Parameter-Hiding Order Revealing Encryption

Cong Zhang

Rutgers University

Joint work with David Cash, Feng-Hao Liu, Adam O'Neill and Mark Zhandry

CNN tech

BUSINESS CULTURE GADGETS FUTURE STARTUPS

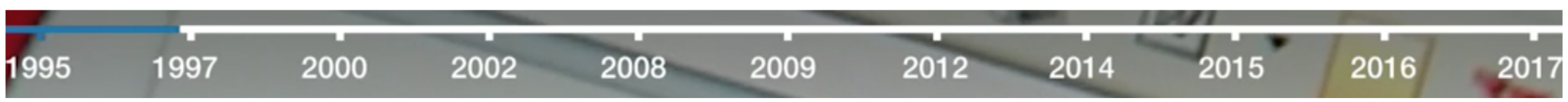
Cyber-Safe

Every single Yahoo account was hacked - 3 billion in all

by Selena Larson @selenalarson

October 4, 2017: 6:36 AM ET

Recommend 14K



"because it would hurt Yahoo's ability to index and search messages to provide new user services"
~Be Jeff Bonfort(Yahoo SVP)

Order-Revealing Encryption (ORE) [BCLO'09]

In this talk, message domain is always some integer interval $[M] = \{0,1,\dots,M\}$

Order Revealing Encryption (ORE): Three algorithms:

$(sk, pk) \leftarrow \mathsf{Gen}$ outputs a secret key and a public "comparison" key

$c \leftarrow \mathsf{E}_{sk}(x)$ outputs ciphertext

$b \leftarrow \mathsf{Compare}(pk, c_1, c_2)$ outputs a bit

Correctness: $x_1 \leq x_2 \Leftrightarrow \mathsf{Compare}(E_{sk}(x_1), E_{sk}(x_2)) = 1 \, (w.h.p.)$

Decryption: Not required to be useful, but always possible using comparison.

Order Preserving Encryption (OPE): Is an ORE scheme where ciphertexts are also integers and comparison is simple integer comparison.

Correctness: $x_1 < x_2 \Leftrightarrow E_{sk}(x_1) < E_{sk}(x_2) \, (w.h.p.)$

- If encryption is deterministic, then OPE encryption is an increasing function
- pk is emptystring in OPE, and often in ORE as well

ORE in Encrypted Databases

First Name	Last Name	Zip	D.O.B.
		…04	Aug 18 1982
			Feb 12 1988
			Jan 22 1970
			May 30 1968

deployed by:

CipherCloud
Trust in the Cloud™

Symantec.

skyhigh

prototyped by:

Microsoft

SAP

academic projects:

CryptDB[PRZB'11]

- E… …nge from **x** to **y**, we rewrite th…

query bet… **10000** an…

plaintext column

Zip
08k065
26861e
2hc36e
48eb42

query between **26861e** and **2hc36e**

encrypted column

Two Flavors of ORE: Ideal and Leaky

Encrypted column:

Zip Code
68k065
48eb42
26861e
01c36e

only order revealed

ideal ORE:

Zip Code
4
3
2
1

- only known way achieved via iO, multilinear maps [BLRSZZ'15]
- interactive protocols [PLZ'13,KS'14,Ker'15]

