Regular Lossy Functions and Applications in Leakage-Resilient Cryptography

Yu Chen\textsuperscript{1} Baodong Qin\textsuperscript{2} Haiyang Xue\textsuperscript{1}

\textsuperscript{1}SKLOIS, IIE, Chinese Academy of Sciences
\textsuperscript{2}Xi’an University of Posts and Telecommunication

CT-RSA 2018
April 20th, 2018
Outline

1 Backgrounds

2 Regular Lossy Functions

3 Constructions of ABO RLFs
   - Concrete Construction
   - Generic Construction

4 Applications of RLFs
   - Leakage-Resilient OWFs
   - Leakage-Resilient MAC
   - Leakage-Resilient CCA-secure KEM
Outline

1. Backgrounds

2. Regular Lossy Functions

3. Constructions of ABO RLFs
   - Concrete Construction
   - Generic Construction

4. Applications of RLFs
   - Leakage-Resilient OWFs
   - Leakage-Resilient MAC
   - Leakage-Resilient CCA-secure KEM
Lossy Trapdoor Functions

Lossy object *indistinguishable* from original

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

injective
\[ \text{Gen}(\lambda) \rightarrow (ek, td) \]

\[ \approx_c \]

lossy
\[ \text{Gen}(\lambda) \rightarrow (ek, \perp) \]

\[ 2^n \gg 2^\tau = 2^{n-\ell} \]
Extension of LTFs: ABO LTFs

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

\[
\text{Gen}(\lambda, b^*) \rightarrow (ek, td)
\]

\[
\begin{align*}
f_{ek,b_1}(\cdot) & \quad f_{ek,b_i}(\cdot) & \quad (\text{b^* is hidden from } ek) \\
f_{ek,b_2}(\cdot) & \quad f_{ek,b^*}(\cdot) \\
\ldots & \\
\end{align*}
\]

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

$\text{LTFs} \Leftrightarrow \text{ABO LTFs}$
Constructions and Applications

(ABO)-LTF
Constructions and Applications

- DDH
- QR/DCR
- LWE

(ABO)-LTF
Constructions and Applications

DDH \quad QR/DCR \quad LWE

Homo/Dual HPS \xrightarrow{(ABO)-LTF} TDF \quad CRHF \quad CCA PKE

CP-TDF \quad OT \quad Lossy PKE

ATDF \quad D-PKE
Constructions and Applications

- DDH
- QR/DCR
- LWE
- (ABO)-LTF
- ABM-LTF
- Homo/Dual HPS
- TDF
- CRHF
- CCA PKE
- CP-TDF
- OT
- Lossy PKE
- ATDF
- D-PKE
Motivations

In all applications of LTF:

- normal mode: \textit{injective+trapdoor} fulfill functionality
- lossy mode: establish security
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 \(c\) lossy
A central goal in cryptography is to base cryptosystems on primitives 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.
A central goal in cryptography is to base cryptosystems on primitives 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

1 Backgrounds

2 Regular Lossy Functions

3 Constructions of ABO RLFs
   - Concrete Construction
   - Generic Construction

4 Applications of RLFs
   - Leakage-Resilient OWFs
   - Leakage-Resilient MAC
   - Leakage-Resilient CCA-secure KEM
A Simple But Important Observation

When trapdoor is not required for normal mode, the injective property may also be unnecessary.
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 LF's Regular Lossy Functions
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 LF's

Regular Lossy Functions

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

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

Regular Lossy Functions

Intuition: the output should 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

\[ f \text{ is } v\text{-to-1 (or } v\text{-regular)} \text{ if } \max_y |f^{-1}(y)| \leq v. \]
Regular Lossy Functions

\[ \text{normal} \quad \text{Gen}(\lambda) \rightarrow ek \]

\[ \approx_c \]

\[ \text{lossy} \quad \text{Gen}(\lambda) \rightarrow ek \]

- When \( v = 1 \), RLFs specialize to standard LF s

\[ 2^n \gg 2^\tau = 2^{n-\ell} \]
Why we choose **regularity** but not **image size** to capture normal mode?
Remarks

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

Lemma 2: Let $f$ be a $v$-to-1 function and $x$ be a random variable over the domain: $H_1(f(x)) \geq H_1(x) - \log v$. 

$\tilde{H}_\infty(x|f(x))$
Why we 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 $\tilde{H}_\infty(x|f(x))$ by the chain rule.

- In contrast, **regularity** is a *local* characterization, which suffices to give the lower bound of $H_\infty(f(x))$.

