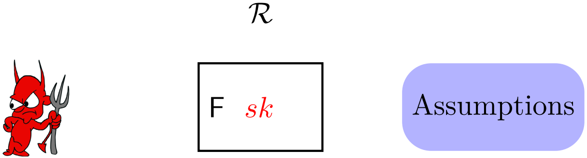
A uniform PPT machine $i\mathcal{O}$ is called an indistinguishability obfuscator if:

- Preserving Functionality: $\forall C \in \mathcal{C}_\lambda$, $\forall x \in \{0,1\}^*$
$$\Pr[C'(x) = C(x) : C' \leftarrow i\mathcal{O}(C)] = 1$$
- Indistinguishability of Obfuscation
$\forall$ PPT adversaries $(\mathcal{S}, \mathcal{D})$, $\exists$ a negl. function α:
$\Pr[\forall x, C_0(x) = C_1(x) : (C_0, C_1, aux) \leftarrow \mathcal{S}(\lambda)] \geq 1 - \alpha(\lambda) \Rightarrow$
$$|\Pr[\mathcal{D}(aux, i\mathcal{O}(C_0)) = 1] - \Pr[\mathcal{D}(aux, i\mathcal{O}(C_1)) = 1]| \leq \alpha(\lambda)$$

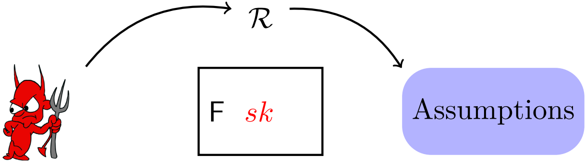


Outline

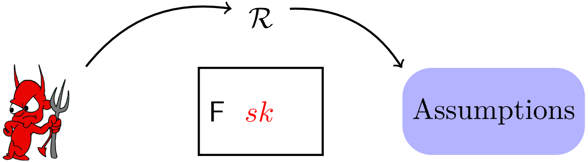
Approaches towards Leakage Resilience



Approaches towards Leakage Resilience

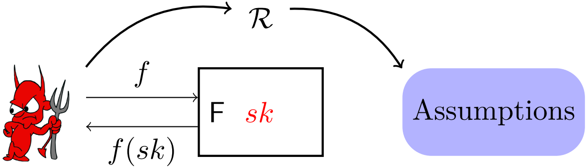


Approaches towards Leakage Resilience



Technical hurdle: a seemingly paradox

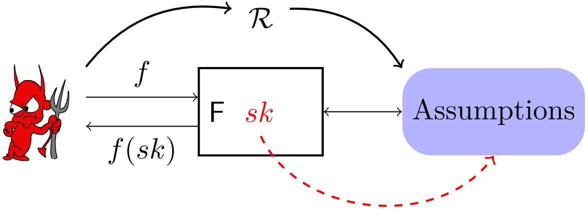
Approaches towards Leakage Resilience



Technical hurdle: a seemingly paradox

- In order to answer *arbitrary* leakage queries, it seems $\mathcal{R}$ must know sk

Approaches towards Leakage Resilience

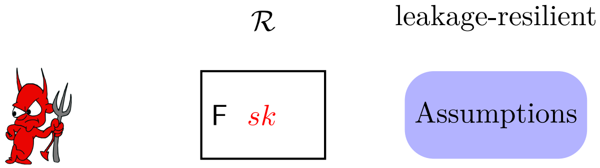


Technical hurdle: a seemingly paradox

- In order to answer *arbitrary* leakage queries, it seems $\mathcal{R}$ must know sk
- Typically $\mathcal{R}$ does not know sk since the challenge instance is embedded in it

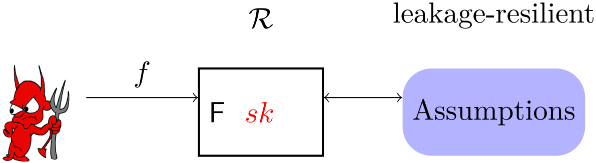
Approach I

Rely on leakage-resilient assumptions, i.e., the assumption still holds even in the presence of partial leakage of secret



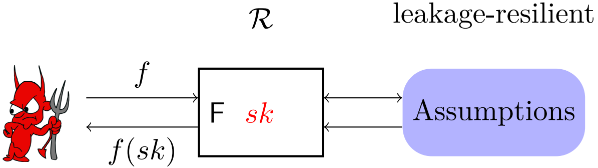
Approach I

Rely on leakage-resilient assumptions, i.e., the assumption still holds even in the presence of partial leakage of secret



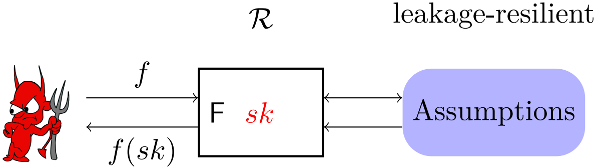
Approach I

Rely on leakage-resilient assumptions, i.e., the assumption still holds even in the presence of partial leakage of secret



Approach I

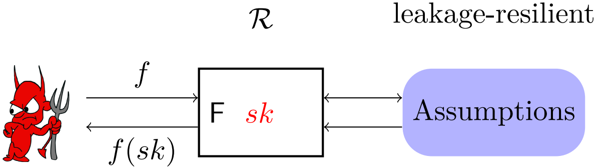
Rely on leakage-resilient assumptions, i.e., the assumption still holds even in the presence of partial leakage of secret



- Katz and Vaikuntanathan [KV09]: UOWHF is LR-OW + ss-NIZK $\Rightarrow$ LR SIG

Approach I

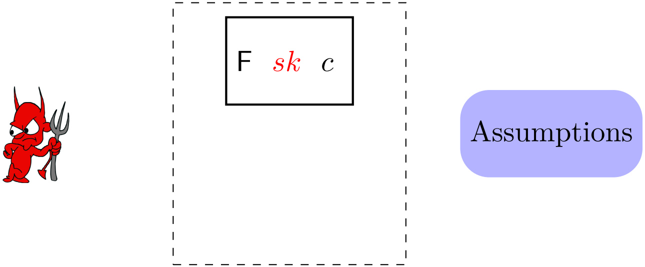
Rely on leakage-resilient assumptions, i.e., the assumption still holds even in the presence of partial leakage of secret