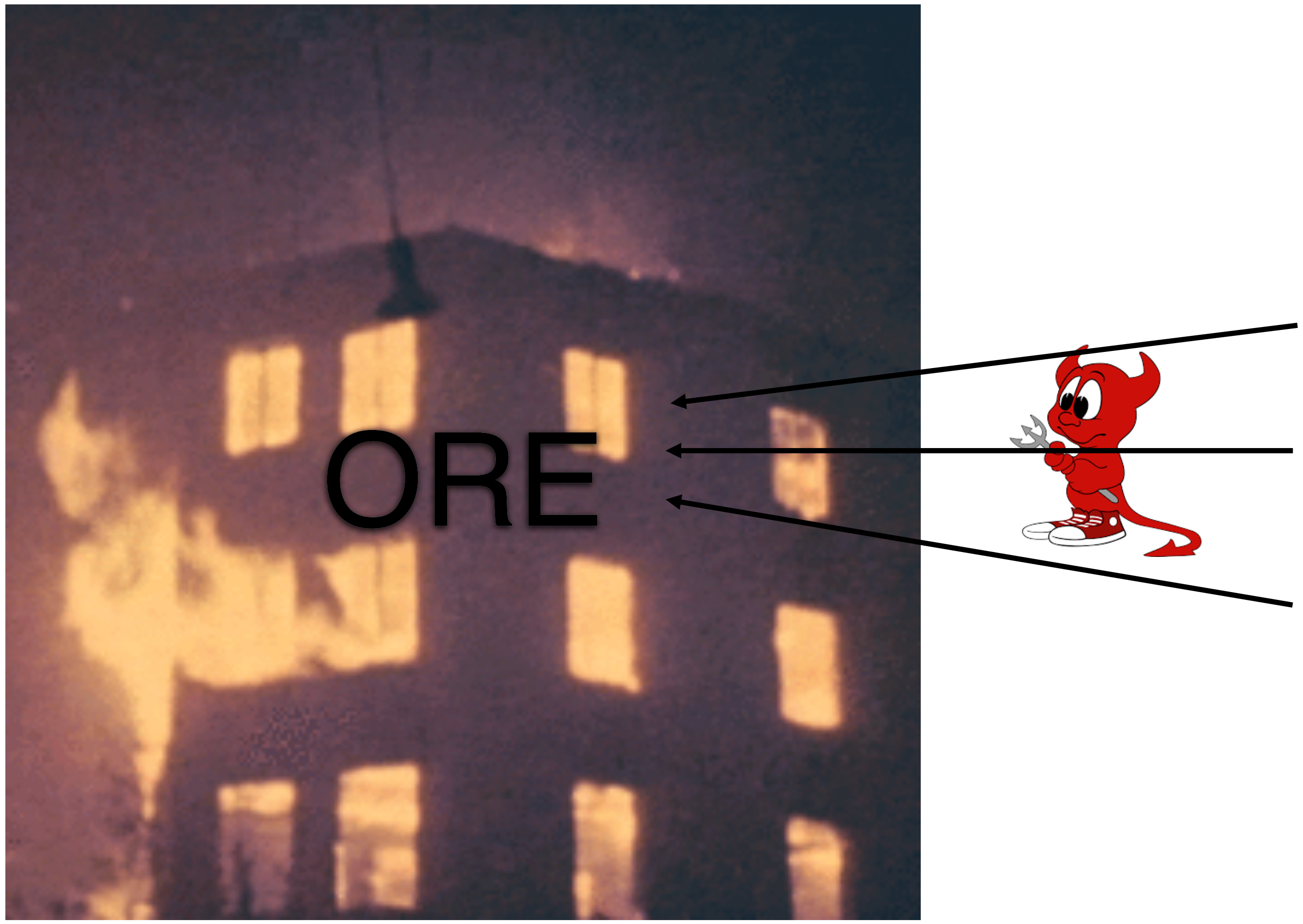
leaky ORE (example):

Zip Code
98211
10761
10065
10028

order + extra info revealed

- fast, block cipher based constructions [BCLO'09, CLWW'16]
- extra info includes: some plaintext bits, statistics, or more.

Known attacks on ORE



Inference attacks [NKW'15]

Non-crossing attacks [GSBNR'17]

Correlation attacks [DDC'16, BGCRS'18]

Security does not imply Privacy!!!

