RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: CRYP-F03

PRACTICAL, ANONYMOUS, AND PUBLICLY LINKABLE UNIVERSALLY-COMPOSABLE REPUTATION SYSTEMS

Jakob Juhnke

PhD Student/Research Assistant
Paderborn University
Paderborn, Germany

Why do we need reputation systems?

Why do we need reputation systems?

Customers and product providers want to get valuable information about previous transactions:

Why do we need reputation systems?

Customers and product providers want to get valuable information about previous transactions:

- Customers:
  - What kind of experiences have other people had with this product?
  - Does the product cover my needs?
  - Is the product worth its price?

Why do we need reputation systems?

Customers and product providers want to get valuable information about previous transactions:

- Customers:
  - What kind of experiences have other people had with this product?
  - Does the product cover my needs?
  - Is the product worth its price?
- Providers:
  - How can I improve my products to fit the customer's needs?
  - Are my products easy to use?
  - Are products from other providers better/worse than mine and why?

Desirable properties

We want trustworthy, reliable and honest ratings:

Desirable properties

We want trustworthy, reliable and honest ratings:

- Customers want:
  - → Raters stay anonymous
  - → Verifiable binding of transactions and ratings
  - → Everyone is able to detect multiple ratings from the same user
  - → Providers are not able to rate their own products

Desirable properties

We want trustworthy, reliable and honest ratings:

- Customers want:
  - → Raters stay anonymous
  - → Verifiable binding of transactions and ratings
  - → Everyone is able to detect multiple ratings from the same user
  - → Providers are not able to rate their own products

- Providers want:
  - → Raters allowed to rate purchased products only once
  - → In case of misuse: anonymity of raters should be rescinded
  - → Ratings should be arbitrary strings to support arbitrary higher-level protocols

Our starting point

Our starting point

What we already have:

[BJK15] Blömer, Juhnke, Kolb: Anonymous and Publicly Linkable Reputation Systems, Financial Cryptography and Data Security (FC) 2015:

Our starting point

What we already have:

[BJK15] Blömer, Juhnke, Kolb: Anonymous and Publicly Linkable Reputation Systems, Financial Cryptography and Data Security (FC) 2015:

- Definition of an abstract model for reputation systems

Our starting point

What we already have:

[BJK15] Blömer, Juhnke, Kolb: Anonymous and Publicly Linkable Reputation Systems, Financial Cryptography and Data Security (FC) 2015:

- Definition of an abstract model for reputation systems
- Experiment-based security definitions for anonymity, public linkability, traceability, and strong-exculpability

Our starting point

What we already have:

[BJK15] Blömer, Juhnke, Kolb: Anonymous and Publicly Linkable Reputation Systems, Financial Cryptography and Data Security (FC) 2015:

- Definition of an abstract model for reputation systems
- Experiment-based security definitions for anonymity, public linkability, traceability, and strong-exculpability
- Concrete scheme based on group signatures

Our starting point

What we already have:

[BJK15] Blömer, Juhnke, Kolb: Anonymous and Publicly Linkable Reputation Systems, Financial Cryptography and Data Security (FC) 2015:

- Definition of an abstract model for reputation systems
- Experiment-based security definitions for anonymity, public linkability, traceability, and strong-exculpability
- Concrete scheme based on group signatures
- Drawbacks: Security definitions use 11 oracles (plus 3 random oracles)
  Customers and providers are distinct sets

Our Goals

Our Goals

What we want:

- Remove the strict separation of customers and providers
  → Treat them as different roles of the same party

Our Goals

What we want:

- Remove the strict separation of customers and providers
  - → Treat them as different roles of the same party
- “Simplify” security definitions
  - → Circumvent oracles
  - → Combine the different security experiments

Our Goals

What we want:

- Remove the strict separation of customers and providers
  - → Treat them as different roles of the same party
- “Simplify” security definitions
  - → Circumvent oracles
  - → Combine the different security experiments
- Cover additional security properties

Our Goals

What we want:

- Remove the strict separation of customers and providers
  - → Treat them as different roles of the same party
- "Simplify" security definitions
  - → Circumvent oracles
  - → Combine the different security experiments
- Cover additional security properties

→ **Formulate an ideal functionality for reputation systems**

Our Goals

What we want:

- Remove the strict separation of customers and providers
  - → Treat them as different roles of the same party
- "Simplify" security definitions
  - → Circumvent oracles
  - → Combine the different security experiments
- Cover additional security properties

→ **Formulate an ideal functionality for reputation systems**

Our result: a holistic functionality that also covers initially unexpected security properties

Why Universal Composability?

Why Universal Composability?

- Reputation systems are often integrated into other applications
- Experiment-based security definitions only guarantee stand-alone security
- We exclude concrete reputation functions as they are application specific

Why Universal Composability?

- Reputation systems are often integrated into other applications
- Experiment-based security definitions only guarantee stand-alone security
- We exclude concrete reputation functions as they are application specific

  But:
  - we want a system that can be combined with every reputation function

Why Universal Composability?

- Reputation systems are often integrated into other applications
- Experiment-based security definitions only guarantee stand-alone security
- We exclude concrete reputation functions as they are application specific

  But:
  - we want a system that can be combined with every reputation function
  - possible to incorporate into every application

Why Universal Composability?

- Reputation systems are often integrated into other applications
- Experiment-based security definitions only guarantee stand-alone security
- We exclude concrete reputation functions as they are application specific

  But:
  - we want a system that can be combined with every reputation function
  - possible to incorporate into every application

⇒ Composition of protocols is explicitly desired
⇒ Universal Composability is the framework of our choice (Canetti, 2001)

The Basic Model - 1/2

Our model consists of one Identity Manager and a group of users with different roles:

The Basic Model - 1/2

Our model consists of one Identity Manager and a group of users with different roles:

- Identity Manager:
  - User Registration, avoiding double-registration
  - De-Anonymization in case of misbehaving users

The Basic Model - 1/2

Our model consists of one Identity Manager and a group of users with different roles:

- Identity Manager:
  - User Registration, avoiding double-registration
  - De-Anonymization in case of misbehaving users
- User acting as Seller/Product Owner:
  - Publish product-specific keys
  - hand Rating-Tokens to designated Raters

The Basic Model - 1/2

Our model consists of one Identity Manager and a group of users with different roles:

- Identity Manager:
  - User Registration, avoiding double-registration
  - De-Anonymization in case of misbehaving users
- User acting as Seller/Product Owner:
  - Publish product-specific keys
  - hand Rating-Tokens to designated Raters
- User acting as Rater:
  - use registration information and rating token to rate a specific product

The Basic Model - 1/2

Our model consists of one Identity Manager and a group of users with different roles:

- Identity Manager:
  - User Registration, avoiding double-registration
  - De-Anonymization in case of misbehaving users
- User acting as Seller/Product Owner:
  - Publish product-specific keys
  - hand Rating-Tokens to designated Raters
- User acting as Rater:
  - use registration information and rating token to rate a specific product

Seller

Rater

Outsider

The Basic Model - 2/2

The Basic Model - 2/2

1. $pp \leftarrow$ KeyGen

The Basic Model - 2/2

1. $pp \leftarrow \text{KeyGen}$

2. $ppk \leftarrow \text{NewProduct}(P_i, prod)$

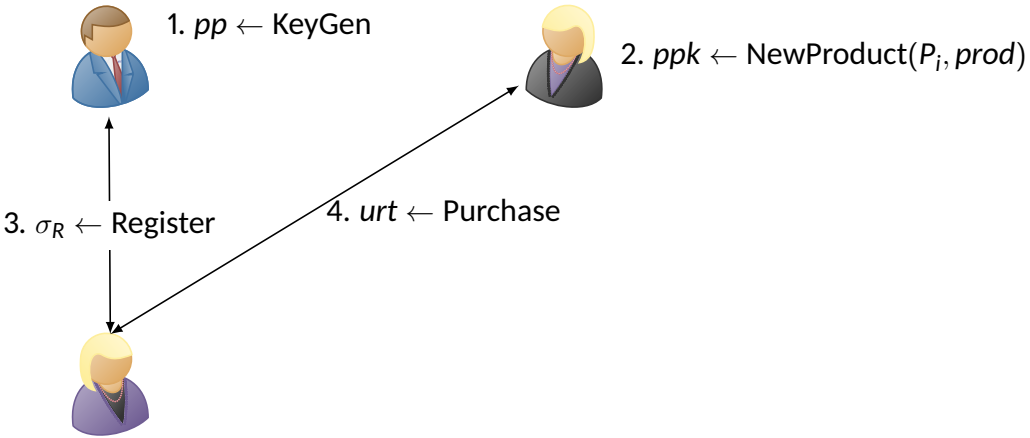
The Basic Model - 2/2

1. $pp \leftarrow$ KeyGen

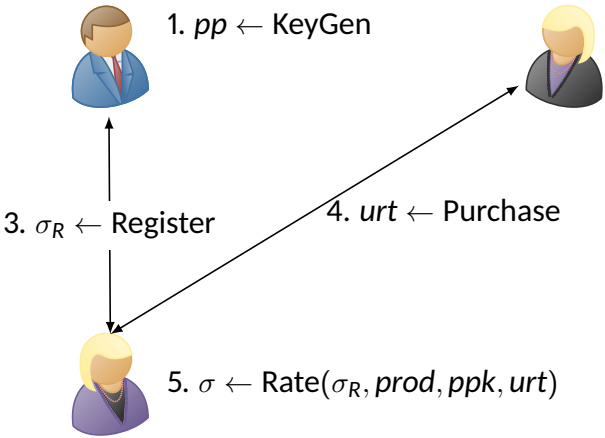
2. $ppk \leftarrow$ NewProduct$(P_i, prod)$