The following technical 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:*

\[
H_\infty(f(x)) \geq H_\infty(x) - \log v
\]
**All-But-One Regular Lossy Functions**

- **Gen(λ, b*)** has an extra input: branch $b^* \in B$.

\[ \text{Gen}(\lambda, b^*) \rightarrow ek \]

\[
\begin{align*}
  f_{ek,b_1}(\cdot) & \quad  f_{ek,b_i}(\cdot) \\
  f_{ek,b_2}(\cdot) & \quad  f_{ek,b^*}(\cdot) \\
  \ldots
\end{align*}
\]

$b^*$ is hidden from $ek$

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

\[ \text{RLF} \Leftrightarrow \text{ABO-RLF} \]
Outline

1 Backgrounds

2 Regular Lossy Functions

3 Constructions of ABO RLFs
   • Concrete Construction
   • Generic Construction

4 Applications of RLFs
   • Leakage-Resilient OWFs
   • Leakage-Resilient MAC
   • Leakage-Resilient CCA-secure KEM
Concrete Construction from the DDH Assumption

Matrix approach for ABO-LTFs $f_{ek,b}(x) \rightarrow y$ due to Peikert and Waters
Concrete Construction from the DDH Assumption

Matrix approach for ABO-LTFs $f_{ek,b}(x) \rightarrow y$ due to Peikert and Waters

$x \in \mathbb{Z}_2^n$
Concrete Construction from the DDH Assumption

Matrix approach for ABO-LTFs $f_{ek,b}(x) \rightarrow y$ due to Peikert and Waters

$$\text{Gen} (\lambda, b^*) \rightarrow ek$$

$x \in \mathbb{Z}_2^n$
Concrete Construction from the DDH Assumption

Matrix approach for ABO-LTFs $f_{ek,b}(x) \rightarrow y$ due to Peikert and Waters

\[
\begin{align*}
\text{Gen}(\lambda, b^*) & \rightarrow ek \\
\text{GenConceal}(n, m) & = g^V \\
\downarrow \quad x \in \mathbb{Z}_2^n \\
\begin{pmatrix}
g^{r_1s_1} & g^{r_1s_2} & \ldots & g^{r_1s_m} \\
g^{r_2s_1} & g^{r_2s_2} & \ldots & g^{r_2s_m} \\
\vdots & \vdots & \ddots & \vdots \\
g^{r_ns_1} & g^{r_ns_2} & \ldots & g^{r_ns_m}
\end{pmatrix}
\end{align*}
\]
Concrete Construction from the DDH Assumption

Matrix approach for ABO-LTFs $f_{e_k, b}(x) \rightarrow y$ due to Peikert and Waters

$Gen(\lambda, b^*) \rightarrow ek$

$GenConceal(n, m) = g^V$

$x \in \mathbb{Z}_2^n$

\[
\begin{pmatrix}
g^{r_1s_1} & g^{r_1s_2} & \cdots & g^{r_1s_m} \\
g^{r_2s_1} & g^{r_2s_2} & \cdots & g^{r_2s_m} \\
\vdots & \vdots & \ddots & \vdots \\
g^{r_ns_1} & g^{r_ns_2} & \cdots & g^{r_ns_m}
\end{pmatrix}
\]

$DDH \Rightarrow \approx_c U_{\mathbb{G}^{n \times m}}$
Concrete Construction from the DDH Assumption

Matrix approach for ABO-LTFs $f_{ek,b}(x) \rightarrow y$ due to Peikert and Waters

$$\text{Gen}(\lambda, b^*) \rightarrow ek$$

$$\text{GenConceal}(n, m) = g^V$$

$$\begin{pmatrix}
g^{r_1 s_1} & g^{r_1 s_2} & \cdots & g^{r_1 s_m} \\
g^{r_2 s_1} & g^{r_2 s_2} & \cdots & g^{r_2 s_m} \\
& & \vdots & \vdots \\
g^{r_n s_1} & g^{r_n s_2} & \cdots & g^{r_n s_m}
\end{pmatrix} - b^*(e_1, \ldots, e_m)$$

$$x \in \mathbb{Z}_2^n$$

$DDH \Rightarrow \approx_c U_{G \times m}$
Concrete Construction from the DDH Assumption