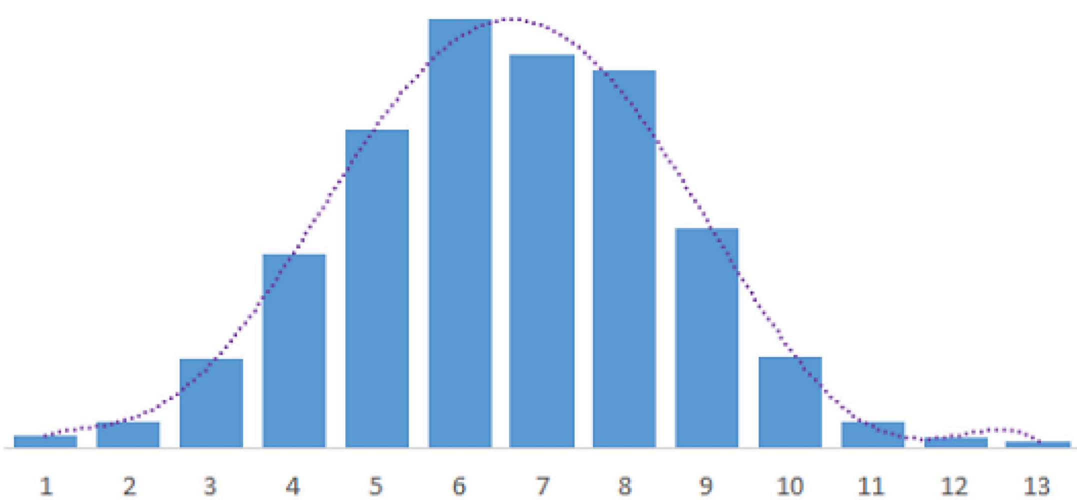
Semantically meaningful privacy notion?