- Katz and Vaikuntanathan [KV09]: UOWHF is LR-OW + ss-NIZK $\Rightarrow$ LR SIG
- Akavia et al. [AGV09]: normal pk $\approx_c$ lossy pk even in the presence of sk leakage $\Rightarrow$ Regev PKE is LR

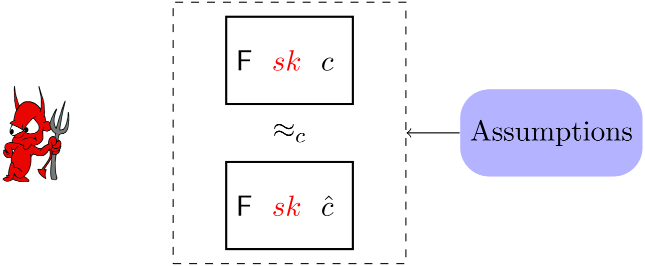
Approach II

detached strategy + leakage-resilient assumptions/facts



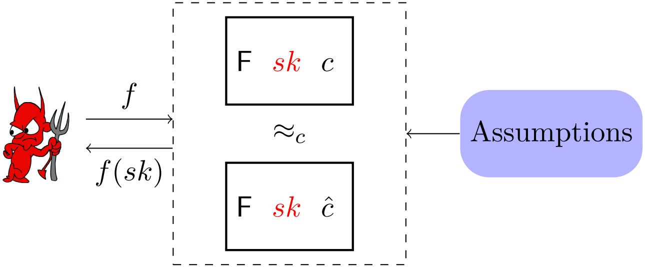
Approach II

detached strategy + leakage-resilient assumptions/facts



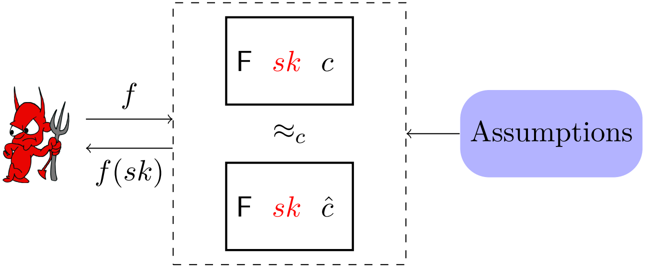
Approach II

detached strategy + leakage-resilient assumptions/facts



Approach II

detached strategy + leakage-resilient assumptions/facts

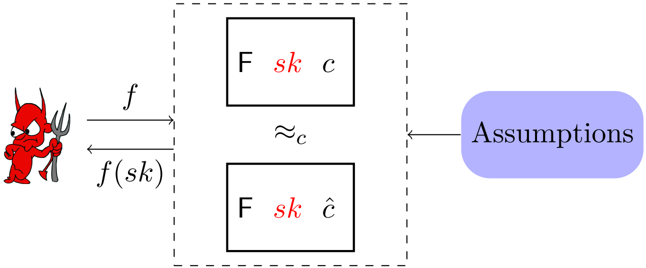


- Naor and Segev [NS09]: SMP $\Rightarrow c \approx_c \hat{c}$; $k \leftarrow \mathsf{Ext}(sk, \hat{c})$

  leftover hash lemma (leakage-resilient fact)

Approach II

detached strategy + leakage-resilient assumptions/facts



- Naor and Segev [NS09]: SMP $\Rightarrow c \approx_c \hat{c}$; $k \leftarrow \mathsf{Ext}(sk, \hat{c})$

  leftover hash lemma (leakage-resilient fact)

- Dodis et al. [DGK+10]: DDH $\Rightarrow c \approx_c \hat{c}$; $k \leftarrow \mathsf{hc}_{\hat{c}}(sk)$ w.r.t. f (auxliary-input model)

  Goldreich-Levin theorem (leakage-resilient assumption)

A common theme of the two above main approaches

- $\mathcal{R}$ always try to simulate leakage oracle *perfectly*, i.e., answering leakage queries with *real* secret key.

To do so, we have to either rely on LR assumptions or resort to sophisticated design with specific structure.

It is interesting to investigate the possibility of

> simulate leakage oracle *computationally*, i.e., answering leakage queries with simulated leakage

This might lend new techniques to address the unsolved problems in LRC.

Dachman-Soled et al. [DGL+16] discovered powerful applications of $i\mathcal{O}$ to LRC

- Sahai-Waters PKE $\leadsto$ leakage resilient

Background: Sahai-Waters KEM

Ingredients: $i\mathcal{O}$, PRG $\mathsf{G} : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$, weak puncturable PRF $F : SK \times \{0,1\}^{2\lambda} \to Y$

- $\mathsf{Gen}(\lambda)$: pick $sk \xleftarrow{R} SK$, $pk \leftarrow i\mathcal{O}(\text{Encaps})$
- $\mathsf{Encaps}(pk; r)$: $(c, k) \leftarrow pk(r)$
- $\mathsf{Decaps}(sk, c)$: $k \leftarrow F(sk, c)$

Encaps
Constants: PPRF key sk
Input: randomness $r \in \{0,1\}^\lambda$
1. compute $x \leftarrow \mathsf{G}(r)$; output $c = x$, $k \leftarrow F(sk, x)$

Why Sahai-Waters is not Leakage-Resilient?

The proof uses "punctured programs" technique and security is reduced to the weak pseudorandomness of punctured PRF

$$pk \leftarrow i\mathcal{O}(\text{Encaps}(sk)) \rightsquigarrow pk \leftarrow i\mathcal{O}(\text{Encaps}^*(sk_{x^*}))$$
$$\text{session key } k^* \leftarrow y^* \leftarrow F(sk, x^*), \text{ where } x^* \xleftarrow{\text{R}} \{0,1\}^{2\lambda}$$

Why Sahai-Waters is not Leakage-Resilient?

The proof uses "punctured programs" technique and security is reduced to the weak pseudorandomness of punctured PRF

$$pk \leftarrow i\mathcal{O}(\text{Encaps}(sk)) \rightsquigarrow pk \leftarrow i\mathcal{O}(\text{Encaps}^*(sk_{x^*}))$$
$$\text{session key } k^* \leftarrow y^* \leftarrow F(sk, x^*), \text{ where } x^* \xleftarrow{R} \{0,1\}^{2\lambda}$$

The sources for non-leakage-resilient

- Construction perspective: the information of y^* could be leaked via leakage queries on sk, and thus may not be random anymore in $\mathcal{A}$'s view.
- Proof perspective: in some hybrid game, $\mathcal{R}$ only knows sk_{x^*}, and thus unable to handle arbitrary leakage queries.