3. $\sigma_R \leftarrow$ Register

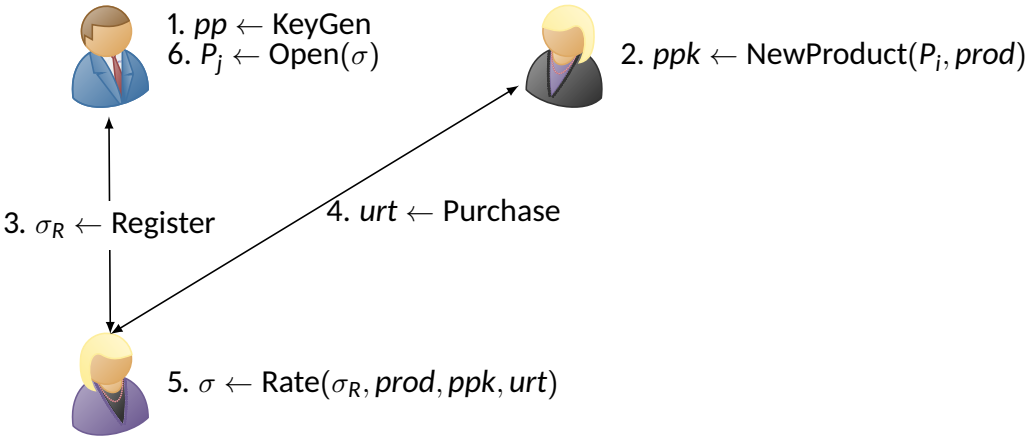
The Basic Model - 2/2



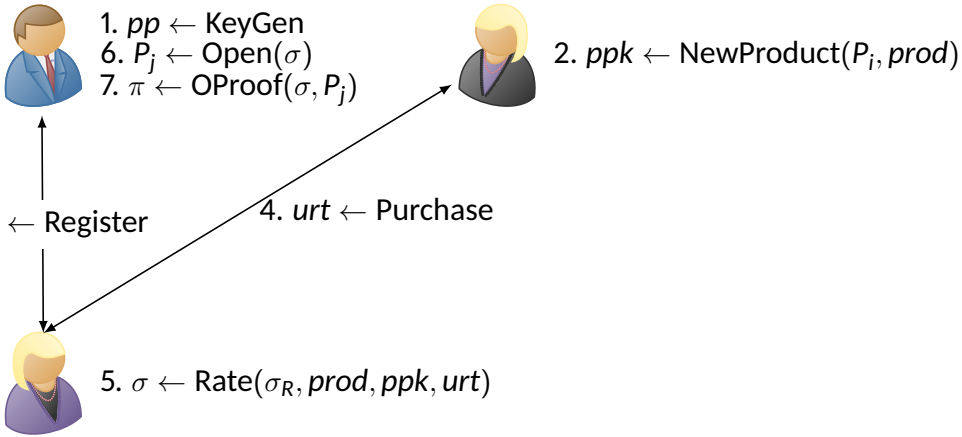
The Basic Model - 2/2



The Basic Model - 2/2



The Basic Model - 2/2



The Basic Model - 2/2

1. $pp \leftarrow \text{KeyGen}$
6. $P_j \leftarrow \text{Open}(\sigma)$
7. $\pi \leftarrow \text{OProof}(\sigma, P_j)$

2. $ppk \leftarrow \text{NewProduct}(P_i, prod)$

3. $\sigma_R \leftarrow \text{Register}$

4. $urt \leftarrow \text{Purchase}$

5. $\sigma \leftarrow \text{Rate}(\sigma_R, prod, ppk, urt)$

The Basic Model - 2/2

1. $pp \leftarrow \text{KeyGen}$
6. $P_j \leftarrow \text{Open}(\sigma)$
7. $\pi \leftarrow \text{OProof}(\sigma, P_j)$

2. $ppk \leftarrow \text{NewProduct}(P_i, prod)$

3. $\sigma_R \leftarrow \text{Register}$

4. $urt \leftarrow \text{Purchase}$

5. $\sigma \leftarrow \text{Rate}(\sigma_R, prod, ppk, urt)$

$f \leftarrow \text{Verify}(prod, ppk, P_i, \sigma)$
$b \leftarrow \text{Link}(prod, ppk, P_i, \sigma_1, \sigma_2)$

The Formal Model

We define the ideal functionality $\mathcal{F}_{\text{RS}}$ in the UC-Framework (too complex to discuss here in detail) which guarantees:

The Formal Model

We define the ideal functionality $\mathcal{F}_{\text{RS}}$ in the UC-Framework (too complex to discuss here in detail) which guarantees:

- Unforgeability of product public keys

The Formal Model

We define the ideal functionality $\mathcal{F}_{\text{RS}}$ in the UC-Framework (too complex to discuss here in detail) which guarantees:

- Unforgeability of product public keys
- Valid ratings guarantee
  - Anonymity for the rater
  - No Self-Ratings: the rater is not the owner of the product
  - Linkability: multiple ratings from the same user for the same product can be detected (even after multiple purchases)
  - Traceability: the rater is registered and can be identified

The Formal Model

We define the ideal functionality $\mathcal{F}_{\text{RS}}$ in the UC-Framework (too complex to discuss here in detail) which guarantees:

- Unforgeability of product public keys
- Valid ratings guarantee
  - Anonymity for the rater
  - No Self-Ratings: the rater is not the owner of the product
  - Linkability: multiple ratings from the same user for the same product can be detected (even after multiple purchases)
  - Traceability: the rater is registered and can be identified
- Non-Frameability: no user can be accused being the author of a given rating (when the user is not the author)

Realization of $\mathcal{F}_{\mathsf{RS}}$

We use the following techniques for our realization:

Realization of $\mathcal{F}_{\text{RS}}$

We use the following techniques for our realization:

- Interactive ZK-proofs and digital signatures on committed values for Register and Purchase

Realization of $\mathcal{F}_{RS}$

We use the following techniques for our realization:

- Interactive ZK-proofs and digital signatures on committed values for Register and Purchase
- Non-Interactive ZK-proofs and digital signatures for NewProduct

Realization of $\mathcal{F}_{RS}$

We use the following techniques for our realization:

- Interactive ZK-proofs and digital signatures on committed values for Register and Purchase
- Non-Interactive ZK-proofs and digital signatures for NewProduct
- Signatures of Knowledge for Rate (inspired by dynamic group signatures):

Realization of $\mathcal{F}_{RS}$

We use the following techniques for our realization:

- Interactive ZK-proofs and digital signatures on committed values for Register and Purchase
- Non-Interactive ZK-proofs and digital signatures for NewProduct
- Signatures of Knowledge for Rate (inspired by dynamic group signatures):
  - proofs knowledge of registration and rating token
  - includes a unique element (based on rater's identity) for linkability
  - can be "opened" by Identity Manager to obtain rater's identity

Realization of $\mathcal{F}_{RS}$

We use the following techniques for our realization:

- Interactive ZK-proofs and digital signatures on committed values for Register and Purchase
- Non-Interactive ZK-proofs and digital signatures for NewProduct
- Signatures of Knowledge for Rate (inspired by dynamic group signatures):
  - proofs knowledge of registration and rating token
  - includes a unique element (based on rater's identity) for linkability
  - can be "opened" by Identity Manager to obtain rater's identity
- Non-Interactive ZK-proofs and public-key encryption for Open and OProof

Realization of $\mathcal{F}_{RS}$

We use the following techniques for our realization:

- Interactive ZK-proofs and digital signatures on committed values for Register and Purchase
- Non-Interactive ZK-proofs and digital signatures for NewProduct
- Signatures of Knowledge for Rate (inspired by dynamic group signatures):
  - proofs knowledge of registration and rating token
  - includes a unique element (based on rater's identity) for linkability
  - can be "opened" by Identity Manager to obtain rater's identity
- Non-Interactive ZK-proofs and public-key encryption for Open and OProof

Finally, we prove that our protocol UC-realizes the functionality $\mathcal{F}_{RS}$ (is UC-secure).

Future Work

Our plan for future research includes:

Future Work

Our plan for future research includes:

- Remove the trusted Identity Manager

Future Work

Our plan for future research includes:

- Remove the trusted Identity Manager
- Add attributes to ratings to make them more expressive

Future Work

Our plan for future research includes:

- Remove the trusted Identity Manager
- Add attributes to ratings to make them more expressive
- Realization that is secure against adaptive adversaries

Future Work

Our plan for future research includes:

- Remove the trusted Identity Manager
- Add attributes to ratings to make them more expressive
- Realization that is secure against adaptive adversaries
- Integrate revocation into the system

Future Work

Our plan for future research includes:

- Remove the trusted Identity Manager
- Add attributes to ratings to make them more expressive
- Realization that is secure against adaptive adversaries
- Integrate revocation into the system

Thank you very much for your attention and feel free to ask questions!

RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: CRYP-F03

REGULARLY LOSSY FUNCTIONS AND APPLICATIONS IN LEAKAGE-RESILIENT CRYPTOGRAPHY

Yu Chen

Associate Professor
Institute of Information Engineering, Chinese Academy of Sciences

Regularly Lossy Functions and Applications in Leakage-Resilient Cryptography

Yu Chen[1] Baodong Qin[2] Haiyang Xue[1]

[1]SKLOIS, IIE, Chinese Academy of Sciences
[2]Xi'an University of Posts and Telecommunication

CT-RSA 2018

April 15, 2018

Outline

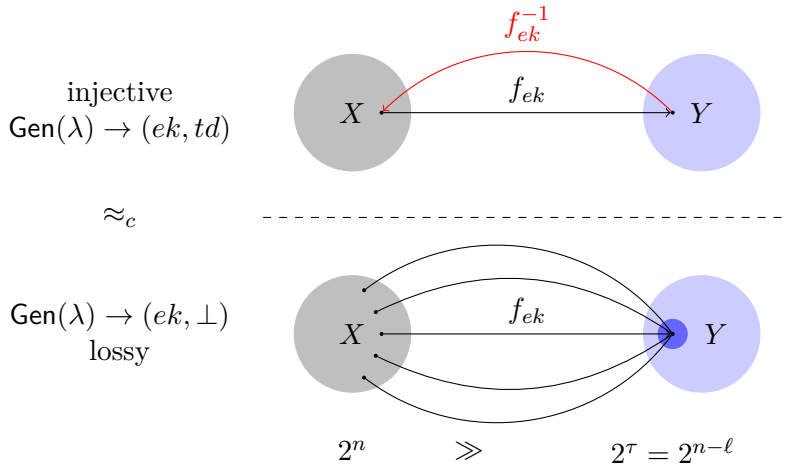
Outline

Lossy Trapdoor Functions

Lossy object *indistinguishable* from original

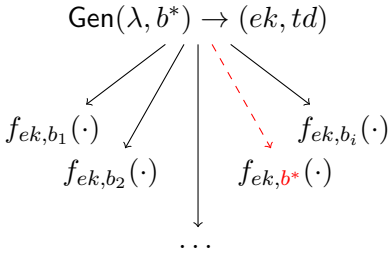
STOC 2008 Peikert and Waters: Lossy Trapdoor Functions and Their Applications

Lossy TDFs



Extension of LTFs: ABO LTFs

- $\mathsf{Gen}(\lambda, b^*)$ has extra input: branch $b^* \in B$.



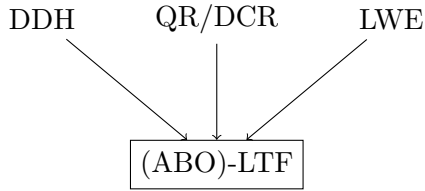
b^* is hidden from ek

$$f_{ek,b}(\cdot) = \begin{cases} \text{lossy} & b = b^* \\ \text{injective and invertible} & b \neq b^* \end{cases}$$

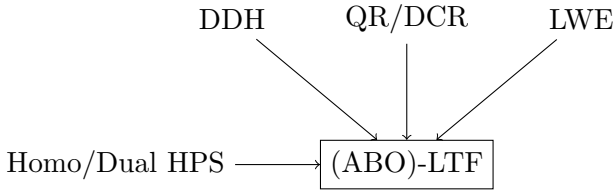
LTFs $\Leftrightarrow$ ABO LTFs

(ABO)-LTF

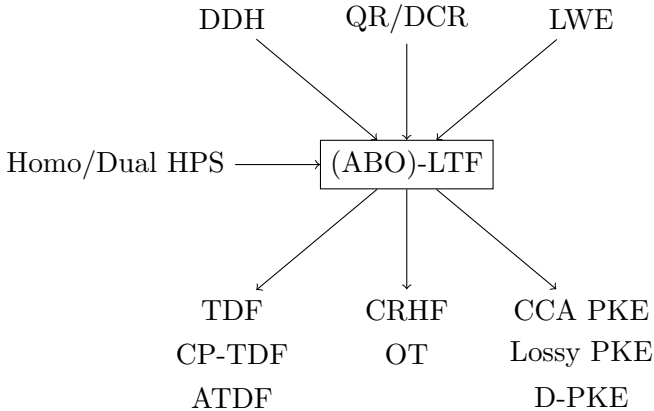
Constructions and Applications



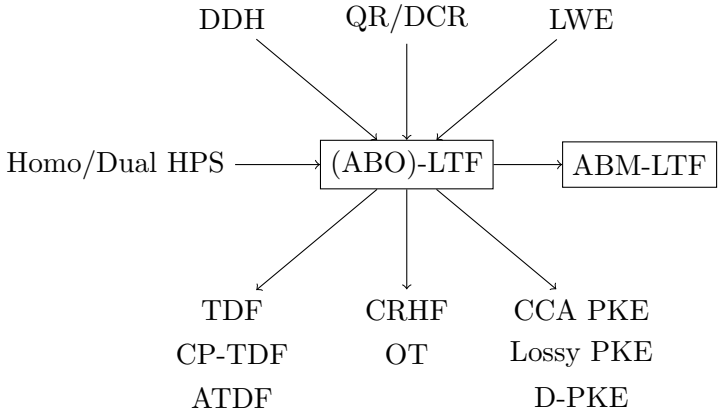
Constructions and Applications



Constructions and Applications



Constructions and Applications



Motivations

In all applications of LTF:

- normal mode: injective+trapdoor fulfill functionality
- lossy mode: establish security

However, the full power of LTF is

- expensive: large key size/high computation cost
- overkill: some applications (e.g., injective OWF, CRHF) do not require a trapdoor, but only normal $\approx_c$ lossy

A central goal in cryptography is to base cryptosystems on primtives that are as weak as possible.

- Peikert and Waters conjectured "the weaker notion LF could be achieved more simply and efficiently than LTF".
- They left the investigation of this question as an interesting problem.

We are motivated to consider the following problems:

How to realize LF efficiently?
Are there any other applications of LF?
Can we further weaken the notion of LF?

Outline

A Simple But Important Observation

When trapdoor is not required for normal mode, the injective property may also be unnecessary.

This observation leads to our further relaxation of LFs

Regularly Lossy Functions

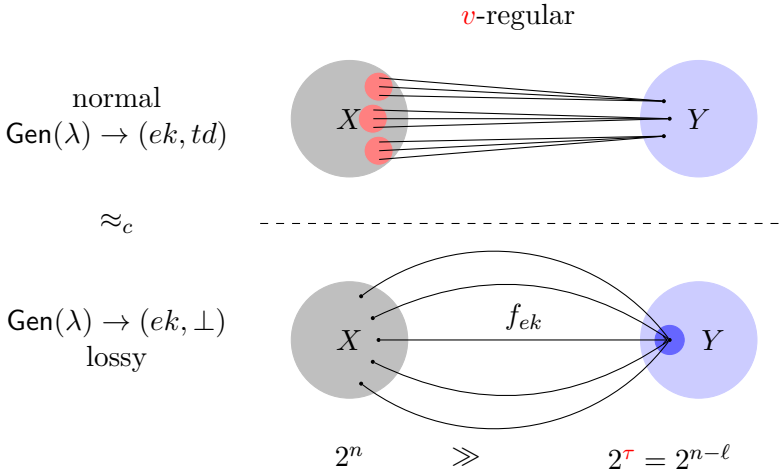
Intuition: the output preserves much *min-entropy* of input

- In RLFs, functions of normal mode could also be lossy, but has to lose in a regular manner.

Definition 1

f is v-to-1 (or v-regular) if $\max_y |f^{-1}(y)| \leq v$.

Regularly Lossy Functions



- When $v = 1$, RLFs specialize to standard LFs

Remarks

Why we have to choose regularity but not image size to capture normal mode?

Remarks

Why we have to choose regularity but not image size to capture normal mode?

- image size is a *global* characterization, which only suffices to give the lower bound of $\tilde{\mathsf{H}}_\infty(x|f(x))$ by the chain rule.

Remarks

Why we have to choose regularity but not image size to capture normal mode?

- image size is a *global* characterization, which only suffices to give the lower bound of $\tilde{\mathsf{H}}_\infty(x|f(x))$ by the chain rule.
- In contrast, regularity is a *local* characterization, which suffices to give the lower bound of $\mathsf{H}_\infty(f(x))$.

Remarks

Why we have to choose regularity but not image size to capture normal mode?

- image size is a *global* characterization, which only suffices to give the lower bound of $\tilde{\mathsf{H}}_\infty(x|f(x))$ by the chain rule.
- In contrast, regularity is a *local* characterization, which suffices to give the lower bound of $\mathsf{H}_\infty(f(x))$.

The following lemma establishes the relation between the min-entropy of x and $f(x)$:

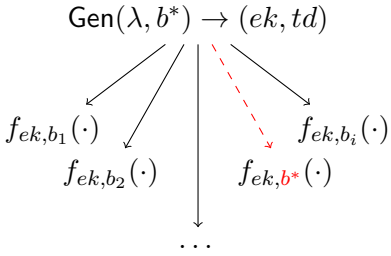
Lemma 2

Let f be a v-to-1 function and x be a random variable over the domain, we have:

$$\mathsf{H}_\infty(f(x)) \geq \mathsf{H}_\infty(x) - \log v$$

All-But-One Regularly Lossy Functions

- $\mathsf{Gen}(\lambda, b^*)$ has extra input: branch $b^* \in B$.



b^* is hidden from ek

$$f_{ek,b}(\cdot) = \begin{cases} \text{lossy} & b = b^* \\ \text{regularly lossy} & b \neq b^* \end{cases}$$

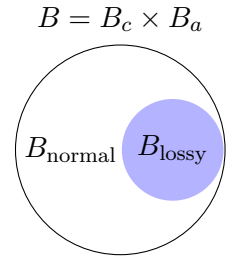
$$\text{RLF} \Leftrightarrow \text{ABO-RLF}$$

One-Time Regularly Lossy Filters

$$\mathsf{Gen}(\lambda) \to (ek, td)$$

$$\downarrow$$

$$f_{ek, b=(b_c, b_a)}(\cdot)$$



$$\mathsf{SampLossy}(td, b_a) \to b_c \text{ s.t. } (b_c, b_a) \in B_{\text{lossy}}$$

- Indistinguishability: $\forall b_a \in B_a$, we have:

$$b_c \leftarrow \mathsf{SampLossy}(td, b_a) \approx_c b_c \xleftarrow{\mathrm{R}} B_c$$

- Evasiveness: it is hard to generate a new lossy branch even given a lossy branch.