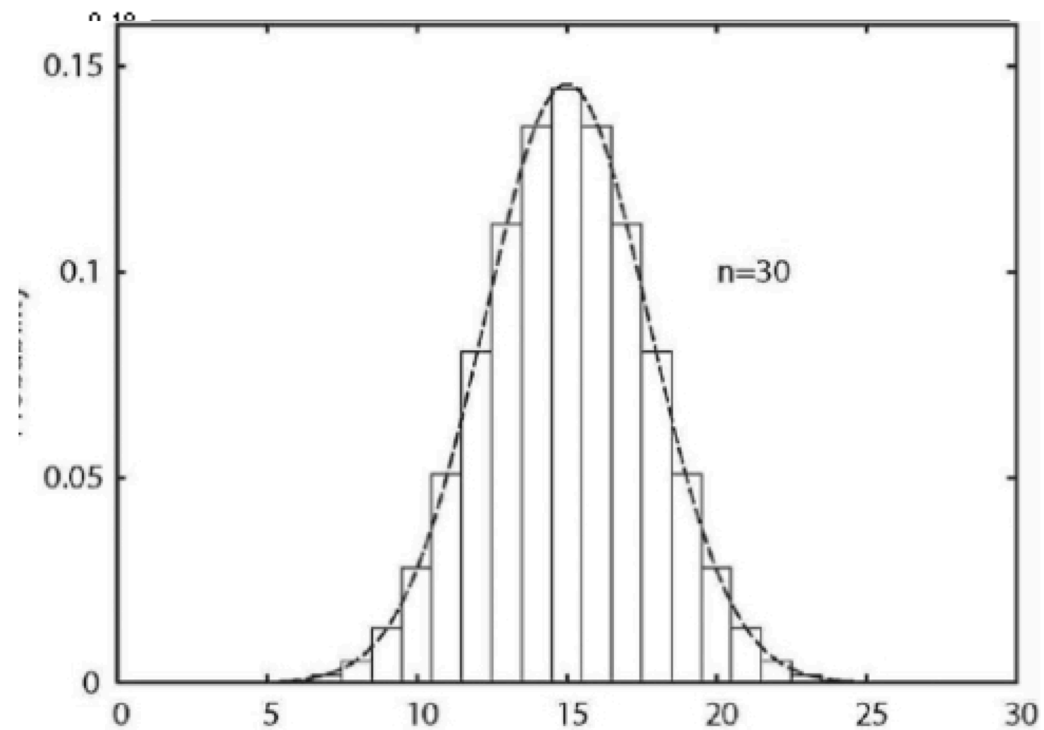
Privacy Notions

- Distribution-Hiding
- Parameter-Hiding

Privacy Notions

- **Distribution-Hiding**
- **Parameter-Hiding**
- Parameter-Hiding





$$(m_1, \dots, m_q) \qquad (m'_1, \dots, m'_q)$$

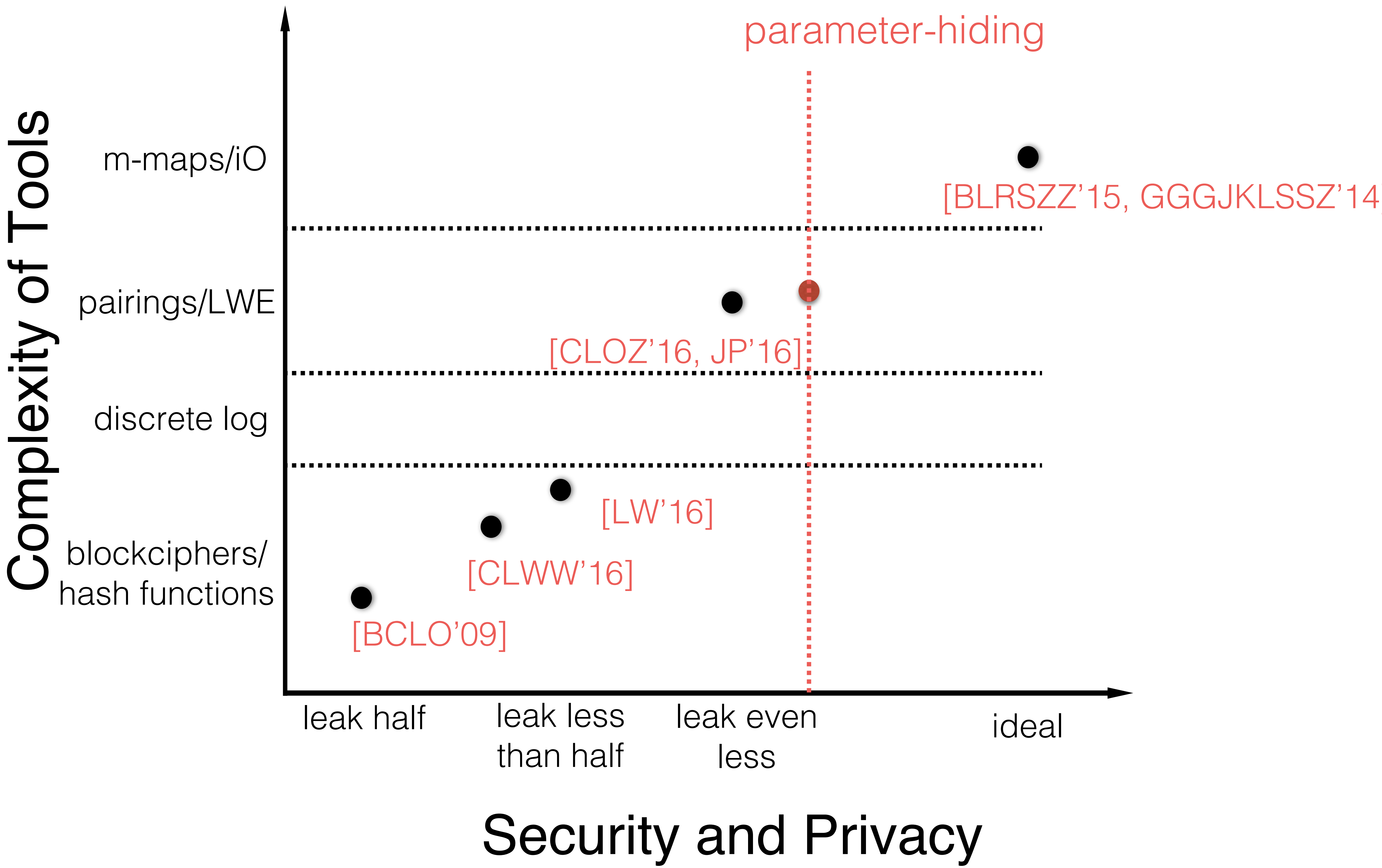
Can be achieved *only* by Ideal ORE

“Mean” and “Variance” are hidden

Why does parameter-hiding matter?