Why Sahai-Waters is not Leakage-Resilient?

The proof uses "punctured programs" technique and security is reduced to the weak pseudorandomness of punctured PRF

$$pk \leftarrow i\mathcal{O}(\text{Encaps}(sk)) \leadsto pk \leftarrow i\mathcal{O}(\text{Encaps}^*(sk_{x^*}))$$
$$\text{session key } k^* \leftarrow y^* \leftarrow F(sk, x^*), \text{ where } x^* \xleftarrow{\text{R}} \{0,1\}^{2\lambda}$$

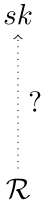
The sources for non-leakage-resilient

- Construction perspective: the information of y^* could be leaked via leakage queries on sk, and thus may not be random anymore in $\mathcal{A}$'s view.
- Proof perspective: in some hybrid game, $\mathcal{R}$ only knows sk_{x^*}, and thus unable to handle arbitrary leakage queries.

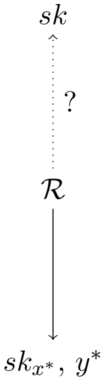
Dachman-Soled et al. [DGL+16] made Sahai-Waters KEM leakage-resilient by using $i\mathcal{O}$ twice.

Outline

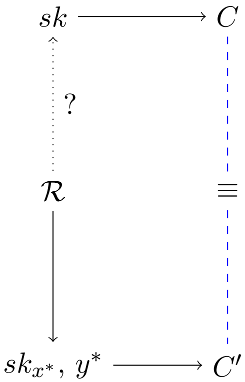
Abstract and Generalize the Core Idea



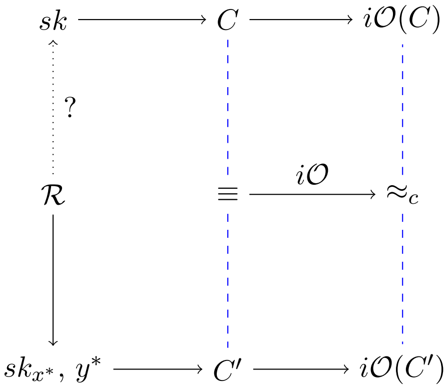
Abstract and Generalize the Core Idea



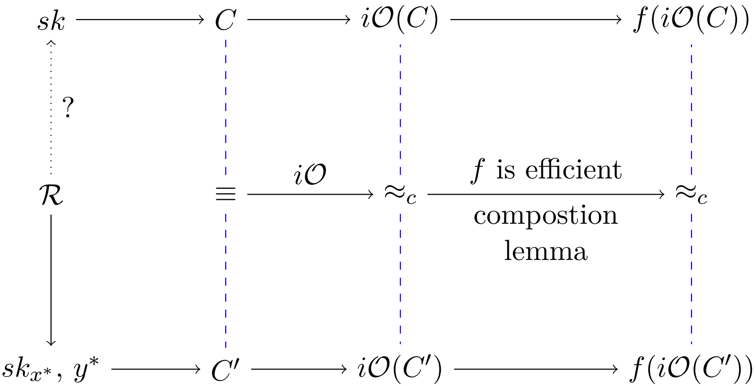
Abstract and Generalize the Core Idea



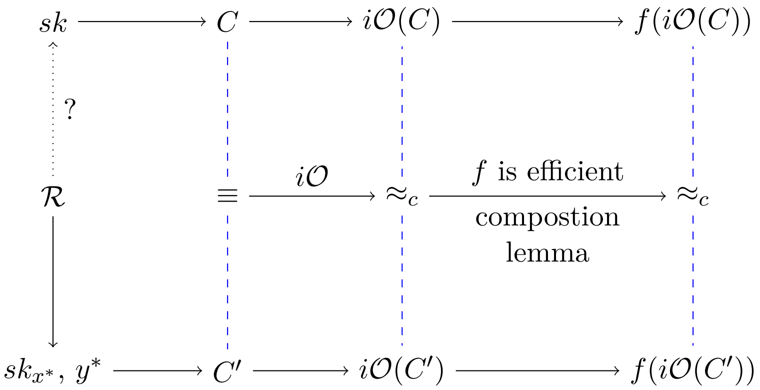
Abstract and Generalize the Core Idea



Abstract and Generalize the Core Idea



Abstract and Generalize the Core Idea



simulate leakage in a computationally indistinguishable manner

Key Observation

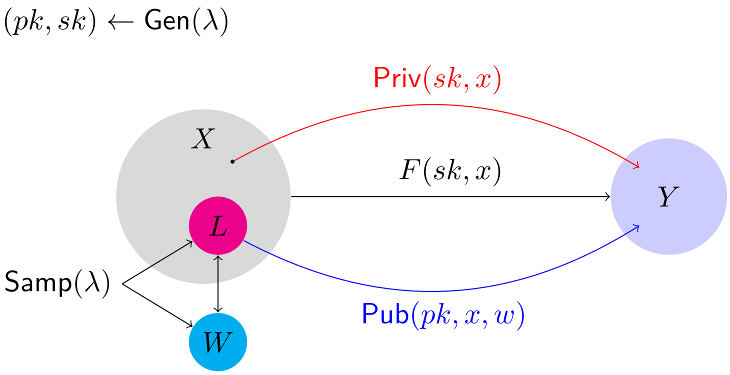
Can we push the idea to extreme?