Matrix approach for ABO-LTFs $f_{ek,b}(x) \rightarrow y$ due to Peikert and Waters

$$\text{Gen}(\lambda, b^*) \rightarrow ek$$

$$\text{GenConceal}(n, m) = g^V$$

$$x \in \mathbb{Z}_2^n \times \left( \begin{array}{cccc} g^{r_1s_1} & g^{r_1s_2} & \cdots & g^{r_1s_m} \\ g^{r_2s_1} & g^{r_2s_2} & \cdots & g^{r_2s_m} \\ \vdots & \vdots & \vdots & \vdots \\ g^{rn_1s_1} & g^{rn_1s_2} & \cdots & g^{rn_1s_m} \end{array} \right)$$

$$-b^*(e_1, \ldots, e_m) + b(e_1, \ldots, e_m) \rightarrow y \in \mathbb{G}^m$$

$$\text{DDH} \Rightarrow \approx_c U_{\mathbb{G}^n \times m}$$
Concrete Construction from the DDH Assumption

Matrix approach for ABO-LTFs \( f_{e_k,b}(x) \rightarrow y \) due to Peikert and Waters

\[
\text{Gen}(\lambda, b^*) \rightarrow e_k \\
\text{GenConceal}(n, m) = g^V \\
x \in \mathbb{Z}_2^n \times \\
\begin{pmatrix}
g^{r_{1s_1}} & g^{r_{1s_2}} & \ldots & g^{r_{1s_m}} \\
g^{r_{2s_1}} & g^{r_{2s_2}} & \ldots & g^{r_{2s_m}} \\
\vdots & \vdots & \vdots & \vdots \\
g^{r_{ns_1}} & g^{r_{ns_2}} & \ldots & g^{r_{ns_m}}
\end{pmatrix}
\]

\[-b^*(e_1, \ldots, e_m) + b(e_1, \ldots, e_m) \rightarrow y \in \mathbb{G}^m\]

DDH \( \Rightarrow \approx_c U_{\mathbb{G}^n \times m} \)

To ensure invertible property

- input space is restricted to \( \mathbb{Z}_2^n \) (a.k.a. \( \{0, 1\}^n \))
- column dimension \( m = n + 1 \)
\( x \in \mathbb{Z}_p^n \times \mathbb{Z}_2^m \)

\[
\begin{pmatrix}
g^{r_{1s_1}} & g^{r_{1s_2}} & \cdots & g^{r_{1s_m}} \\
g^{r_{2s_1}} & g^{r_{2s_2}} & \cdots & g^{r_{2s_m}} \\
\vdots & \vdots & \ddots & \vdots \\
g^{r_{ns_1}} & g^{r_{ns_2}} & \cdots & g^{r_{ns_m}}
\end{pmatrix}
\]

\[\text{DDH} \Rightarrow \approx_c U_{\mathbb{G}^{n \times m}}\]

\[\text{Gen}(\lambda, b^*) \rightarrow ek\]

\[\text{GenConceal}(n, m) = g^V\]

\[-b^*(e_1, \ldots, e_m) + b(e_1, \ldots, e_m) \rightarrow y \in \mathbb{G}^m\]
(ABO)-RLFs do not require invertible or even injective

\[ \text{Gen}(\lambda, b^*) \rightarrow ek \]

\[ \text{GenConceal}(n, m) = g^V \quad m \ll n \]

\[
\begin{pmatrix}
g^{r_1 s_1} & g^{r_1 s_2} & \ldots & g^{r_1 s_m} \\
g^{r_2 s_1} & g^{r_2 s_2} & \ldots & g^{r_2 s_m} \\
\vdots & \vdots & \ddots & \vdots \\
g^{r_n s_1} & g^{r_n s_2} & \ldots & g^{r_n s_m}
\end{pmatrix}
\]

\[-b^*(e_1, \ldots, e_m) + b(e_1, \ldots, e_m) \rightarrow y \in \mathbb{G}^m\]

\[ x \in \mathbb{Z}_p^n \times \mathbb{Z}_2^n \gg \mathbb{Z}_2^n \]

DDH \Rightarrow \sim_c U_{\mathbb{G}^n \times m}
(ABO)-RLFs do not require invertible or even injective

\[ \text{Gen}(\lambda, b^*) \rightarrow ek \]

\[ \text{GenConceal}(n, m) = g^V \quad m \ll n \]

\[ x \in \mathbb{Z}_p^n \times \begin{pmatrix} g^{r_1s_1} & g^{r_1s_2} & \ldots & g^{r_1s_m} \\ g^{r_2s_1} & g^{r_2s_2} & \ldots & g^{r_2s_m} \\ \vdots & \vdots & \ddots & \vdots \\ g^{rn_1} & g^{rn_2} & \ldots & g^{rn_m} \end{pmatrix} \rightarrow -b^*(e_1, \ldots, e_m) + b(e_1, \ldots, e_m) \rightarrow y \in \mathbb{G}^m \]

\[ \text{DDH} \Rightarrow^c \mathbb{U}_{\mathbb{G}^{n \times m}} \]

