Parameter-Hiding Order-Revealing Encryption without Pairings

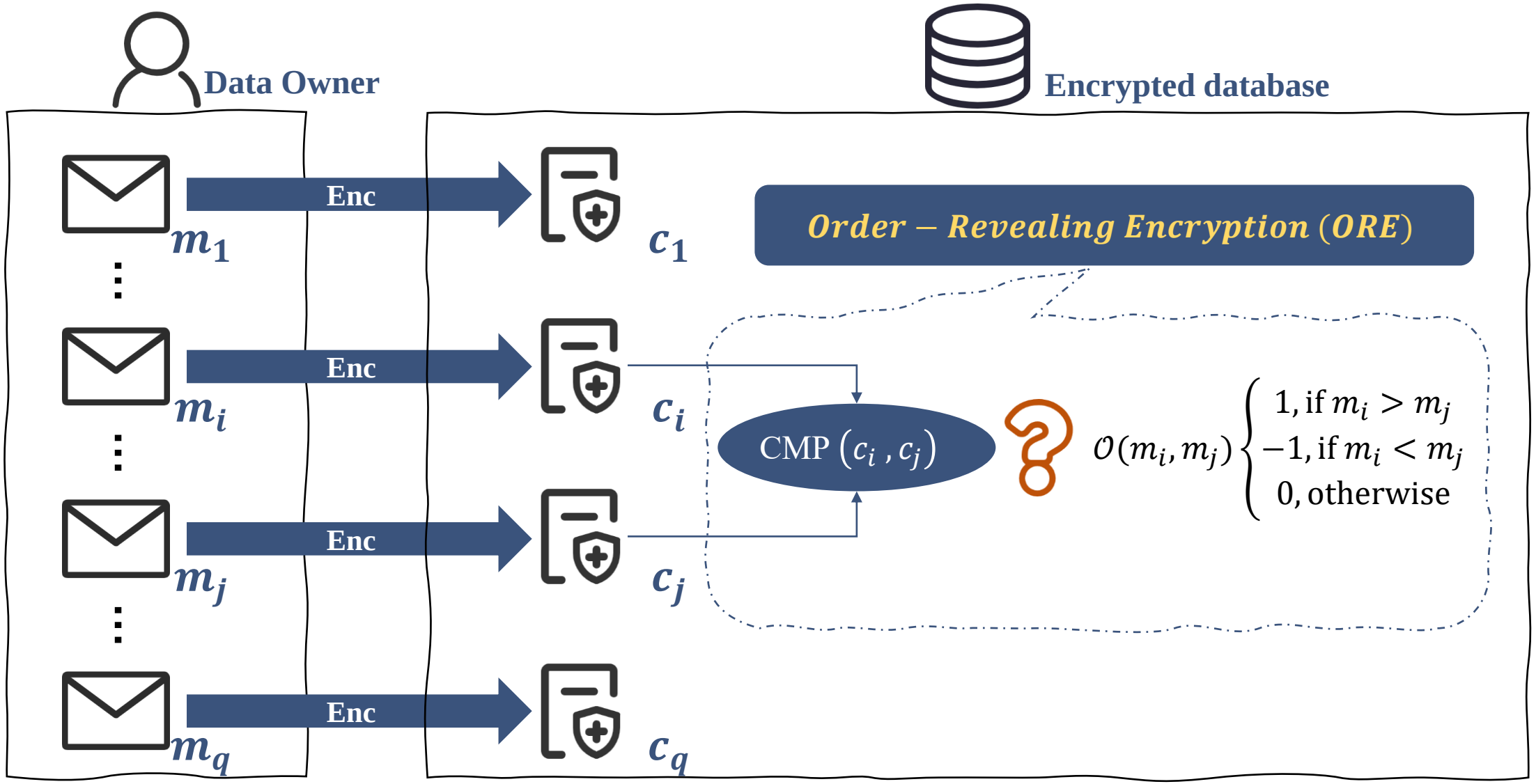
Cong Peng

Wuhan University

Join work with Rongmao Chen, Yi Wang, Debiao He, Xinyi Huang

PKC 2024 April 2024

Setting: Range query on encrypted database



On the Leakage of ORE

- $\mathcal{L}_0$ - Ideal leakage

$$\mathcal{L}_0(m_1, \cdots, m_q) \coloneqq \left(\forall 1 \le i, j \le q, \mathbf{1}(m_i < m_j)\right)$$