- Parameter-Hiding is the current strongest privacy notion achieved efficiently;
- It captures potential real world application
    - Adversary only have the curve information of message distribution;
    - Statistics (mean, variance etc.) are important.

This work: Construct PH-ORE based on bi-linear maps



Main Result

Theorem: Assuming bilinear map, it is possible to construct parameter-hiding ORE for any “smooth” distribution D, provided the scaling term is large enough.

- large scaling term means D has high min-entropy;
- smoothness is define as having bounded derivative except constant points

Outline

1. ORE Security Definition
2. EP-MSDB-secure Constructions and Leakage Profile
3. High-level Intuition
4. Bonus: Impossibility results on OPE
5. Conclusion

Outline

1. **ORE Security Definition**
2. EP-MSDB-secure Constructions and Leakage Profile
3. High-level Intuition
4. Bonus: Impossibility results on OPE
5. Conclusion

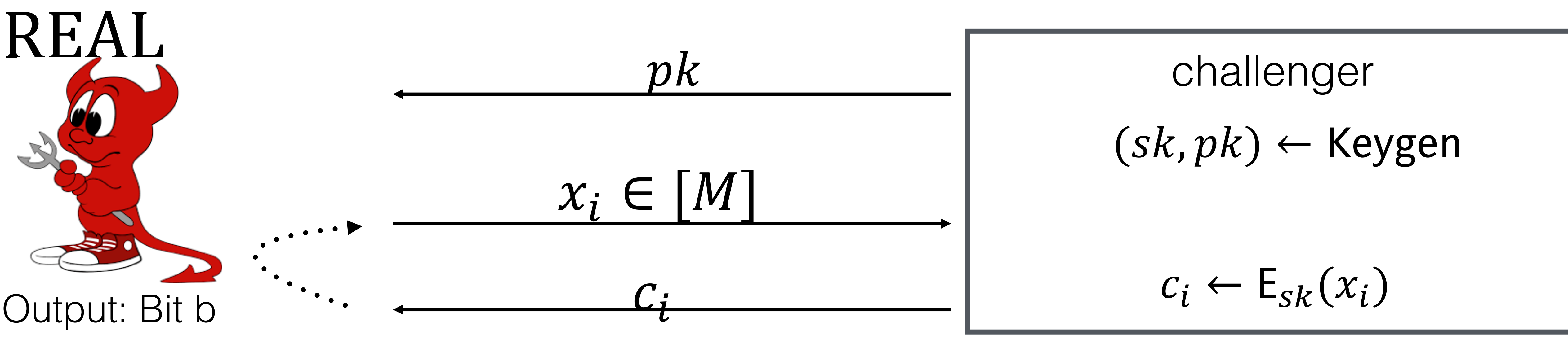
ORE Security Definition

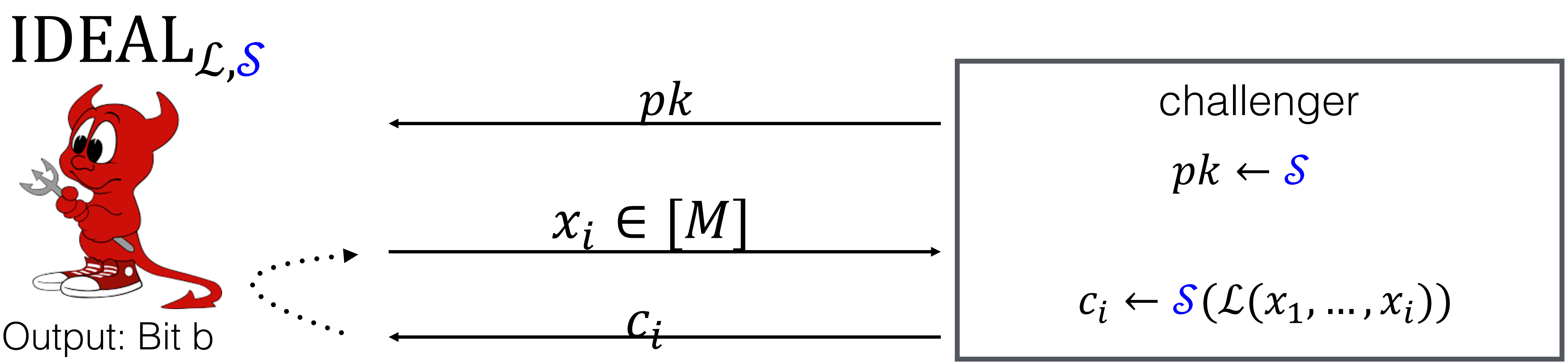
$$\textit{Def. An ORE scheme}\,\Pi\,\textit{is}\,\mathcal{L}\textit{-secure if}\,\forall \mathcal{A}\,\exists \mathcal{S}:$$