Lemma 3

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

- \(\forall b \neq b^*, \text{rank}(Y + bI') = m\) and \#(solution space) for every \(y \in \mathbb{G}^m\) is \(p^{n-m}\).
- \(b = b^*, \text{rank}(Y + bI') = 1\) and thus the image size is at most \(p\).
- Pseudorandomness of \(C = g^V \Rightarrow \text{hidden lossy branch}\)
Our DDH construction applies to extended DDH \( \sim \) generalize DDH, QR, DCR

- We have a more efficient and direct DCR-based construction

<table>
<thead>
<tr>
<th>ABO-LTF/RLF</th>
<th>Assump.</th>
<th>Input</th>
<th>Lossiness</th>
<th>Key</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABO-LTF[PW08]</td>
<td>DDH</td>
<td>(2^n)</td>
<td>(n - \log p)</td>
<td>(nm</td>
<td>G</td>
</tr>
<tr>
<td>ABO-RLF</td>
<td>DDH</td>
<td>(p^n)</td>
<td>((n - 1) \log p)</td>
<td>(nm</td>
<td>G</td>
</tr>
<tr>
<td>ABO-LTF[FGK+13]</td>
<td>DCR</td>
<td>(N^2)</td>
<td>(\log N)</td>
<td>(</td>
<td>\mathbb{Z}_N^*</td>
</tr>
<tr>
<td>ABO-LF</td>
<td>(N^2/4)</td>
<td>DCR</td>
<td>(\log N)</td>
<td>(</td>
<td>\mathbb{Z}_N^*</td>
</tr>
</tbody>
</table>
Generic Construction from HPS

Wee (Eurocrypt 2012): dual HPS ⇒ LTF
- dual HPS: HPS satisfying strong property
- No efficient ABO construction is known
Generic Construction from HPS

Wee (Eurocrypt 2012): dual HPS $\Rightarrow$ LTF
- dual HPS: HPS satisfying strong property
- No efficient ABO construction is known

We show $\text{HPS} \Rightarrow \text{ABO-RLF}$
- exploit algebra property of the underlying SMP
(Algebra) Subset Membership Problem

Task: distinguish $U_X \approx_c U_L$

Solution: \{0, 1\}

\begin{itemize}
  \item SampAll$(\lambda)$
  \item SampYes$(\lambda)$
  \item SampNo$(\lambda)$
  \item SampR$(\lambda)$
\end{itemize}

\[ X \overset{R_L}{\longrightarrow} W \]

Algebraic properties

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

(Algebra) Subset Membership Problem

Task: distinguish $U_X \cong_c U_L$

Solution: \{0, 1\}

Algebra SMP (mild & natural)
- $X$ forms an Abelian group, $L$ forms a subgroup of $X$
- The quotient group $H = X/L$ is cyclic with order $p = |X|/|L|$
(Algebra) Subset Membership Problem

Task: distinguish $U_X \approx_c U_L$  

Solution: $\{0, 1\}$

Algebra SMP (mild & natural)

- $X$ 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 $a \in X\setminus L$ be a generator of $H$, the co-sets $(aL, 2aL, \ldots, (p-1)aL, paL = L)$ constitute a partition of $X$.
2. For each $x \in L$, $ia + x \notin L$ for $1 \leq i < p$
Hash Proof System

- $L \subset X$ — language defined by $R_L$ where SMP holds.
- HPS equips $L \subset X$ with $\text{Gen}$, $\text{Priv}$, $\text{Pub}$.

$$\text{Gen}(\lambda) \rightarrow (pk, sk)$$

s.t. $\alpha(sk) = pk$

$SK \xrightarrow{\alpha \ (\text{projection})} PK$

$X \xrightarrow{\text{SampR}(r)} L \xrightarrow{\Lambda_{sk}(x)} \Pi$

$\text{Priv}(sk, x) \xrightarrow{\text{Pub}(pk, x, w)}$
Hash Proof System

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

$$\text{Gen}(\lambda) \rightarrow (pk, sk)$$

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:

- \( \text{Gen}(\lambda, b^*) \): \((x, w) \leftarrow \text{SampYes}(\lambda)\), output \( ek = -b^*a + x \)
- \( f_{ek, b}(sk) \): 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:

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

**Lemma 4**

Assume \( g_x(sk) := \alpha(sk)||\Lambda_{sk}(x) \) is \( \nu \)-regular for any \( x \notin L \). The above construction is \((\nu, \log|\text{Img}\alpha|)\)-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:

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

**Lemma 4**

Assume $g_x(sk) := \alpha(sk)||\Lambda_{sk}(x)$ is $v$-regular for any $x \notin L$. The above construction is $(v, \log |\text{Img}\alpha|)$-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 $u \leftarrow X$. 


Outline

1 Backgrounds

2 Regular Lossy Functions

3 Constructions of ABO RLFs
   - Concrete Construction
   - Generic Construction

4 Applications of RLFs
   - Leakage-Resilient OWFs
   - Leakage-Resilient MAC
   - Leakage-Resilient CCA-secure KEM
Leakage-Resilient Cryptography

\[ \text{Sign} \quad \text{Dec} \quad x \]

\[ F_{sk} \]

\[ F(sk, x) \]
Leakage-Resilient Cryptography

\[ F_{sk}(x) \]

Leakage proof

leakage attacks (since 1996) invalidate this idealized assumption
Leakage-Resilient Cryptography

Leakage proof

black-box

$F_{sk}$

$F(sk, x)$
Leakage-Resilient Cryptography

$F_{sk}(sk; x)$

leakage prone

$x$

$F(sk, x)$
leakage attacks (since 1996) invalidate this idealized assumption

\[
F_{sk}(x) = F(sk, x)
\]
Leakage attacks (since 1996) invalidate this idealized assumption.
leakage attacks (since 1996) invalidate this idealized assumption
Leakage-Resilient Cryptography

leakage attacks (since 1996) invalidate this idealized assumption

\[ F(\text{sk}, x) \]
leakage attacks (since 1996) invalidate this idealized assumption
Leakage-Resilient Cryptography

leakage attacks (since 1996) invalidate this idealized assumption
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

The normal mode of \((1; 2)\) -RLFs (i.e., LFs) over domain \(f_{0}; 1\) constitutes a family of \(\ell\)-leakage-resilient injective OWFs, for any \(\ell \geq \log n\).
Leakage-Resilient OWFs

\[(f, y^*)\]

Theorem 5

The normal mode of \((1; 1)\)-RLFs (i.e., LFs) over domain \(f_0; 1\) constitutes a family of \(\ell\)-leakage-resilient injective OWFs, for any \(\ell > (\log n)^2\).

\[x^* \leftarrow \{0, 1\}^n\]
\[y^* \leftarrow f(x^*)\]
The normal mode of $(1; R)$-RLFs (i.e., LFs) over domain $f_0; 1$ constitute a family of $f$-leakage-resilient injective OWFs, for any $f$.
Theorem 5

The normal mode of $(1; R)$-RLFs (i.e., LFs) over domain $f_{0;1^n}$ constitutes a family of $\ell$-leakage-resilient injective OWFs, for any $\ell > (\log n)$. 

The diagram illustrates the process:

- $x^* \leftarrow \{0, 1\}^n$
- $y^* \leftarrow f(x^*)$
- $g_i(x^*)$
- $(f, y^*)$
Leakage-Resilient OWFs

Theorem 5

The normal mode of $(1; 0)$-RLFs (i.e., LFs) over domain $f_{0; 1}$ constitutes a family of $\ell$-leakage-resilient injective OWFs, for any $\ell \not= (\log(n))$. 

\[
x^* \xleftarrow{\text{R}} \{0, 1\}^n \\
y^* \leftarrow f(x^*) \\
x =? x^*
\]
Leakage-Resilient OWFs

\[ (f, y^*) \]

\[ g_i \]

\[ g_i(x^*) \]

\[ x \]

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

\[ y^* \leftarrow f(x^*) \]

\[ x = \_x^* \]

Theorem 5

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

1. **Setup:** $\mathcal{CH}$ generates $f \leftarrow \text{RLF.GenNormal}(\lambda)$, picks $x^* \leftarrow \{0, 1\}^n$ and sends $(f, y^* = f(x^*))$ to $A$.

2. **Leakage queries:** $A \rightarrow g_i$, $\mathcal{CH}$ responds with $g_i(x^*)$.