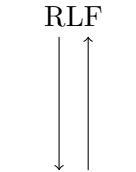
Relations

RLF

ABO-RLF

OT-RLF

Relations



OT-RLF

Relations

Relations

- ABO-RLF: the lossy branch is fixed by ek

Relations

- ABO-RLF: the lossy branch is fixed by ek
- OT-RLF: the lossy branch can be generated "on-the-fly" with td in a semi-customized manner

Outline

Concrete Construction from the DDH Assumption

Matrix approach for ABO-LTFs due to Peikert and Waters

- input: n-dimension vector $\mathbf{x}$ over $\mathbb{Z}_2$
- evaluation key: $n \times (n+1)$ matrix $\mathbf{M}$ over $\mathbb{G}$
- output: $\mathbf{xM}$

$$\mathbf{M} = \begin{cases} \text{invertible} & \text{if } \mathrm{rank}(\mathbf{M}) = n \\ \text{lossy} & \text{if } \mathrm{rank}(\mathbf{M}) < n \end{cases}$$

To ensure invertible property

- input space is restricted to $\{0,1\}^n$
- line dimension $m = n + 1$

For (ABO)-RLFs, we do not require invertible or even injective

- input space extends to $\mathbb{Z}_q^n$
- line dimension $m \ll n$

Concealer Matrix

$\mathsf{GenConceal}(n, m) \to \mathbb{G}^{n \times m}$

1. Choose $\mathbf{r} = (r_1, \ldots, r_n) \in \mathbb{Z}_p$ and $\mathbf{s} = (s_1, \ldots, s_m) \in \mathbb{Z}_p$
2. $\mathbf{V} = \mathbf{r} \otimes \mathbf{s} = \mathbf{r}^t \mathbf{s}$
3. Output $\mathbf{C} = g^{\mathbf{V}} \in \mathbb{G}^{n \times m}$ as the concealer matrix

$$\mathbb{G} = \begin{pmatrix} g^{r_1 s_1} & g^{r_1 s_2} & \ldots & g^{r_1 s_m} \\ g^{r_2 s_1} & g^{r_2 s_2} & \ldots & g^{r_2 s_m} \\ \vdots & \vdots & \vdots & \vdots \\ g^{r_n s_1} & g^{r_n s_2} & \ldots & g^{r_n s_m} \end{pmatrix}$$

- all columns lie in a one-dimensional subspace
- $\mathbf{C}$ is pseudorandom under the DDH assumption

We construct ABO-RLF with $B = \mathbb{Z}_p$ as below:

- $\mathsf{Gen}(\lambda, b^*)$: $\mathsf{GenConceal}(n, m) \to \mathbf{C} = g^{\mathbf{V}}$, output $ek = g^{\mathbf{Y}} = g^{\mathbf{V} - b^*\mathbf{I}'}$, where $\mathbf{I}' = (\mathbf{e}_1, \ldots, \mathbf{e}_m)$.
- $f_{ek,b}(\mathbf{x})$: $g^{\mathbf{x}(\mathbf{Y}+b\mathbf{I}')} = g^{\mathbf{x}(\mathbf{V}+(b-b^*)\mathbf{I}')} \in \mathbb{G}^m$.

Lemma 3

The above construction constitutes $(p^{n-m}, \log p)$-ABO-RLF.

- For any $b \neq b^*$, $\mathrm{rank}(\mathbf{Y} + b\mathbf{I}') = m$ and the size of the solution space for every $y \in \mathbb{G}^m$ is p^{n-m}.
- For $b = b^*$, $\mathrm{rank}(\mathbf{Y} + b\mathbf{I}') = 1$ and thus the image size is p.
- Pseudorandomness of $\mathbf{C} \Rightarrow$ hidden lossy branch

Extension and Comparison

Our DDH construction applies to the extended DDH, which generalize DDH, QR, DCR

We have a more efficient and direct DCR-based construction

DDH/DCR	Input	Lossiness	Key	Efficiency
[PW08]	2^n	$n - \log p$	$nm\|\mathbb{G}\|$	nm Add
ABO-RLF	p^n	$(n-1)\log p$	$nm\|\mathbb{G}\|$	nm (Exp+Add)
[FGK+13]	N^2	$\log N$	$\|\mathbb{Z}^*_{N^3}\|$	1 Exp
ABO-LF	$N^2/4$	$\log N$	$\|\mathbb{Z}^*_{N^2}\|$	1 Exp

Generic Construction from HPS

Wee (Eurocrypt 2012): dual HPS $\Rightarrow$ LTF

- HPS has to satisfy strong property
- No efficient ABO construction is known

Generic Construction from HPS

Wee (Eurocrypt 2012): dual HPS $\Rightarrow$ LTF

- HPS has to satisfy strong property
- No efficient ABO construction is known

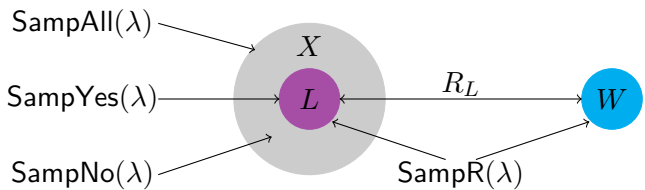
We show $\boxed{\text{HPS} \Rightarrow \text{ABO-RLF}}$

- exploit algebra property of the underlying SMP

(Algebraic) Subset Membership Problem

Task: distinguish

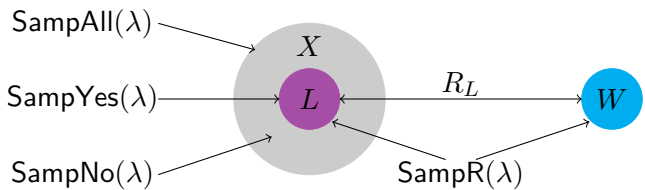
Solution: $\{0, 1\}$



(Algebraic) Subset Membership Problem

Task: distinguish

Solution: $\{0, 1\}$



SMP: $U_X \approx_c U_L$

Algebraic SMP

- X forms an Abelian group, L forms a subgroup of X
- The quotient group $H = X/L$ is cyclic with order $p = |X|/|L|$

Algebraic SMP

- X forms an Abelian group, L forms a subgroup of X
- The quotient group $H = X/L$ is cyclic with order $p = |X|/|L|$

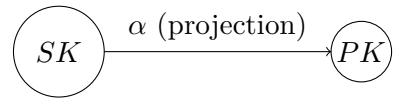
Algebraic properties $\Rightarrow$ two useful facts

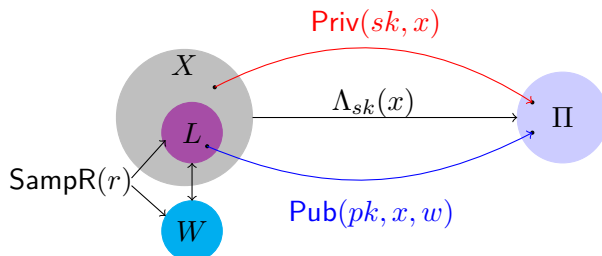
1. Let $\bar{a} = aL$ for some $a \in X \backslash L$ be a generator of H, the co-sets $(aL, 2aL, \ldots, (p-1)aL, paL = L)$ constitute a partition of X.
2. For each $x \in L$, $ia + x \in X \backslash L$ for $1 \leq i < p$

Hash Proof System

- $L \subset X$ — language defined by R_L associated with SMP.
- HPS equips $L \subset X$ with Gen, Priv, Pub.

$\mathsf{Gen}(\lambda) \to (pk, sk)$
s.t. $\alpha(sk) = pk$

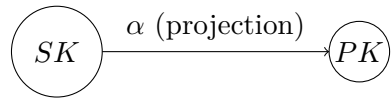


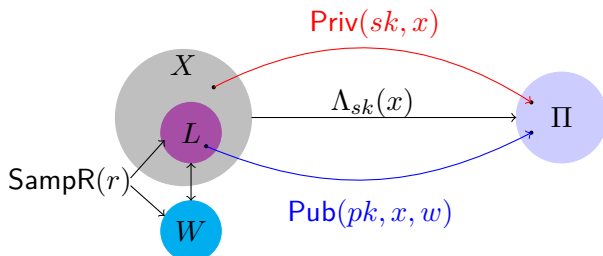


Hash Proof System

- $L \subset X$ — language defined by R_L associated with SMP.
- HPS equips $L \subset X$ with Gen, Priv, Pub.

$\mathsf{Gen}(\lambda) \to (pk, sk)$
s.t. $\alpha(sk) = pk$





Projective: $\forall x \in L$, $\Lambda_{sk}(x)$ is uniquely determined by x and $pk \leftarrow \alpha(sk)$.

ABO-RLF from HPS for ASMP

Let aL be a generator for $H = X/L$, we build ABO-RLF from HPS for ASMP as below:

- $\mathsf{Gen}(\lambda, b^*)$: $(x, w) \leftarrow \mathsf{SampYes}(\lambda)$, output $ek = -b^*a + x$
- $f_{ek,b}(x)$: output $\alpha(sk)||\Lambda_{sk}(ek + ba)$

ABO-RLF from HPS for ASMP

Let aL be a generator for $H = X/L$, we build ABO-RLF from HPS for ASMP as below:

- $\mathsf{Gen}(\lambda, b^*)$: $(x, w) \leftarrow \mathsf{SampYes}(\lambda)$, output $ek = -b^*a + x$
- $f_{ek,b}(x)$: output $\alpha(sk)||\Lambda_{sk}(ek + ba)$

Theorem 4

Assume $X = \{0,1\}^n$ and the function $g_x(sk) := \alpha(sk)||\Lambda_{sk}(x)$ is v-regular for any $x \notin L$. The above construction is $(v, \log|Img\alpha|)$-ABO-RLF under ASMP.