- Dachman-Soled et al. [DGL+16]: Sahai-Waters KEM can be made LR by setting sk as an obfuscated program
- Chen et al. [CZ14]: the essence of Sahai-Waters KEM – $i\mathcal{O}$ bootstraps Punc-PRF into Punc-"publicly evaluable" PRF

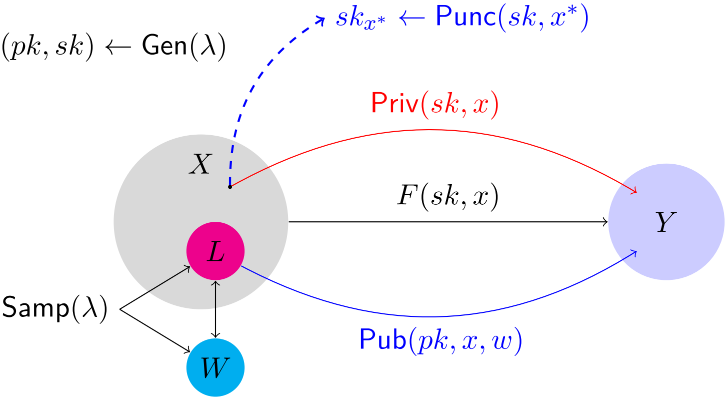
These two results suggest:

$$i\mathcal{O}(\text{Punc-PEPRF}) \rightsquigarrow \text{LR PEPRF}$$

(Puncturable) Publicly Evaluable PRF



(Puncturable) Publicly Evaluable PRF



Security of (Puncturable) Publicly Evaluable PRF

Security of (Puncturable) Publicly Evaluable PRF

$\xleftarrow{\quad pk \quad}$

$(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$

Security of (Puncturable) Publicly Evaluable PRF

$\beta \xleftarrow{R} \{0,1\}$

$\xleftarrow{\quad pk \quad}$

$$(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$$
$$(x^*, w^*) \leftarrow \mathsf{Samp}(\lambda)$$
$$sk_{x^*} \leftarrow \mathsf{Punc}(sk, x^*)$$
$$y_0^* \leftarrow F(sk, x^*)$$
$$y_1^* \xleftarrow{R} Y$$

$\xleftarrow{\quad x^*,\ y_\beta^*,\ sk_{x^*} \quad}$

Security of (Puncturable) Publicly Evaluable PRF

$\beta \xleftarrow{R} \{0,1\}$

$(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$

$(x^*, w^*) \leftarrow \mathsf{Samp}(\lambda)$

$sk_{x^*} \leftarrow \mathsf{Punc}(sk, x^*)$

$y_0^* \leftarrow F(sk, x^*)$

$y_1^* \xleftarrow{R} Y$

$\xleftarrow{\quad pk \quad}$

$\xleftarrow{\quad x^*,\ y_\beta^*,\ sk_{x^*} \quad}$

$\xrightarrow{\quad \beta' \quad}$

Security of (Puncturable) Publicly Evaluable PRF

$\beta =?$

$\beta \xleftarrow{R} \{0,1\}$

$$(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$$
$$(x^*, w^*) \leftarrow \mathsf{Samp}(\lambda)$$
$$sk_{x^*} \leftarrow \mathsf{Punc}(sk, x^*)$$
$$y_0^* \leftarrow F(sk, x^*)$$
$$y_1^* \xleftarrow{R} Y$$

pk ←

$x^*,\ y_\beta^*,\ sk_{x^*}$ ←

β' →

$$|\Pr[\beta = \beta'] - 1/2| \leq \mathsf{negl}(\lambda)$$

Security of (Puncturable) Publicly Evaluable PRF

$\beta \xleftarrow{R} \{0,1\}$

$(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$

$(x^*, w^*) \leftarrow \mathsf{Samp}(\lambda)$

$sk_{x^*} \leftarrow \mathsf{Punc}(sk, x^*)$

$y_0^* \leftarrow F(sk, x^*)$

$y_1^* \xleftarrow{R} Y$

$\xleftarrow{\quad pk \quad}$

$\xrightarrow{\quad f_i \quad}$

$\xleftarrow{\quad f_i(sk) \quad}$

$\xleftarrow{\quad x^*,\ y_\beta^*,\ sk_{x^*} \quad}$

$\xrightarrow{\quad \beta' \quad}$

$|\Pr[\beta = \beta'] - 1/2| \leq \mathsf{negl}(\lambda)$

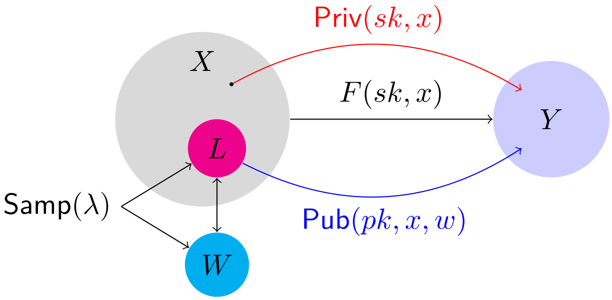
LR-PEPRF from Punc-PEPRF

Idea: Obfuscate-and-Extract

LR-PEPRF from Punc-PEPRF

Idea: Obfuscate-and-Extract

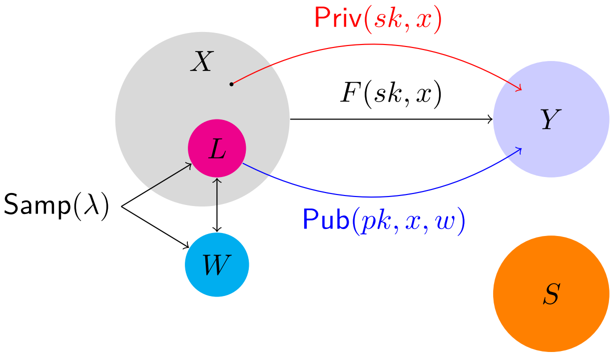
$(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$



LR-PEPRF from Punc-PEPRF

Idea: Obfuscate-and-Extract

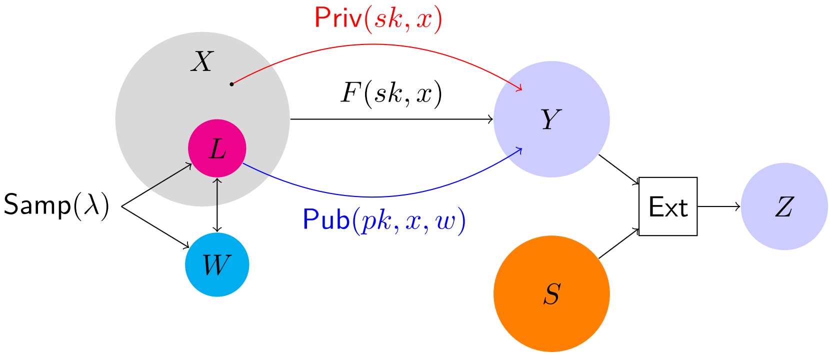
$(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$