- $\mathcal{L}_1$ - Smooth CLWW leakage

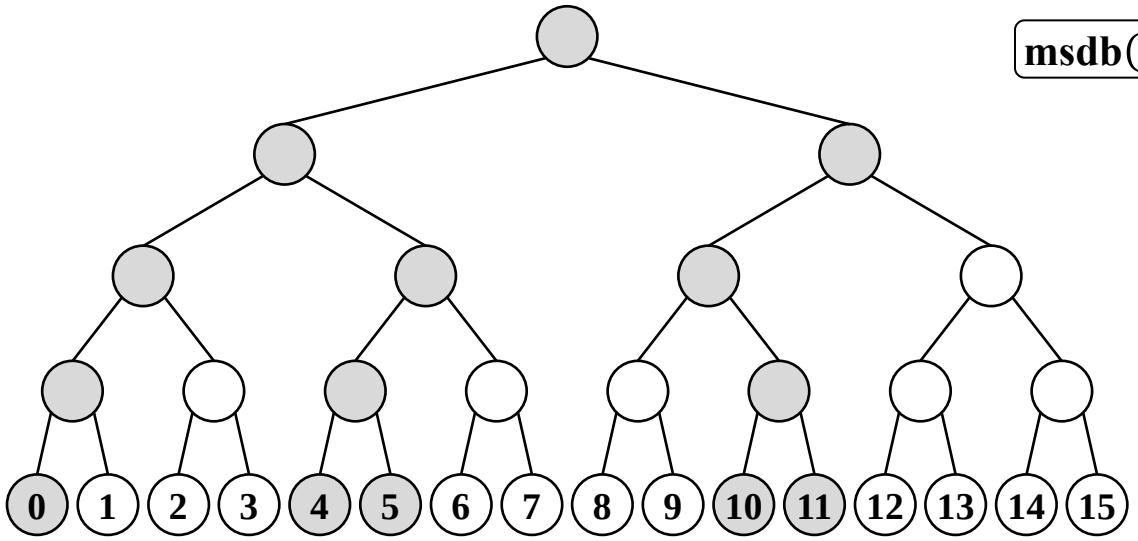
$$\mathcal{L}_1(m_1, \cdots, m_q) \coloneqq \left(\forall 1 \le i, j, k \le q, \mathbf{1}(m_i < m_j), \mathbf{1}(\mathbf{msdb}(m_i, m_j) = \mathbf{msdb}(m_i, m_k))\right)$$

- $\mathcal{L}_2$ - CLWW leakage [CLWW16,]

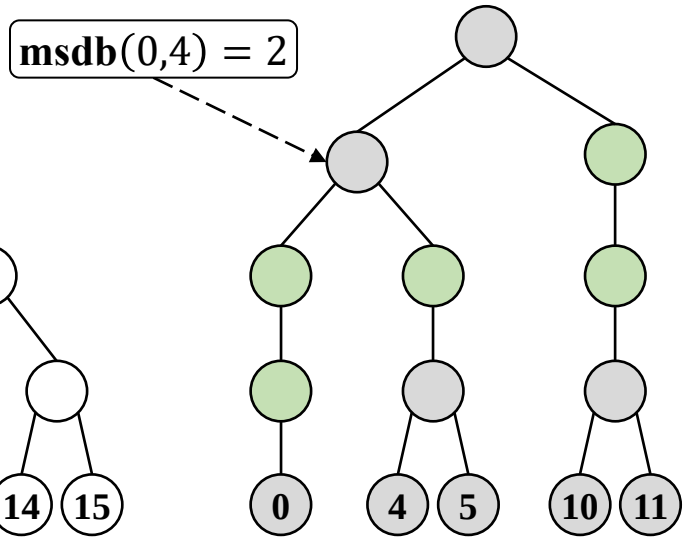
$$\mathcal{L}_2(m_1, \cdots, m_q) \coloneqq \left(\forall 1 \le i, j \le q, \mathbf{1}(m_i < m_j), \mathbf{msdb}(m_i, m_j)\right)$$