$$\Pr[\mathcal{A}\,\textit{outputs}\,1\,\textit{in}\,\mathrm{REAL}] \approx \Pr[\mathcal{A}\,\textit{outputs}\,1\,\textit{in}\,\mathrm{IDEAL}_{\mathcal{L},\mathcal{S}}]$$

- "Leakage function" $\mathcal{L}$ and simulator $\mathcal{S}$ are stateful, randomized

Formal security model games





Outline

1. ORE Security Definition
2. **EP-MSBD—secure Constructions and Leakage Profile**
3. High-level Intuition
4. Bonus: Impossibility results on OPE
5. Conclusion

New Leakage Profile

Equality Pattern of Most Significant Differ-Bit (EP-MSDB)

Inspired by MSDB leakage profile [CLWW'16]

- The order for every pair of plaintexts.
- For every pair of ciphertexts $c = \mathsf{E}_{sk}(x), c' = \mathsf{E}_{sk}(x')$,scheme reveal

$$\mathrm{MSDB}(x, x') = min\{i: x_i \neq x'_i\}$$

plaintexts	leaked bits
x_i = 1110110	x_i = x11xxxx
x_j = 1101000	x_j = x10xxxx
x_k = 1001100	x_k = x0xxxxx

EP-MSDB Leakage Profile

- The order for every pair of plaintexts.
- For every pair of ciphertexts $c = \mathsf{E}_{sk}(x), c' = \mathsf{E}_{sk}(x'), c'' = \mathsf{E}_{sk}(x'')$

$$\mathrm{MSDB}(x, x') \stackrel{?}{=} \mathrm{MSDB}(x, x'')$$

Example

x = 0 0 0 0 1 0 1 0 1 0 1

x’ = 0 0 1 0 1 1 1 0 1 0 0

x’’= 0 0 1 1 1 1 1 1 1 1 1

$\Rightarrow$

2x = 0 0 0 1 0 1 0 1 0 1 0

$\mathrm{MSDB}(x, x') = \mathrm{MSDB}(x, x'')$

2x’ = 0 1 0 1 1 1 0 1 0 0 0

$\mathrm{MSDB}(x', x) \neq \mathrm{MSDB}(x', x'')$

2x’’ = 0 1 1 1 1 1 1 1 1 1 0

$\mathrm{MSDB}(x'', x) \neq \mathrm{MSDB}(x'', x')$

MSDB construction [CLWW'16]

- Ingredient: PRF $F: \mathcal{K} \times \{0,1\}^* \to \{0,1\}^\lambda \setminus \{1^\lambda\}$

1. **Key generation**: Output PRF key as secret, and no public key

$$(K, \perp) \leftarrow \text{Keygen}$$

2. **Encryption**: Input x ∈ [M], $E_{sk}(x)$ works as follows:
   - $Parse\ x\ into\ bits\ x_1 x_2 \ldots x_m, where\ m = \log M$
   - $For\ i = 1, \ldots, m: c_i \leftarrow (F_K(x_1, \ldots, x_{i-1}) + x_i mod\ 2^\lambda)$
   - $Output\ (c_1, c_2, \ldots, c_m) \in \{0,1\}^{m\lambda}$