3. **Invert:** $A$ outputs $x$ and wins if $x = x^*$.

$$\text{Adv}_A(\lambda) = \Pr[S_0]$$

**Game 1:** same as Game 0 except that:

4. **Setup:** $\mathcal{CH}$ generates $\boxed{f \leftarrow \text{RLF.GenLossy}(\lambda)}$.

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

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

- By the parameter choice, $\tilde{H}_\infty(x^*|(y^*, \text{leak})) \geq \omega(\log \lambda) \Rightarrow \Pr[S_1] \leq \text{negl}(\lambda)$

  even w.r.t. unbounded adversary
Leakage-Resilient MAC
Leakage-Resilient MAC

\[ \text{Setup}(\lambda) \Rightarrow (pp, k) \]

\[ \Rightarrow \text{Verify}(k; m; t) = 1 \]

Strong unforgeability can be relaxed in several ways:

- One-time: \( A \) only makes one tag query

- Selective: \( A \) commits the target message before seeing
Leakage-Resilient MAC

\[ pp \quad (pp, k) \leftarrow \text{Setup}(\lambda) \]

Strong unforgeability can be relaxed in several ways:

One-time: A only makes one tag query

Selective: A commits the target message before seeing
Leakage-Resilient MAC

\[
\begin{align*}
pp & \quad \quad \quad \quad \quad (pp, k) \leftarrow \text{Setup}(\lambda) \\
mi & \quad \quad \quad \quad \quad t_i \leftarrow \text{Tag}(k, mi)
\end{align*}
\]
Leakage-Resilient MAC

\[
\begin{align*}
pp & \leftarrow \text{Setup}(\lambda) \\
\leftarrow m_i & \quad \leftarrow t_i \leftarrow \text{Tag}(k, m_i) \\
& \leftarrow g_i
\end{align*}
\]

Strong unforgeability can be relaxed in several ways:

- One-time: A makes only one tag query
- Selective: A commits the target message before seeing
Leakage-Resilient MAC

\[
\begin{align*}
(pp, k) & \leftarrow \text{Setup}(\lambda) \\
pp & \quad m_i \\
& \quad t_i \leftarrow \text{Tag}(k, m_i) \\
g_i & \quad g_i(k)
\end{align*}
\]

Strong unforgeability can be relaxed in several ways:

- One-time: \(A\) only makes one tag query
- Selective: \(A\) commits the target message before seeing
Leakage-Resilient MAC

\[ \text{Setup}(\lambda) \]

\[ (pp, k) \leftarrow \text{Setup}(\lambda) \]

\[ t_i \leftarrow \text{Tag}(k, m_i) \]

\[ g_i \]

\[ g_i(k) \]

\[ (m^*, t^*) \]

\[ \text{Vefy}(k, m^*, t^*) = 1 \]

\[ (m^*, t^*) \neq (m_i, t_i) \]
Leakage-Resilient MAC

\[(pp, k) \leftarrow \text{Setup}(\lambda)\]

\[m_i \quad t_i \leftarrow \text{Tag}(k, m_i)\]

\[g_i \quad g_i(k) \quad (m^*, t^*) \quad \Box \]

\[\text{Vefy}(k, m^*, t^*) = 1\]
\[(m^*, t^*) \neq (m_i, t_i)\]

Strong unforgeability can be relaxed in several ways:
Leakage-Resilient MAC

Strong unforgeability can be relaxed in several ways:
- One-time: $A$ only makes one tag query

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

$pp \quad m_i \quad t_i \leftarrow \text{Tag}(k, m_i) \quad g_i \quad g_i(k) \quad (m^*, t^*) \quad \text{Vefy}(k, m^*, t^*) = 1 \quad (m^*, t^*) \neq (m_i, t_i)$
Strong unforgeability can be relaxed in several ways:

- One-time: $A$ only makes one tag query
- Selective: $A$ commits the target message before seeing $pp$
Construction

Ingredient

$(v, \tau)$-ABORLF
Construction

Ingredient
$(v, \tau)$-ABORLF

KeyGen

\[ ek \leftarrow \text{ABORLF.Gen}(\lambda, 0^d) \]
\[ k \leftarrow \{0, 1\}^n \]
Construction

Ingredient

\((v, \tau)\)-ABORLF

KeyGen

\[ e_k \leftarrow \text{ABORLF.Gen}(\lambda, 0^d) \]
\[ k \leftarrow \{0, 1\}^n \]