msdb : most significant differing bit

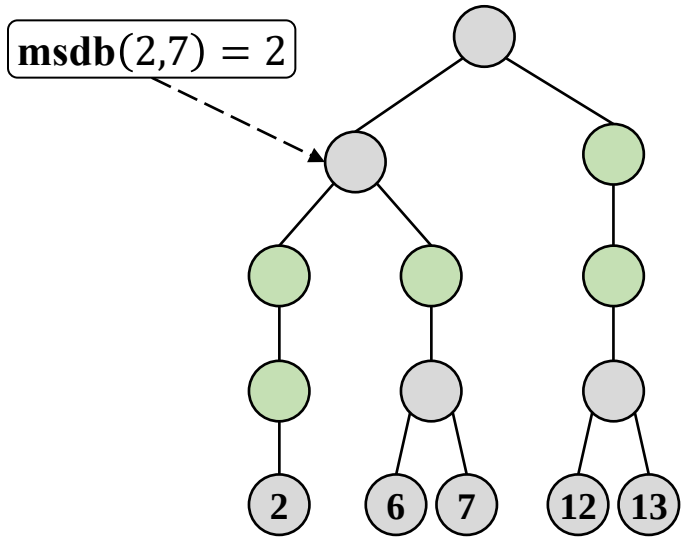
About $\mathcal{L}_2$ - CLWW leakage



Plaintext space $\{0,1\}^4$



CLWW leakage
w.r.t. {0,4,5,10,11}



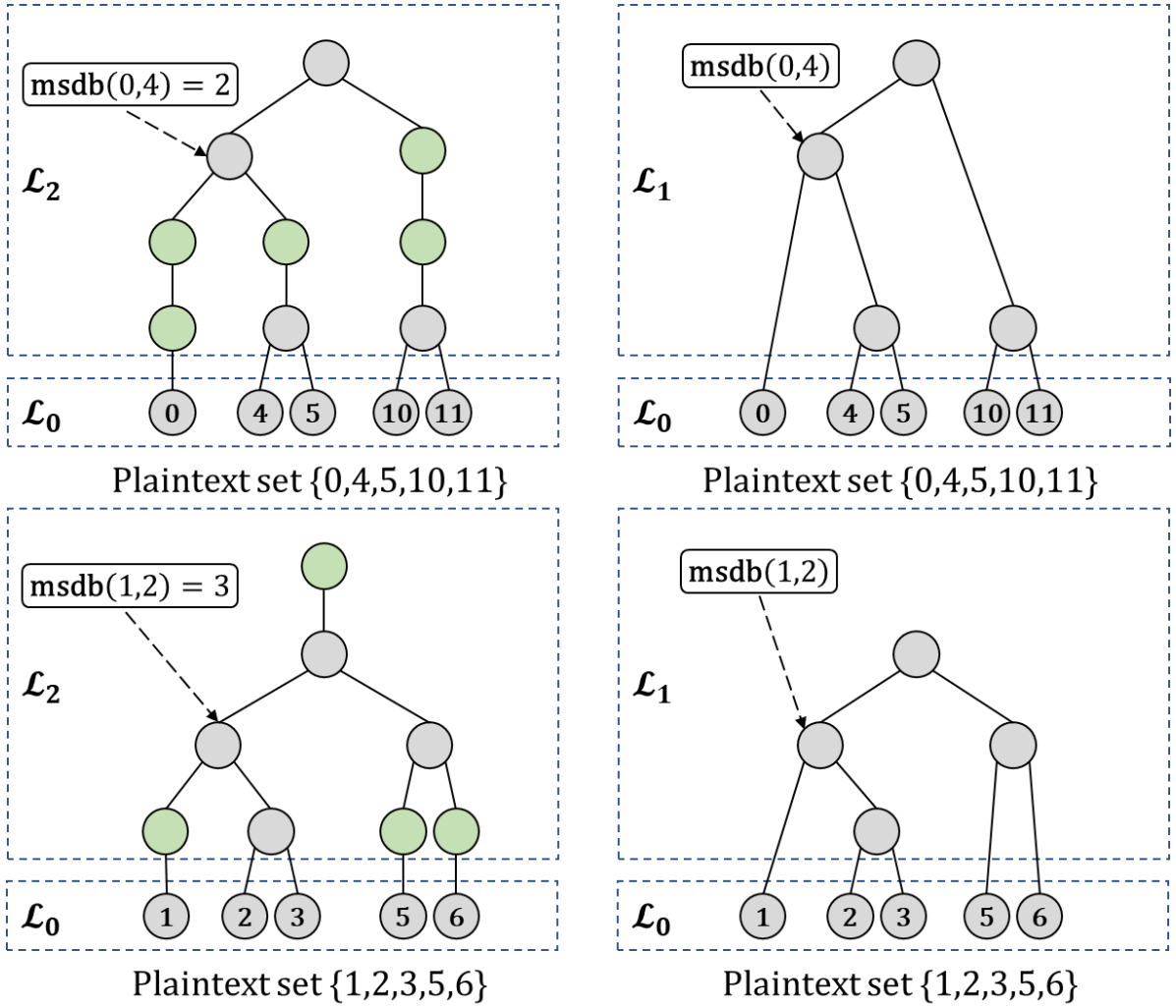
CLWW leakage
w.r.t. {2,6,7,12,13}

Note that for some plaintexts such as {2, 6, 7, 12, 13}, it would also have equivalent subtree w.r.t. {0, 4, 5, 10, 11}.