LR-PEPRF from Punc-PEPRF

Idea: Obfuscate-and-Extract

$(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$

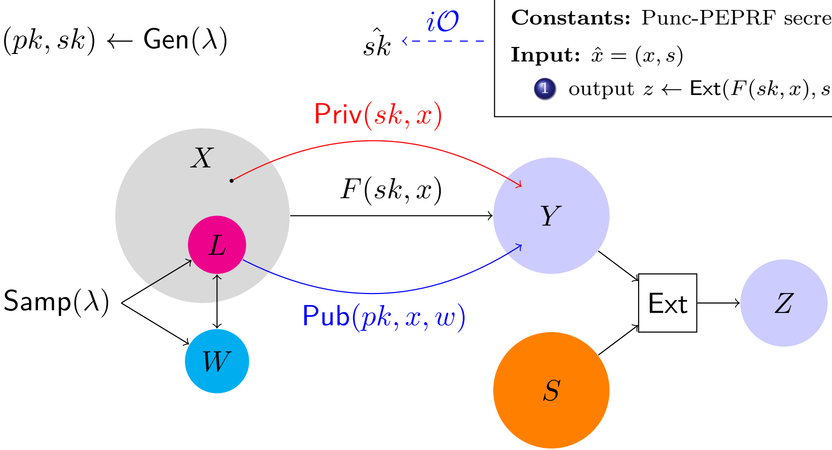


LR PEPRF $\hat{F}$ from $X \times S$ to Z: $\mathsf{Ext}(F(sk, x), s)$

LR-PEPRF from Punc-PEPRF

Idea: Obfuscate-and-Extract

Priv
Constants: Punc-PEPRF secret key sk
Input: $\hat{x} = (x, s)$
1. output $z \leftarrow \mathsf{Ext}(F(sk, x), s)$



LR PEPRF $\hat{F}$ from $X \times S$ to Z: $\mathsf{Ext}(F(sk, x), s)$

Theorem: *The above PEPRF $\hat{F}$ is leakage-resilient under appropriate parameter setting.*

Game 0. (the original game) $\hat{sk} \leftarrow i\mathcal{O}(\mathrm{Priv})$

Game 1. $\hat{sk} \leftarrow i\mathcal{O}(\mathrm{Priv}^*)$, where $y^* \leftarrow F(sk, x^*)$

Priv^*
Constants: Punc-PEPRF punctured key sk_{x^*}, x^* and y^*
Input: $\hat{x} = (x, s)$
1. If $x = x^*$, output $\mathsf{Ext}(y^*, s)$. Else, output $\mathsf{Ext}(F(sk_{x^*}, x), s)$.

Game 2. $y^* \xleftarrow{R} Y$

- $\mathrm{Priv} \equiv \mathrm{Priv}^* + i\mathcal{O} \Rightarrow$ **Game 0** $\approx_c$ **Game 1**
- punc-PEPRF $\Rightarrow$ **Game 1** $\approx_c$ **Game 2**
- randomness extractor $\Rightarrow z^* \leftarrow \mathsf{Ext}(y^*, s^*) \approx_s U_Z$

Constructions of Punc-PEPRF

$i\mathcal{O}$(Punc-PEPRF) $\rightsquigarrow$ LR-PEPRF $\Rightarrow$ LR-KEM

How to construct Punc-PEPRF?

wPPRF+PRG+$i\mathcal{O}$ (a slight modification of SW KEM)

- clarify and encompass Dachman-Soled et al's construction

Punc-TDF $\Leftarrow$ correlated-product TDF [RS09]

- PTDF can be viewed as a special type of adaptive TDF – $\mathcal{O}_{\text{inv}}$ can be instantiated succinctly

Punc-EHPS $\Leftarrow$ derivable EHPS

- "derivable" is a mild property that satisfied by all the known realizations of EHPS [Wee10]

Significance

Matsuda and Hanaoka [MH15]: Punc-KEM – capture a common pattern towards CCA security

- Punc-PEPRF $\Rightarrow$ Punc-KEM with perfect punctured decapsulation soundness

CCA security obtained via punctured road can be converted to Leakage-Resilience in a *non-black-box* manner via $i\mathcal{O}$

- PKE via CP-TDF
- PKE via EHPS

Outline

1. Background
2. Motivation
3. Primitives
4. Our Framework Towards Leakage-Resilience
   - Leakage-Resilient PKE
   - Leakage-Resilient SKE
   - Leakage-Resilient Signature
5. Achieving Optimal Leakage Rate

Extension to the Symmetric Setting

$i\mathcal{O}(\text{weak-Punc-PRF}) \rightsquigarrow \text{LR-weak-PRF} \Rightarrow \text{LR-SKE}$

$(pp, sk) \leftarrow \mathsf{Gen}(\lambda)$

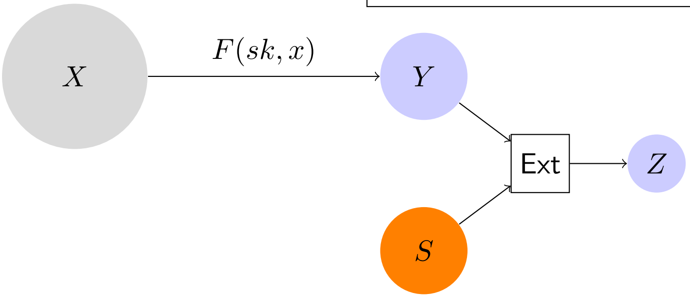
$\hat{sk} \xleftarrow{i\mathcal{O}}$

Priv

Constants: wPPRF secret key sk

Input: $\hat{x} = (x, s)$

1. output $z \leftarrow \mathsf{Ext}(F(sk, x), s)$



LR wPRF $\hat{F}$ from $X \times S$ to Z: $\mathsf{Ext}(F(sk, x), s)$

Outline

Review of Sahai-Waters Signature

Essence of Sahai-Waters Signature: $i\mathcal{O}$ makes PRF-based MAC publicly verifiable

- $\mathsf{Gen}(\lambda)$: pick $k \xleftarrow{R} K$ for sPPRF $F : K \times M \to Y$, pick a OWF $g : Y \to Z$; set $sk \leftarrow k$, $vk \leftarrow i\mathcal{O}(\text{Verify})$.
- $\mathsf{Sign}(sk, m)$: output $\sigma \leftarrow F(k, m)$.
- $\mathsf{Verify}(vk, m, \sigma)$: output $vk(m, \sigma)$.