ABO-RLF from HPS for ASMP

Let aL be a generator for $H = X/L$, we build ABO-RLF from HPS for ASMP as below:

- $\mathsf{Gen}(\lambda, b^*)$: $(x, w) \leftarrow \mathsf{SampYes}(\lambda)$, output $ek = -b^*a + x$
- $f_{ek,b}(x)$: output $\alpha(sk)||\Lambda_{sk}(ek + ba)$

Theorem 4

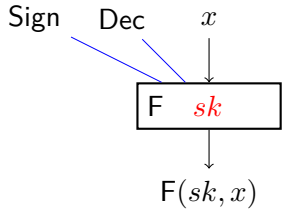
Assume $X = \{0,1\}^n$ and the function $g_x(sk) := \alpha(sk)||\Lambda_{sk}(x)$ is v-regular for any $x \notin L$. The above construction is $(v, \log|Img\alpha|)$-ABO-RLF under ASMP.

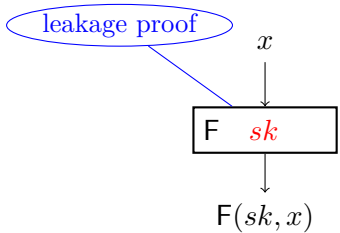
- $ek + ba = x + (b - b^*)a \notin L$ if $b \neq b^* \Rightarrow v$-regular
- $ek + ba = x + (b - b^*)a \in L$ if $b = b^* \Rightarrow$ lossy by the projective property
- ASMP $\Rightarrow$ Hidden lossy branch. For any $b_0^*, b_1^* \in \mathbb{Z}_p$:

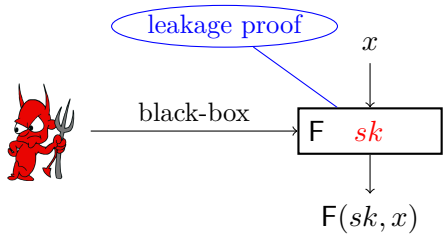
$$(-b_0^*a + x) \approx_c (b_0^*a + u) \equiv (b_1^*a + u) \approx_c (b_1^*a + x)$$

where $u \xleftarrow{R} X$.

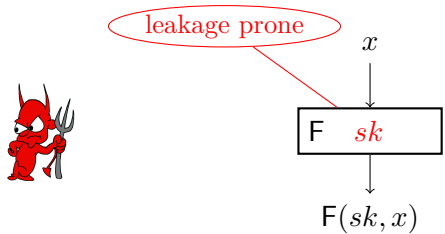
Outline





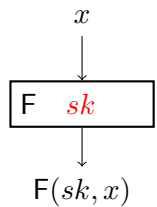


- Traditional security model assumes *black-box access* to cryptographic device.



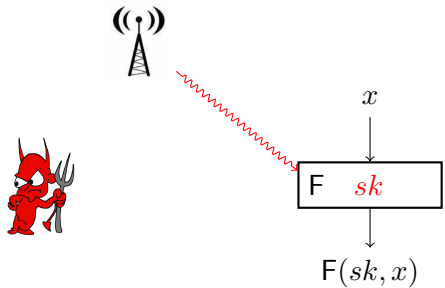
- Traditional security model assumes *black-box access* to cryptographic device.

leakage attacks (since 1996) invalidate this idealized assumption



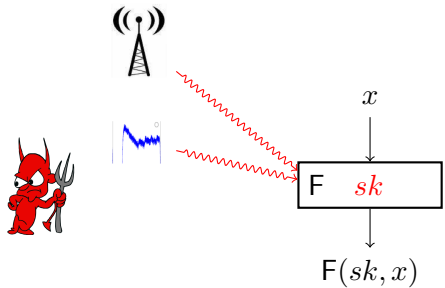
- Traditional security model assumes *black-box access* to cryptographic device.

leakage attacks (since 1996) invalidate this idealized assumption



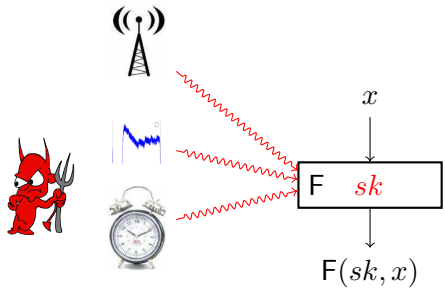
- Traditional security model assumes *black-box access* to cryptographic device.

leakage attacks (since 1996) invalidate this idealized assumption



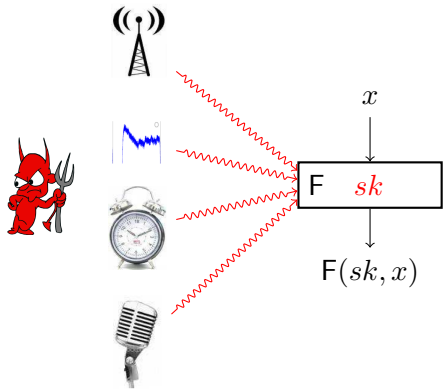
- Traditional security model assumes *black-box access* to cryptographic device.

leakage attacks (since 1996) invalidate this idealized assumption



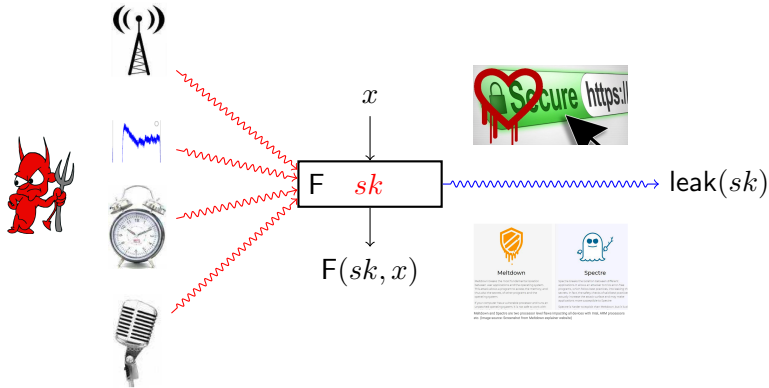
- Traditional security model assumes *black-box access* to cryptographic device.

leakage attacks (since 1996) invalidate this idealized assumption



- Traditional security model assumes *black-box access* to cryptographic device.

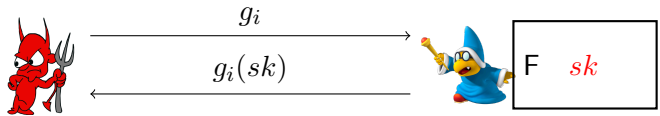
leakage attacks (since 1996) invalidate this idealized assumption



- Traditional security model assumes *black-box access* to cryptographic device.

Bounded Leakage Model

In this work, we focus on a simple yet general leakage model called Bounded Leakage Model



$$\sum |g_i(sk)| \leq |sk|$$

Leakage-Resilient OWFs

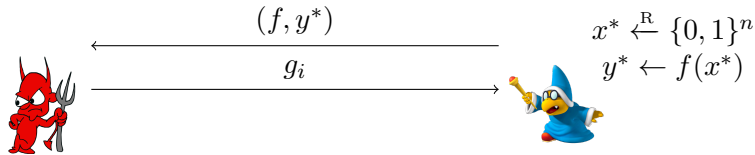
Leakage-Resilient OWFs

(f, y^*)

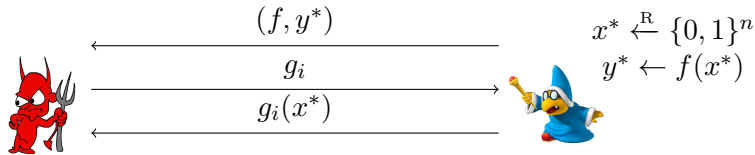
$x^* \xleftarrow{R} \{0,1\}^n$

$y^* \leftarrow f(x^*)$

Leakage-Resilient OWFs



Leakage-Resilient OWFs



Leakage-Resilient OWFs

(f, y^*)

g_i

$g_i(x^*)$

x

$x^* \xleftarrow{R} \{0,1\}^n$

$y^* \leftarrow f(x^*)$

$x =? x^*$

(f, y^*)

g_i

$g_i(x^*)$

x

$x^* \xleftarrow{R} \{0,1\}^n$

$y^* \leftarrow f(x^*)$

$x =? x^*$

Theorem 5

The normal mode of $(1, \tau)$-RLFs (i.e., LFs) over domain $\{0,1\}^n$ constitutes a family of ℓ-leakage-resilient injective OWFs, for any $\ell \leq n - \tau - \omega(\log \lambda)$.

Game 0: real game

1. Setup: $\mathcal{CH}$ generates $f \leftarrow \mathrm{RLF.GenNormal}(\lambda)$, picks $x^* \xleftarrow{R} \{0,1\}^n$ and sends $(f, y^* = f(x^*))$ to $\mathcal{A}$.
2. Leakage queries: $\mathcal{A} \hookrightarrow g_i$, $\mathcal{CH}$ responds with $g_i(x^*)$.
3. Invert: $\mathcal{A}$ outputs x and wins if $x = x^*$.

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

Game 1: same as Game 0 except that:

1. Setup: $\mathcal{CH}$ generates $\boxed{f \leftarrow \mathrm{RLF.GenLossy}(\lambda)}$.

Security of RLFs $\Rightarrow |\Pr[S_1] - \Pr[S_0]| \leq \mathsf{negl}(\lambda)$

In Game 1, $\tilde{\mathsf{H}}_\infty(x^*|(y^*, leak)) \geq n - \tau - \ell$.