Comparison Algorithm for MSDB scheme [CLWW'16]

3. **Comparison**: On input $(c_1, \ldots, c_m), (c'_1, \ldots, c'_m)$

$c_1 = F_K(\varepsilon) + x_1$	$c_2 = F_K(x_1) + x_2$	$c_3 = F_K(x_1 x_2) + x_3$
$c'_1 = F_K(\varepsilon) + x'_1$	$c'_2 = F_K(x'_1) + x'_2$	$c'_3 = F_K(x'_1 x'_2) + x'_3$
↑ Equal?	↑ Equal?	↑ Equal?

…

- At first index i where $x_i \neq x'_i$ either $c_i = c'_i + 1$ or $c'_i = c_i + 1$
- Determine which is larger by checking cases

EP-MSDB construction

- Ingredient : PRF $F: \mathcal{K} \times \{0,1\}^* \to \{0,1\}^\lambda \setminus \{1^\lambda\}$

property-preserving hash $\mathcal{H}: \mathrm{sk} \times \{0,1\}^\lambda \to \mathit{Group\ elements}$

1. **Key generation**: Output PRF key as secret, and no public key

$$(K, \perp) \leftarrow \mathrm{Keygen}$$

2. **Encryption**: Input x ∈ [M], $E_{sk}(x)$ works as follows:

   - $\mathit{Parse}\ x\mathrm{I}\ \mathit{nto\ bits}\ x_1 x_2 \ldots x_m, \mathit{where}\ m = \log M$
   - $\mathit{For}\ i = 1, \ldots, m: c_i \leftarrow (F_K(x_1, \ldots, x_{i-1}) + x_i \mathit{mod} 2^\lambda)$
   - $\mathit{Output}\ (c_1, c_2, \ldots, c_m) \in \{0,1\}^{m\lambda}$

Property-preserving Hash

Consists of two algorithms: Hash $\mathcal{H}$ and Test $\mathcal{T}$

$$\mathcal{T}(\mathcal{H}(x),\mathcal{H}(y)) = \begin{cases} 1 & \text{if } y = x + 1 \\ 0 & \text{Otherwise} \end{cases}$$

Scheme:

$$\mathcal{H}_{\mathsf{sk}}(x) = (g_1^{r_1}, g_1^{r_1 \cdot PRF_{\mathsf{sk}}(x)}, g_2^{r_2}, g_2^{r_2 \cdot \underline{PRF_{\mathsf{sk}}(x+1)}})$$

If $y = x + 1$

$$\mathcal{H}_{\mathsf{sk}}(y) = (g_1^{r_1}, g_1^{r_1 \cdot \underline{PRF_{\mathsf{sk}}(y)}}, g_2^{r_2}, g_2^{r_2 \cdot PRF_{\mathsf{sk}}(y+1)})$$

EP-MSDB construction

- Ingredient : PRF $F: \mathcal{K} \times \{0,1\}^* \rightarrow \{0,1\}^\lambda \setminus \{1^\lambda\}$

  property-preserving hash $\mathcal{H}: \mathrm{sk} \times \{0,1\}^\lambda \rightarrow \textit{Group elements}$

1. **Key generation**: Output PRF key as secret, and public key is the test key

   $$((\mathrm{tk}, \mathrm{sk})) \leftarrow \mathrm{Keygen}$$
   $$(K, \perp) \leftarrow \mathrm{Keygen}$$

   PK SK

2. **Encryption**: Input x ∈ [M], $E_{sk}(x)$ works as follows:

   - $\textit{Parse } x \textit{ into bits } x_1 x_2 \ldots x_m, \textit{ where } m = \log M$
   - $\textit{For } i = 1, \ldots, m: c_i \leftarrow (F_K(x_1, \ldots, x_{i-1}) + x_i \bmod 2^\lambda)$
   - $\textit{Output } (c_1, c_2, \ldots, c_m) \in \{0,1\}^{m\lambda}$ / $\textit{Output } (\mathcal{H}(c_1), \ldots, \mathcal{H}(c_m))$

Comparison EP-MSDB scheme

3. **Comparison**: On input $(C_1, \ldots, C_m), (C'_1, \ldots, C'_m)$

   Identity index (i, j) such that: either $\mathcal{T}(C_i, C'_j) = 1$ or $\mathcal{T}(C'_i, C_j) = 1$

   Determine which is larger by checking cases.