Verify

Constants: sPPRF key k

Input: message m and signature σ

1. output $g(\sigma) =? g(F(k, m))$.

Proof of Selective Security

Theorem: *Sahai-Waters signature is selectively secure.*

Game 0. (original game) $vk \leftarrow i\mathcal{O}(\text{Verify})$.

Game 1. $vk \leftarrow i\mathcal{O}(\text{Verify}^*)$, here $z^* \leftarrow g(\sigma^*)$, $\sigma^* \leftarrow F(k, m^*)$.

Verify^*
Constants: punctured sPPRF key k_{m^*} and z^*
Input: message m and signature σ
1. If $m = m^*$, output $g(\sigma) =? z^*$.
2. Else, output $g(\sigma) =? g(F(k_{m^*}, m))$.

Game 2. $\sigma^* \leftarrow Y$.

- $\text{Verify} \equiv \text{Verify}^*$ + $i\mathcal{O}$ $\Rightarrow$ **Game 0** $\approx_c$ **Game 1**
- sPPRF $\Rightarrow$ **Game 1** $\approx_c$ **Game 2**
- OWF $\Rightarrow$ σ^* is unpredictable in **Game 2**

How to make Sahai-Waters's signature Leakage-Resilient?

Technical hurdle: how to handle leakage queries?

1. express signing algorithm as a program and obfuscate the program as sk
2. simulate leakage queries with function-equivalent key – an obfuscation of a program build from k_{m^*} and σ^*

How to make Sahai-Waters's signature Leakage-Resilient?

Technical hurdle: how to handle leakage queries?

1. express signing algorithm as a program and obfuscate the program as sk
2. simulate leakage queries with function-equivalent key – an obfuscation of a program build from k_{m^*} and σ^*

Problems

- Construction perspective: leakage queries leak the information of σ^* (the preimage of z^*) $\Rightarrow$ unable to reduce unforgeability to one-wayness of g
- Proof perspective: $\mathcal{R}$ does not know σ^*

How to make Sahai-Waters's signature Leakage-Resilient?

Technical hurdle: how to handle leakage queries?

1. express signing algorithm as a program and obfuscate the program as sk
2. simulate leakage queries with function-equivalent key – an obfuscation of a program build from k_{m^*} and σ^*

Problems

- Construction perspective: leakage queries leak the information of σ^* (the preimage of z^*) $\Rightarrow$ unable to reduce unforgeability to one-wayness of g
- Proof perspective: $\mathcal{R}$ does not know σ^*

Our solution: using LR OWF instead of standard OWF

- In the final security game, $\mathcal{R}$ can translate leakage queries on secret key to those on σ^*.

LR OWF + sPPRF + $i\mathcal{O} \Rightarrow$ deterministic LR SIG (selective)

How to achieve adaptive security?

- Using Extremely Lossy Function [Zha16] hash the message before signing: deterministic but relying on exponential hardness assumption
- Applying "prefix-guessing technique" [RW14]: randomized but public-coin

So far the best solution to the open problem posed by Boyle et al. [BSW11] (Eurocrypt' 11)

Outline

How to achieve optimal leakage rate?

The leakage rate of our basic constructions is low

- secret key is an obfuscated program $\rightsquigarrow$ large size
- the maximum leakage amount $\leq \log_2 |Y|$

Can we achieve optimal leakage rate?

Dachman-Soled et al.'s Approach

Secret key – a secret obfuscated program (like a gun that must be kept secretly)

Dachman-Soled et al.'s Approach

Secret key – a secret obfuscated program (like a gun that must be kept secretly)

Decompose the secret obfuscated program

- make the logic part public
- set a trigger device inside the public program and use trigger as the secret key

The Case of LR-PEPRF from Punc-PEPRF

Priv

Constants: Punc-PEPRF secret key sk

Input: $\hat{x} = (x, s)$

1. Output $z \leftarrow \mathsf{Ext}(F(sk, x), s)$

Modification: $ct^* \leftarrow \mathsf{Enc}(k_e, 0^n)$, $n = \log |Y|$; pick a CRHF h, set $h(ct^*) = t^*$

ct^* is set as secret key, obfuscated program is made public.

Priv

Constants: Punc-PEPRF secret key sk, t^*

Input: ct, $\hat{x} = (x, s)$

1. If $h(ct) \neq t^*$, output $\perp$. Else, output $z \leftarrow \mathsf{Ext}(F(sk, x), s)$.

greatly shrink the size of secret key: an obfuscated program $\rightsquigarrow$ a ciphertext

Security Proof

Game 0. $C_{\text{eval}} \leftarrow i\mathcal{O}(\text{Priv})$ as part of pk, $ct^* \leftarrow \text{SKE}.\mathsf{Enc}(k_e, 0^n)$ as sk.

Game 1. $ct^* \leftarrow \text{SKE}.\mathsf{Enc}(k_e, y^*)$, where $y^* \leftarrow F(sk, x^*)$

Game 2. $C_{\text{eval}} \leftarrow i\mathcal{O}(\text{Priv}^*)$

Game 3. $y^* \xleftarrow{\text{R}} Y$

Priv*
Constants: Punc-PEPRF punctured secret key sk_{x^*}, k_e, t^*
Input: ct, $\hat{x} = (x, s)$
1. If $h(ct) \neq t^*$, output $\perp$.
2. Else if $x = x^*$, set $y^* \leftarrow \text{SKE}.\mathsf{Dec}(k_e, ct)$, output $z \leftarrow \mathsf{Ext}(y^*, s)$.
3. Otherwise, output $z \leftarrow \mathsf{Ext}(F(sk, x), s)$.

$$|t^*| + \ell \leq |Y|,\ |Y| \leq |ct^*| \text{ and } \rho = \ell / |ct^*|$$

Analysis

To achieve optimal leakage rate

- h must be compressing to decrease $|t^*|$, otherwise t^* (hardwired in public program) will reveal too much information of $y^* \leftarrow F(sk, x^*)$
- The choice may make the programs in **Game 1** and **Game 2** have differing-inputs

a collision: $ct' \neq ct^*$ but $h(ct') = t^* = h(ct^*)$ where ct' decrypts to $y' \neq y^*$

$\rightsquigarrow$ one have to resort to differing-input obfuscation, which is highly suspicious.