- By the parameter choice, $\tilde{\mathsf{H}}_\infty(x^*|(y^*, leak)) \geq \omega(\log \lambda) \Rightarrow \Pr[S_1] \leq \mathsf{negl}(\lambda)$ even w.r.t. unbounded adversary

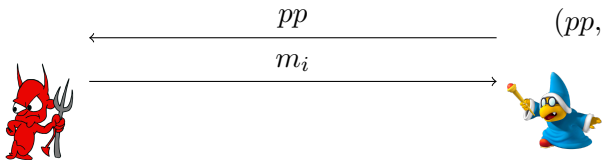
Leakage-Resilient MAC

Leakage-Resilient MAC

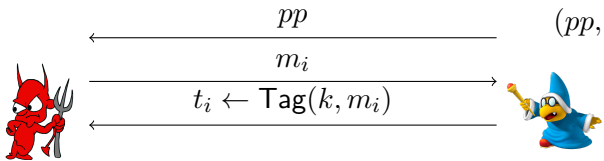
pp

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

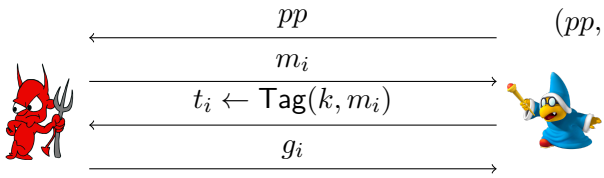
Leakage-Resilient MAC



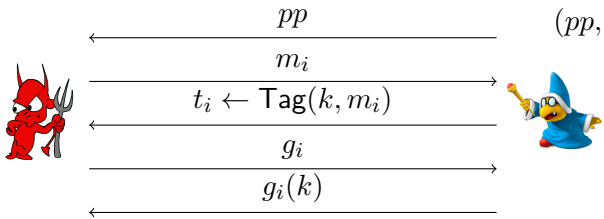
Leakage-Resilient MAC



Leakage-Resilient MAC



Leakage-Resilient MAC



Leakage-Resilient MAC

pp

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

m_i

$t_i \leftarrow \mathsf{Tag}(k, m_i)$

g_i

$g_i(k)$

(m^*, t^*)

$\mathsf{Vefy}(k, m^*, t^*) = 1$

$(m^*, t^*) \neq (m_i, t_i)$

Leakage-Resilient MAC

pp

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

m_i

$t_i \leftarrow \mathsf{Tag}(k, m_i)$

g_i

$g_i(k)$

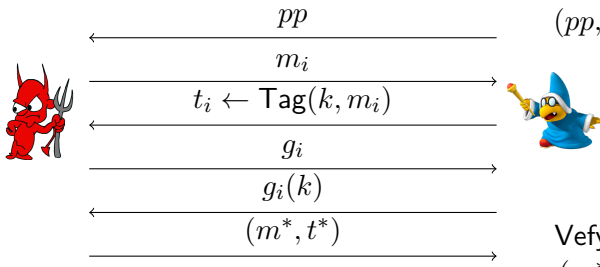
(m^*, t^*)

$\mathsf{Vefy}(k, m^*, t^*) = 1$

$(m^*, t^*) \neq (m_i, t_i)$

Strong unforgeability can be relaxed in several ways:

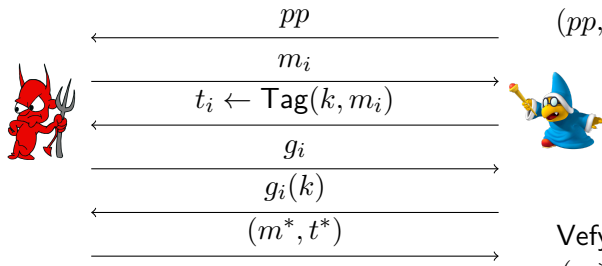
Leakage-Resilient MAC



Strong unforgeability can be relaxed in several ways:

- One-time: $\mathcal{A}$ only makes one tag query

Leakage-Resilient MAC



Strong unforgeability can be relaxed in several ways:

- One-time: $\mathcal{A}$ only makes one tag query
- Static: $\mathcal{A}$ specifies tag queries before seeing pp

Construction

Ingredient
(v, τ)-ABORLF

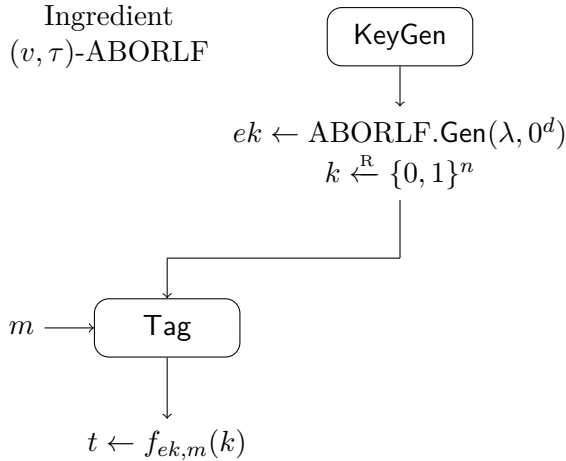
Construction

Ingredient
(v, τ)-ABORLF

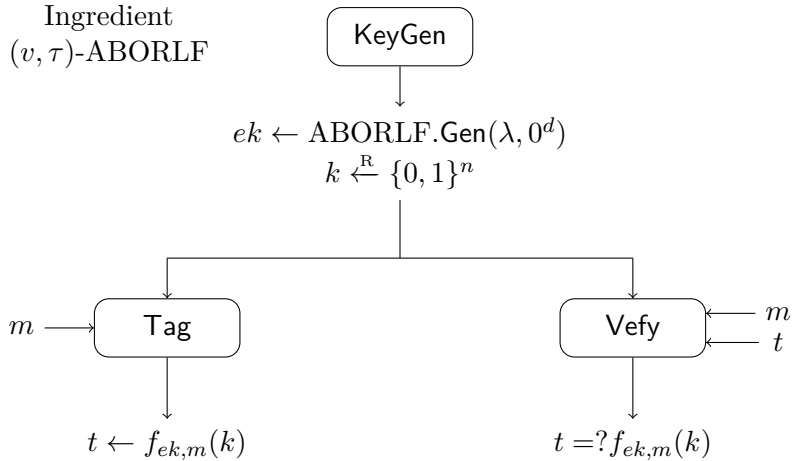
KeyGen

$ek \leftarrow \text{ABORLF.}\mathsf{Gen}(\lambda, 0^d)$
$k \xleftarrow{\text{R}} \{0,1\}^n$

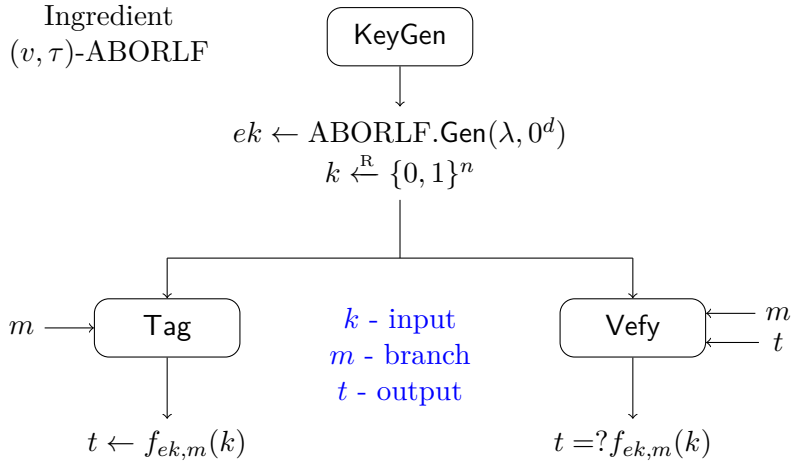
Construction



Construction



Construction



Theorem 6

The above MAC is ℓ-leakage-resilient seletively one-time sUF for any $\ell \leq n - \tau - \log v - \omega(\log \lambda)$.

Game 0: (real game)

1. Setup: $\mathcal{A} \nrightarrow m^*$, $\mathcal{CH}$ generates $ek \leftarrow \text{ABORLF.}\mathsf{Gen}(\lambda, 0^d)$, picks $k \xleftarrow{\text{R}} \{0,1\}^n$, computes $t^* \leftarrow f_{ek,m^*}(k)$ and then sends (ek, t^*) to $\mathcal{A}$.
2. Leakage queries: $\mathcal{A} \nrightarrow g_i$, $\mathcal{CH}$ responds with $g_i(k)$.
3. Forge: $\mathcal{A} \rightarrow (m, t)$ and wins if $m \neq m^* \wedge t = f_{ek,m}(k)$.

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

Game 1: same as Game 0 except that

1. Setup: $\mathcal{CH}$ generates $\boxed{ek \leftarrow \text{ABORLF.Gen}(\lambda, m^*)}$.

Hidden lossy branch $\Rightarrow |\Pr[S_1] - \Pr[S_0]| \leq \mathsf{negl}(\lambda)$

In Game 1, $\mathcal{A}$'s view includes $(ek, leak, t^*)$. We have:

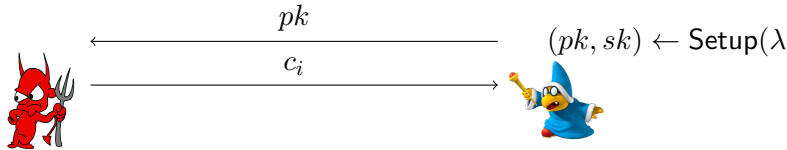
$$\begin{aligned} \tilde{\mathsf{H}}_\infty(t|view) &= \tilde{\mathsf{H}}_\infty(t|ek, leak, t^*) \\ &\geq \tilde{\mathsf{H}}_\infty(t|ek) - \ell - \tau \\ &\geq \tilde{\mathsf{H}}_\infty(k|ek) - \log v - \ell - \tau \\ &= n - \log v - \ell - \tau \end{aligned}$$

- By the parameter choice, $\tilde{\mathsf{H}}_\infty(t|view) \geq \omega(\log \lambda) \Rightarrow \Pr[S_1] \leq \mathsf{negl}(\lambda)$ even w.r.t. unbounded adversary.