Thm: Under SXDH assumption, ∏ is EP-MSDB-secure.

Outline

1. ORE Security Definition
2. EP-MSDB-secure Constructions and Leakage Profile
3. **High-level Intuition**
4. Bonus: Impossibility results on OPE
5. Conclusion

High-level Intuition

Observations on EP-MSDB leakage profile

$$(m_1, \ldots, m_q) \in [0, 2^\ell)$$

periodicity by addition

$$\Rightarrow \mathcal{L}(m_1, \ldots, m_q) = \mathcal{L}(m_1 + 2^\ell, \ldots, m_q + 2^\ell)$$

If we only hide "mean", we can add a random shift:

$$\overline{\mathsf{Enc}}(m) = \mathsf{Enc}(m + \beta), \beta \xleftarrow{\$} [0, 2^\ell)$$

we need find an alternative periodicity

High-level Intuition

Additional observation

periodicity by multiplication

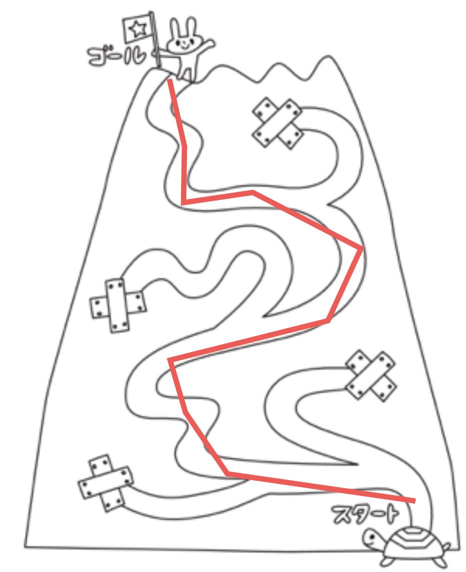
$$(m_1, \ldots, m_q) \in [0, 2^{\ell})$$

$$\Rightarrow \mathcal{L}(m_1, \ldots, m_q) = \mathcal{L}(2m_1, \ldots, 2m_q)$$

applying the same trick

for hiding variance $\overline{\mathsf{Enc}}(m) = \mathsf{Enc}(\alpha m)$

for hiding both $\overline{\mathsf{Enc}}(m) = \mathsf{Enc}(\alpha m + \beta)$



15 pages puzzle

α is sampled from log-uniform distribution on $[2^{\gamma}, 2^{\gamma+1})$

β is sampled from uniform distribution on $[0, 2^{\gamma|m|+1})$

Outline

1. ORE Security Definition
2. EP-MSDB-secure Constructions and Leakage Profile
3. High-level Intuition
4. **Bonus: Impossibility results on OPE**
5. Conclusion

Bonus: Impossibility results of OPE

Ideal ORE $\Rightarrow$ EP-MSDB-secure ORE $\Rightarrow$ PH ORE

- There does not exist non-interactive ideal OPE. [BCLO'09]
- There does not exist non-interactive EP-MSDB-secure OPE. [CLOZ'16]
- This work: There does not exist non-interactive PH OPE.

Conclusion and Open Problems

- Propose two semantically meaningful privacy notions for ORE: distribution-hiding and parameter hiding;
- Construct PH-ORE scheme from an EP-MSDB-secure ORE;
- Build EP-MSDB-secure ORE from bilinear maps.

---

1. Any scheme against adversary with good estimate of message distribution, which still preserving range query? (In progress)
2. Construct PH-ORE based on cryptographic groups?

Thank you!