Our Technique

Idea: replace CRHF with lossy function

- Injective mode: ensure Priv and Priv* are equivalent $\rightsquigarrow$ safely use $i\mathcal{O}$
- Lossy mode: switch to lossy mode to greatly reduce $|t^*|$ $\rightsquigarrow$ t^* only leaks very little information of y^*,

By appropriate parameter choice, $\rho = 1 - o(1)$

This settles the open problem posed by Dachman-Soled et al. [DGL$^+$16]: achieving optimal leakage ratio without resorting to $di\mathcal{O}$

This trick might be instructive elsewhere for avoiding differing-input obfuscation

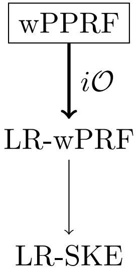
Conclusion

We develop a framework for building leakage-resilient cryptography in BLM from punc-primitives and $i\mathcal{O}$.

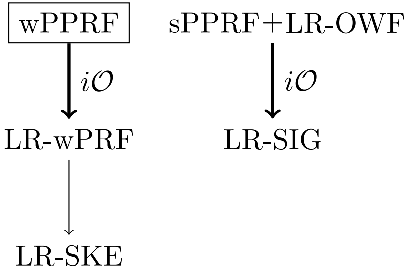
Major insight: various punc-PRFs can achieve LR on an obfuscated street

1. wPPRF+$i\mathcal{O}$ $\rightsquigarrow$ LR wPRF $\Rightarrow$ LR-SKE
2. punc-PEPRF+$i\mathcal{O}$ $\rightsquigarrow$ LR PEPRF $\Rightarrow$ LR-PKE
    - as a building block of independent interest, we realize punc-PEPRF from newly introduced punc-objects such as PTDFs and PEHPS.
3. sPPRF+ LR-OWF + $i\mathcal{O}$ $\Rightarrow$ the first LR-public-coin Sig
    - solve the open problem posed by Boyle et al. (Eurocrypt 2011)
4. By further assuming lossy functions, all the above constructions achieve optimal leakage rate – not known to be achievable for wPRF, PEPRF and public-coin Sig before.
    - solve the open problem posed by Dachman-Soled et al. (PKC 2016, JOC 2018)

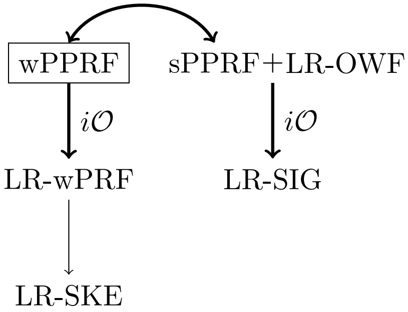
Conclusion



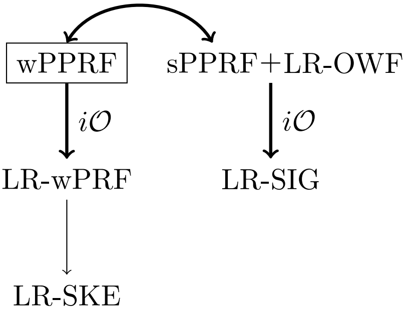
Conclusion



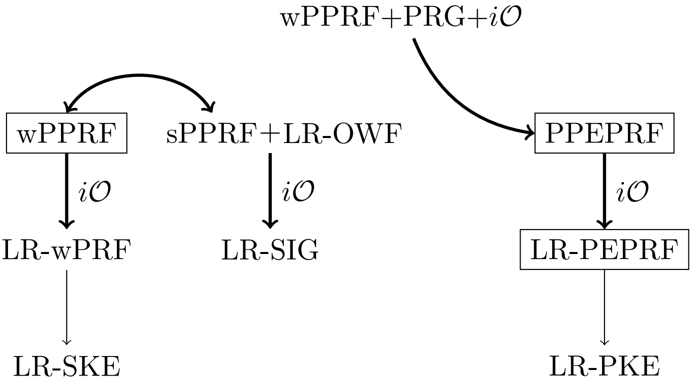
Conclusion



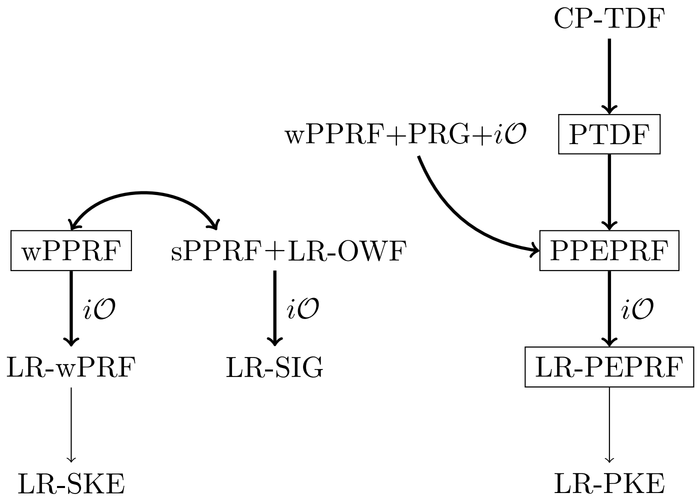
Conclusion


wPPRF
sPPRF+LR-OWF
PPEPRF
$i\mathcal{O}$
$i\mathcal{O}$
$i\mathcal{O}$
LR-wPRF
LR-SIG
LR-PEPRF
LR-SKE
LR-PKE