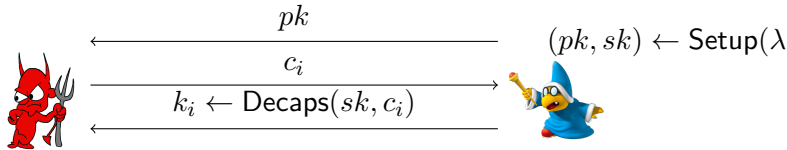
Leakage-Resilient CCA-secure KEM

Leakage-Resilient CCA-secure KEM

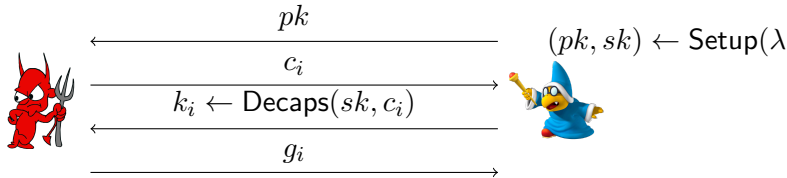
Leakage-Resilient CCA-secure KEM



Leakage-Resilient CCA-secure KEM



Leakage-Resilient CCA-secure KEM



Leakage-Resilient CCA-secure KEM

pk

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

c_i

$k_i \leftarrow \mathsf{Decaps}(sk, c_i)$

g_i

$g_i(sk)$

Leakage-Resilient CCA-secure KEM

pk

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

c_i

$k_i \leftarrow \mathsf{Decaps}(sk, c_i)$

g_i

$g_i(sk)$

$(c^*, k_0^*) \leftarrow \mathsf{Encap}(pk)$

$k_1^* \xleftarrow{\mathrm{R}} K$

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

(c^*, k_β^*)

Leakage-Resilient CCA-secure KEM

pk

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

c_i

$k_i \leftarrow \mathsf{Decaps}(sk, c_i)$

g_i

$g_i(sk)$

(c^*, k^*_β)

β'

$(c^*, k^*_0) \leftarrow \mathsf{Encap}(pk)$

$k^*_1 \xleftarrow{\mathrm{R}} K$

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

$\beta' = \beta$

Leakage-Resilient CCA-secure KEM

pk ←

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

c_i →

$k_i \leftarrow \mathsf{Decaps}(sk, c_i)$ ←

g_i →

$(c^*, k_0^*) \leftarrow \mathsf{Encap}(pk)$

$g_i(sk)$ ←

$k_1^* \xleftarrow{\mathrm{R}} K$

(c^*, k_β^*) ←

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

β' →

$\beta' = \beta$

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

Construction

Ingredients
HPS
ABORLF
strong extractor

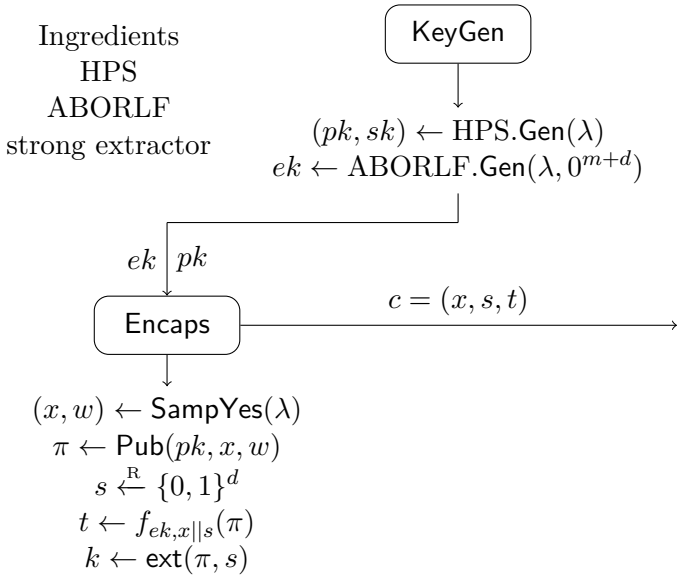
Construction

Ingredients
HPS
ABORLF
strong extractor

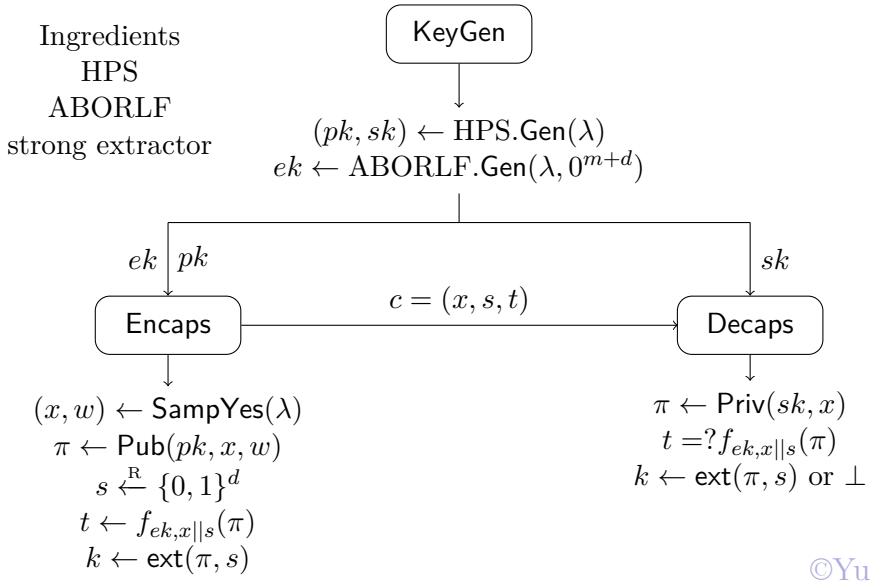
KeyGen

$(pk, sk) \leftarrow \text{HPS.}\mathsf{Gen}(\lambda)$
$ek \leftarrow \text{ABORLF.}\mathsf{Gen}(\lambda, 0^{m+d})$

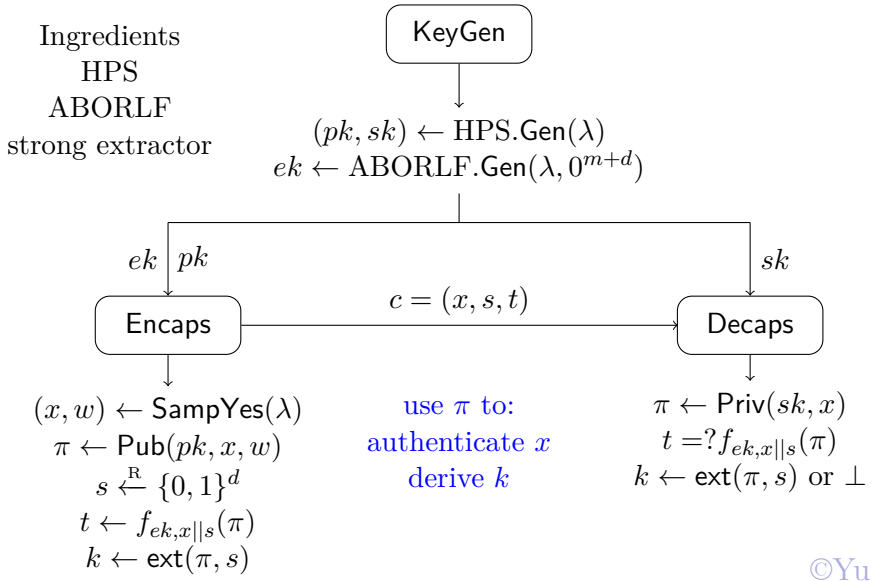
Construction



Construction



Construction



Theorem 7

Suppose SMP for $L \subset \{0,1\}^m$ is hard, HPS is ϵ_1-universal$_1$ and $n = \log(1/\epsilon_1)$, ABORLF is (v,τ)-regularly-lossy, ext is $(n-\tau-\ell,\kappa,\epsilon_2)$-strong extractor, then the above KEM is ℓ-LR CCA secure for any $\ell \leq n - \tau - \log v - \omega(\log\lambda)$.

Game 0: (real game)

1. Setup: $\mathcal{CH}$ generates $(pk, sk) \leftarrow \mathrm{HPS.}\mathsf{Gen}(\lambda)$, $ek \leftarrow \mathrm{ABORLF.}\mathsf{Gen}(\lambda, 0^{m+d})$, sends (pk, ek) to $\mathcal{A}$.
2. Leakage queries $\langle g_i \rangle$: $\mathcal{CH}$ responds with $g_i(sk)$.
3. Challenge: $\mathcal{CH}$ picks $\beta \in \{0,1\}$, $s^* \leftarrow \{0,1\}^d$, $(x^*, w^*) \leftarrow \mathsf{SampYes}(\lambda)$, computes $\pi^* \leftarrow \mathsf{Pub}(pk, x^*, w^*)$, $t^* \leftarrow f_{ek,x^*||s^*}(\pi^*)$, $k_0^* \leftarrow \mathsf{ext}(\pi^*, s^*)$, picks $k_1^* \leftarrow \{0,1\}^\kappa$, sends $c^* = (x^*, s^*, t^*)$ and k_β^* to $\mathcal{A}$
4. Decaps queries $\langle c = (x,s,t) \neq c^* \rangle$: $\mathcal{CH}$ computes $\pi \leftarrow \Lambda_{sk}(x)$, output $k \leftarrow \mathsf{ext}(\pi, s)$ if $t = f_{ek,x||s}(\pi)$ and $\perp$ otherwise.