Tag

\[ m \]

\[ t \leftarrow f_{ek,m}(k) \]

\( k \) - input

\( m \) - branch

\( t \) - output
Construction

Ingredient

\((v, \tau)\)-ABORLF

KeyGen

\[ ek \leftarrow \text{ABORLF.Gen}(\lambda, 0^d) \]

\[ k \leftarrow \{0, 1\}^n \]

Tag

\[ t \leftarrow f_{ek,m}(k) \]

Vefy

\[ t =? f_{ek,m}(k) \]

\( m \rightarrow \)

\( m \rightarrow \)
Theorem 6

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

Game 0: (real game)

1. Setup: $A \leftrightarrow m^*$, $CH$ generates $ek \leftarrow \text{ABORLF.Gen}(\lambda, 0^d)$, picks $k \leftarrow \{0, 1\}^n$, computes $t^* \leftarrow f_{ek,m^*}(k)$ and then sends $(ek, t^*)$ to $A$.

2. Leakage queries: $A \leftrightarrow g_i$, $CH$ responds with $g_i(k)$.

3. Forge: $A \rightarrow (m, t)$ and wins if $m \neq m^* \land t = f_{ek,m}(k)$.

$\text{Adv}_A(\lambda) = \text{Pr}[S_0]$
**Game 1:** same as Game 0 except that

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

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

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

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

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

\[ (pk, sk) \leftarrow \text{Setup}(\lambda) \]
Leakage-Resilient CCA-secure KEM

\[ \text{pk} \leftarrow \text{Setup}(\lambda) \]

\[ (\text{pk}, \text{sk}) \leftarrow \text{Setup}(\lambda) \]
Leakage-Resilient CCA-secure KEM

\[ \begin{align*}
\text{Setup} : & \quad (pk, sk) \leftarrow \text{Setup}(\lambda) \\
\text{Encap} : & \quad k_0, R_0, \ldots, R_f \leftarrow \text{Encap}(pk) \\
\text{Decaps} : & \quad k_i \leftarrow \text{Decaps}(sk, c_i) \\
\end{align*} \]
Leakage-Resilient CCA-secure KEM

\[ \text{pk} \xleftarrow{} \text{Setup}(\lambda) \]

\[ k_i \leftarrow \text{Decaps}(sk, c_i) \]

\[ g_i \]
Leakage-Resilient CCA-secure KEM

\[
\begin{align*}
(pk, sk) & \leftarrow \text{Setup}(\lambda) \\
\text{Setup} & (pk; sk) \\
\text{Encap} & (pk) \\
\text{Decaps} & (sk, c_i) \\
\rightarrow & \\
k_i & \leftarrow \text{Decaps}(sk, c_i) \\
g_i & \\
g_i(sk) & \\
\rightarrow & \\
\rightarrow &
\end{align*}
\]
Leakage-Resilient CCA-secure KEM

\[
(pk, sk) \leftarrow \text{Setup}(\lambda)
\]

\[
\begin{align*}
&pk \\
&c_i \\
&k_i \leftarrow \text{Decaps}(sk, c_i) \\
&g_i \\
&g_i(sk) \\
&(c^*, k^*_0) \\
&(c^*, k^*_\beta)
\end{align*}
\]

\[
\begin{align*}
&(c^*, k^*_0) \leftarrow \text{Encap}(pk) \\
&k^*_1 \leftarrow R K \\
&\beta \leftarrow \{0, 1\}
\end{align*}
\]
Leakage-Resilient CCA-secure KEM

\[(pk, sk) \leftarrow \text{Setup}(\lambda)\]

\[\begin{align*}
  &pk \\
  &c_i \\
  &k_i \leftarrow \text{Decaps}(sk, c_i) \\
  &g_i \\
  &g_i(sk) \\
  &c^*, k^*_\beta \\
  &\beta'
\end{align*}\]

\[\begin{align*}
  &g_i(k^*_0) \leftarrow \text{Encap}(pk) \\
  &k^*_{1} \overset{R}{\leftarrow} K \\
  &\beta \overset{R}{\leftarrow} \{0, 1\} \\
  &\beta' = \beta
\end{align*}\]
Leakage-Resilient CCA-secure KEM

\[ (pk, sk) \leftarrow \text{Setup}(\lambda) \]

\[ (c^*, k_0^*) \leftarrow \text{Encap}(pk) \]

\[ k_1^* \overset{R}{\leftarrow} K \]

\[ \beta \overset{R}{\leftarrow} \{0, 1\} \]

\[ \beta' = \beta \]