Conclusion

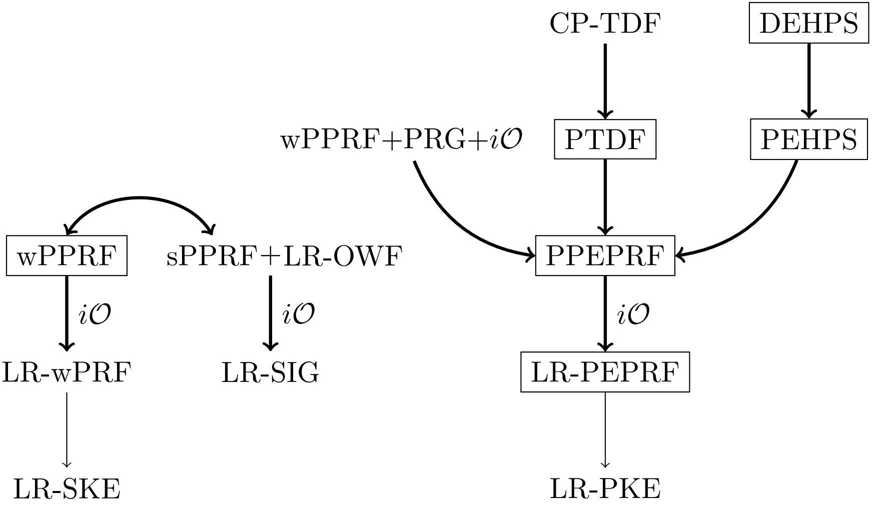

wPPRF+PRG+$i\mathcal{O}$
wPPRF
sPPRF+LR-OWF
PPEPRF
$i\mathcal{O}$
$i\mathcal{O}$
$i\mathcal{O}$
LR-wPRF
LR-SIG
LR-PEPRF
LR-SKE
LR-PKE

Conclusion



CP-TDF
wPPRF+PRG+$i\mathcal{O}$
PTDF
wPPRF
sPPRF+LR-OWF
PPEPRF
$i\mathcal{O}$
$i\mathcal{O}$
$i\mathcal{O}$
LR-wPRF
LR-SIG
LR-PEPRF
LR-SKE
LR-PKE

Conclusion


CP-TDF
DEHPS
wPPRF+PRG+$i\mathcal{O}$
PTDF
PEHPS
wPPRF
sPPRF+LR-OWF
PPEPRF
$i\mathcal{O}$
$i\mathcal{O}$
$i\mathcal{O}$
LR-wPRF
LR-SIG
LR-PEPRF
LR-SKE
LR-PKE

Thanks for Your Attention!

Any Questions?

https://eprint.iacr.org/2018/781

Reference I

[ADN+10] Joël Alwen, Yevgeniy Dodis, Moni Naor, Gil Segev, Shabsi Walfish, and Daniel Wichs. Public-key encryption in the bounded-retrieval model. In *EUROCRYPT*, pages 113–134, 2010.

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In *CRYPTO*, pages 36–54, 2009.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, pages 474–495, 2009.

[BG10] Zvika Brakerski and Shafi Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In *CRYPTO*, pages 1–20, 2010.

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *J. ACM*, 59(2):6, 2012.

[BKKV10] Zvika Brakerski, Yael Tauman Kalai, Jonathan Katz, and Vinod Vaikuntanathan. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *FOCS*, pages 501–510, 2010.

[BSW11] Elette Boyle, Gil Segev, and Daniel Wichs. Fully leakage-resilient signatures. In *EUROCRYPT*, pages 89–108, 2011.

[CQX18] Yu Chen, Baodong Qin, and Haiyang Xue. Regularly lossy functions and their applications. In *CT-RSA*, 2018.

Reference II

[CZ14] Yu Chen and Zongyang Zhang. Publicly evaluable pseudorandom functions and their applications. In *SCN*, pages 115–134, 2014.

[DGK+10] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC*, pages 361–381, 2010.

[DGL+16] Dana Dachman-Soled, S. Dov Gordon, Feng-Hao Liu, Adam O'Neill, and Hong-Sheng Zhou. Leakage-resilient public-key encryption from obfuscation. In *PKC*, pages 101–128, 2016.

[DHLW10] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt, and Daniel Wichs. Efficient public-key cryptography in the presence of key leakage. In *ASIACRYPT*, pages 613–631, 2010.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In *STOC*, pages 621–630, 2009.

[DY13] Yevgeniy Dodis and Yu Yu. Overcoming weak expectations. In *TCC*, pages 1–22, 2013.

[GJS11] Sanjam Garg, Abhishek Jain, and Amit Sahai. Leakage-resilient zero knowledge. In *CRYPTO*, pages 297–315, 2011.

[HLWW13] Carmit Hazay, López-Alt, Hoeteck Wee, and Daniel Wichs. Leakage-resilient cryptography from minimal assumptions. In *EUROCRYPT*, pages 160–176, 2013.

[KV09] Jonathan Katz and Vinod Vaikuntanathan. Signature schemes with bounded leakage resilience. In *ASIACRYPT*, pages 703–720, 2009.

Reference III

[LLW11] Allison B. Lewko, Mark Lewko, and Brent Waters. How to leak on key updates. In *STOC*, pages 725–734, 2011.

[MH15] Takahiro Matsuda and Goichiro Hanaoka. Constructing and understanding chosen ciphertext security via puncturable key encapsulation mechanisms. In *TCC*, pages 561–590, 2015.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In *TCC*, pages 278–296, 2004.

[MTVY11] Tal Malkin, Isamu Teranishi, Yevgeniy Vahlis, and Moti Yung. Signatures resilient to continual leakage on memory and computation. In *TCC*, pages 89–106, 2011.

[NS09] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In *CRYPTO*, pages 18–35, 2009.

[Pie09] Krzysztof Pietrzak. A leakage-resilient mode of operation. In *Advances in Cryptology - EUROCRYPT 2009*, pages 462–482, 2009.

[QL13] Baodong Qin and Shengli Liu. Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter. In *ASIACRYPT*, pages 381–400, 2013.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *TCC*, pages 419–436, 2009.

[RW14] Kim Ramchen and Brent Waters. Fully secure and fast signing from obfuscation. In *ACM CCS*, pages 659–673, 2014.

Reference IV

[SW14] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC*, pages 475–484, 2014.

[Wee10] Hoeteck Wee. Efficient chosen-ciphertext security via extractable hash proofs. In *CRYPTO*, pages 314–332, 2010.

[Zha16] Mark Zhandry. The Magic of ELFs. In *CRYPTO*, pages 479–508, 2016.