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

Game 1: $\mathcal{CH}$ samples (x^*, w^*) and s^* at Setup.

$$\Pr[S_0] = \Pr[S_1]$$

Game 1: $\mathcal{CH}$ samples (x^*, w^*) and s^* at Setup.

$$\Pr[S_0] = \Pr[S_1]$$

Game 2: $\mathcal{CH}$ generates $ek \leftarrow \text{ABORLF.}\mathsf{Gen}(\lambda, x^*||s^*)$.

$$\text{Hidden lossy branch} \Rightarrow |\Pr[S_2] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$$

Game 1: $\mathcal{CH}$ samples (x^*, w^*) and s^* at Setup.

$$\Pr[S_0] = \Pr[S_1]$$

Game 2: $\mathcal{CH}$ generates $ek \leftarrow \text{ABORLF.}\mathsf{Gen}(\lambda, x^* || s^*)$.

$$\text{Hidden lossy branch} \Rightarrow |\Pr[S_2] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$$

Game 3: $\mathcal{CH}$ computes $\pi^* \leftarrow \Lambda_{sk}(x^*)$ via $\mathsf{Priv}(sk, x^*)$.

$$\text{Correctness of HPS} \Rightarrow \Pr[S_3] = \Pr[S_2].$$

Game 1: $\mathcal{CH}$ samples (x^*, w^*) and s^* at Setup.

$$\Pr[S_0] = \Pr[S_1]$$

Game 2: $\mathcal{CH}$ generates $ek \leftarrow \text{ABORLF.}\mathsf{Gen}(\lambda, x^* || s^*)$.

$$\text{Hidden lossy branch} \Rightarrow |\Pr[S_2] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$$

Game 3: $\mathcal{CH}$ computes $\pi^* \leftarrow \Lambda_{sk}(x^*)$ via $\mathsf{Priv}(sk, x^*)$.

$$\text{Correctness of HPS} \Rightarrow \Pr[S_3] = \Pr[S_2].$$

Game 4: $\mathcal{CH}$ samples x^* via SampNo rather than $\mathsf{SampYes}$.

$$\text{SMP} \Rightarrow |\Pr[S_4] - \Pr[S_3]| \leq \mathsf{negl}(\lambda)$$

Game 1: $\mathcal{CH}$ samples (x^*, w^*) and s^* at Setup.

$$\Pr[S_0] = \Pr[S_1]$$

Game 2: $\mathcal{CH}$ generates $ek \leftarrow \text{ABORLF.}\mathsf{Gen}(\lambda, x^*||s^*)$.

$$\text{Hidden lossy branch} \Rightarrow |\Pr[S_2] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$$

Game 3: $\mathcal{CH}$ computes $\pi^* \leftarrow \Lambda_{sk}(x^*)$ via $\mathsf{Priv}(sk, x^*)$.

$$\text{Correctness of HPS} \Rightarrow \Pr[S_3] = \Pr[S_2].$$

Game 4: $\mathcal{CH}$ samples x^* via SampNo rather than $\mathsf{SampYes}$.

$$\text{SMP} \Rightarrow |\Pr[S_4] - \Pr[S_3]| \leq \mathsf{negl}(\lambda)$$

Game 5: $\mathcal{CH}$ directly rejects $\langle c = (x, s, t)\rangle$ if $x \notin L$. Define E: $\mathcal{A}$ makes an invalid but well-formed decaps queries, i.e., $f_{ek,x||s}(\Lambda_{sk}(x)) = t$ and $x \in L \wedge (x, s, t) \neq (x^*, s^*, t^*)$.

$$|\Pr[S_5] - \Pr[S_4]| \leq \Pr[E]$$

To calculate $\Pr[E]$, it suffice to bound $\tilde{\mathsf{H}}_\infty(t|view)$.

- $view$: $(pk, ek, leak, x^*, s^*, t^*, k^*_\beta)$
- $t = f_{ek,x||s}(\Lambda_{sk}(x))$

We bound $\tilde{\mathsf{H}}_\infty(t|view)$ via $\tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x)|view)$ as below:

- (x^*, s^*) determines a lossy branch $\Rightarrow \tau$ only reveal partial info about sk $\Rightarrow$ $\tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x)|view) \geq n - \ell - \tau - \kappa$
- We must have $(x, s) \neq (x^*, s^*)$, which determines a v-regular branch $\Rightarrow$ $\tilde{\mathsf{H}}_\infty(t|view) \geq \tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x)|view) - \log v$

By the parameter choice, $\tilde{\mathsf{H}}_\infty(t|view) \geq \omega(\log \lambda)$, thus we have:

$$\Pr[E] \leq \mathsf{negl}(\lambda)$$

Game 6: $\mathcal{CH}$ samples $k_0^* \leftarrow \{0,1\}^\kappa$ rather than $k_0^* \leftarrow \mathsf{ext}(\Lambda_{sk}(x^*))$. Next, we analysis $\Delta[view_5, view_6]$.

- define $view' = (pk, ek, leak, x^*, s^*, t^*)$, chain rule $\Rightarrow$ $\tilde{\mathsf{H}}_\infty(\Lambda_{sk}(x^*)|view') \geq n - \ell - \tau$
- randomness extractor $\Rightarrow \Delta[(view', k_{5,0}^*), (view', k_{6,0}^*)] \leq \epsilon_2$.
- responses to all decaps queries in Game 5 and 6 are determined by the same function of $(view', k_{5,0}^*)$ and $(view', k_{6,0}^*)$ resp.

$$\Delta[view_5, view_6] \leq \epsilon_2/2 \leq \mathsf{negl}(\lambda)$$

Putting all the above together, $\mathsf{Adv}_{\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.

Importance

Universal$_1$ HPS + ABO RLF $\Rightarrow$ LR-CCA KEM

- proper parameter choice $\Rightarrow \ell/|sk| = 1 - o(1)$
- HPS $\Rightarrow$ ABO-RLF

construct CCA-secure KEM with optimal leakage rate based solely on universal$_1$ HPS

- go beyond the upper bound posed by Dodis et al. (Asiacrypt 2010)

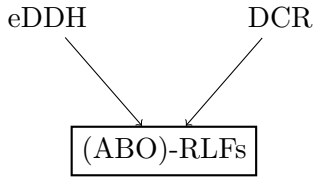
leakage-rate only approaching 1/6. Unfortunately, it seems that the hash proof system approach to building CCA encryption is inherently limited to leakage-rates below 1/2: this is because the secret-key consists of two components (one for verifying that the ciphertext is well-formed and one for decrypting it) and the proofs break down if either of the components is individually leaked in its entirety.

- extend to identity-based setting as well

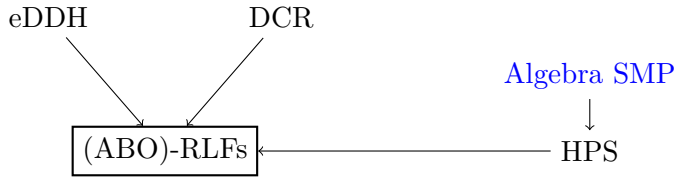
Conclusion

(ABO)-RLFs

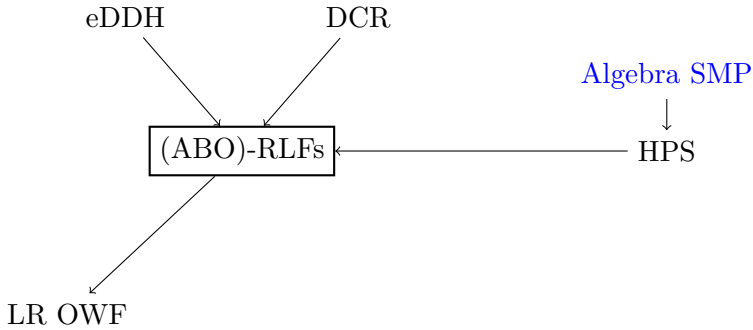
Conclusion



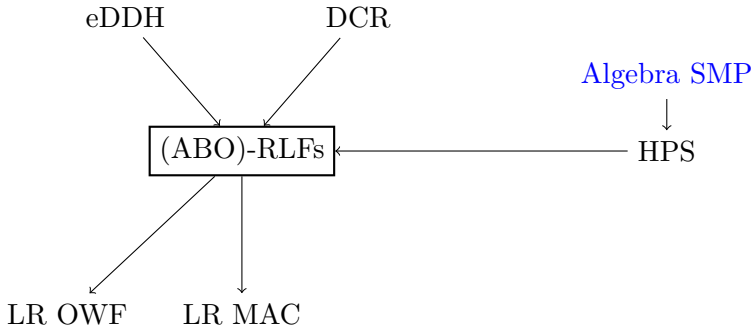
Conclusion



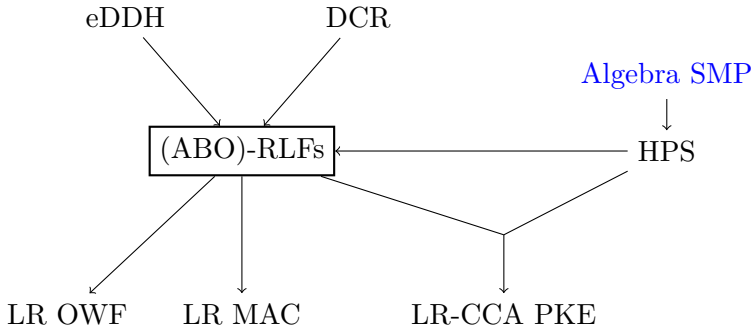
Conclusion



Conclusion



Conclusion



Any Questions?

Thanks for listening!

[FGK+13] David Mandell Freeman, Oded Goldreich, Eike Kiltz, Alon Rosen, and Gil Segev. More constructions of lossy and correlation-secure trapdoor functions. *J. Cryptology*, 26(1):39–74, 2013.

[PW08] 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*, pages 187–196. ACM, 2008.