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

Ingredients
- HPS
- ABORLF
- strong extractor
Construction

Ingredients

HPS
ABORLF
strong extractor

KeyGen

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

Ingredients
HPS
ABORLF
strong extractor

KeyGen

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

Encaps

\((x, w) \leftarrow \text{SampYes}(\lambda)\)

\(\pi \leftarrow \text{Pub}(pk, x, w)\)

\(s \leftarrow \{0, 1\}^d\)

\(t \leftarrow f_{ek, x||s}(\pi)\)

\(k \leftarrow \text{ext}(\pi, s)\)

use \(\pi\) to:
- derive \(k\) &
- authenticate \(x||s\)
Construction

Ingredients
HPS
ABORLF
strong extractor

KeyGen

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

Encaps

$(x, w) \leftarrow \text{SampYes}(\lambda)$
$\pi \leftarrow \text{Pub}(pk, x, w)$
$s \leftarrow \{0, 1\}^d$
$t \leftarrow f_{ek, x\|s}(\pi)$
$k \leftarrow \text{ext}(\pi, s)$

Decaps

use $\pi$ to:
- derive $k$
- authenticate $x\|s$

$t = ? f_{ek, x\|s}(\pi)$
$k \leftarrow \text{ext}(\pi, s)$ or $\bot$

Decaps

$\pi \leftarrow \text{Priv}(sk, x)$
Theorem 7

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

Game 0: (real game)

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

\[ \text{Adv}_A(\lambda) = \Pr[S_0] - 1/2 \]
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 $ek \leftarrow \text{ABORLF.Gen}(\lambda, x^* || s^*)$.

Hidden lossy branch $\Rightarrow |\Pr[S_2] - \Pr[S_1]| \leq \text{negl}(\lambda)$
Game 1: \( \mathcal{CH} \) samples \((x^*, w^*)\) and \(s^*\) at Setup.

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

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

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

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

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 \(ek \leftarrow \text{ABORLF.Gen}(\lambda, x^* || s^*)\).

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

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

Correctness of HPS \( \Rightarrow \Pr[S_3] = \Pr[S_2]\).

Game 4: \( \mathcal{CH} \) samples \(x^*\) via \(\text{SampNo}\) rather than \(\text{SampYes}\).

\(\text{SMP} \Rightarrow |\Pr[S_4] - \Pr[S_3]| \leq \text{negl}(\lambda)\)
Game 1: $\mathcal{CH}$ samples $(x^*, w^*)$ and $s^*$ at Setup.

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

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

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

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

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

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

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

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

$$|\Pr[S_5] - \Pr[S_4]| \leq \Pr[E]$$
To calculate $\Pr[E]$, it suffice to bound $\tilde{H}_\infty(t|\text{view})$.

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

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

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

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

$$\Pr[E] \leq \text{negl}(\lambda)$$
Game 6: $\mathcal{CH}$ samples $k_0^* \leftarrow \{0, 1\}^\kappa$ rather than $k_0^* \leftarrow \text{ext}(\Lambda_{sk}(x^*))$. Next, we analysis $\Delta[\text{view}_5, \text{view}_6]$.

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

$$\Delta[\text{view}_5, \text{view}_6] \leq \epsilon_2/2 \leq \text{negl}(\lambda)$$

Putting all the above together, $\text{Adv}_A(\lambda) = \text{negl}(\lambda)$. 
Significance

Universal\textsubscript{1} HPS + ABO-RLF $\Rightarrow$ LR-CCA KEM

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

Universal_1 HPS + ABO-RLF ⇒ LR-CCA KEM

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

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

\[ \text{eDDH} \quad \text{DCR} \quad \text{(ABO)-RLFs} \]
Conclusion

\begin{itemize}
\item eDDH
\item DCR
\item \((ABO)\)-RLFs
\end{itemize}

\begin{itemize}
\item Algebra SMP
\item HPS
\end{itemize}
Conclusion

(ABO)-RLFs

eDDH  DCR

LR OWF

Algebra SMP

HPS
Conclusion

(ABO)-RLFs

eDDH \rightarrow \text{LR OWF}

DCR \rightarrow \text{LR MAC}

Algebra SMP

HPS
Conclusion

(ABO)-RLFs

- eDDH
- DCR
- LR OWF
- LR MAC
- LR-CCA PKE

Algebra SMP

HPS
Thanks for Your Attention!

Any Questions?