!!! The ciphertext that $m_i = 4$ has the bit form "01-0", where "-" indicates the unknown bits.

Ignored node

About $\mathcal{L}_1$ - Smooth CLWW leakage



Considering the leakage $\mathcal{L}_1$ which leaks the equality pattern of **msdb**:

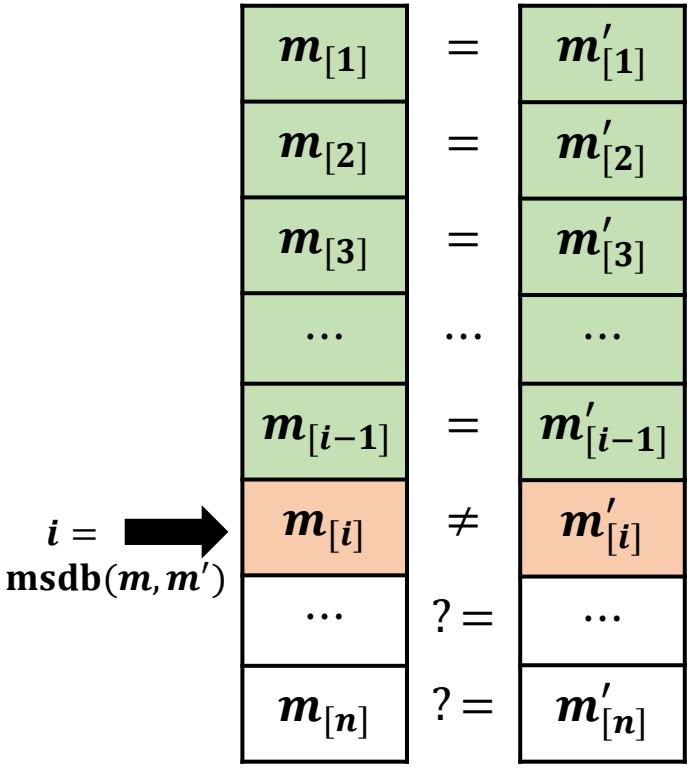
- The comparator can infer additional information

  i.e., $\mathbf{msdb}(0,4) < \mathbf{msdb}(5,11)$

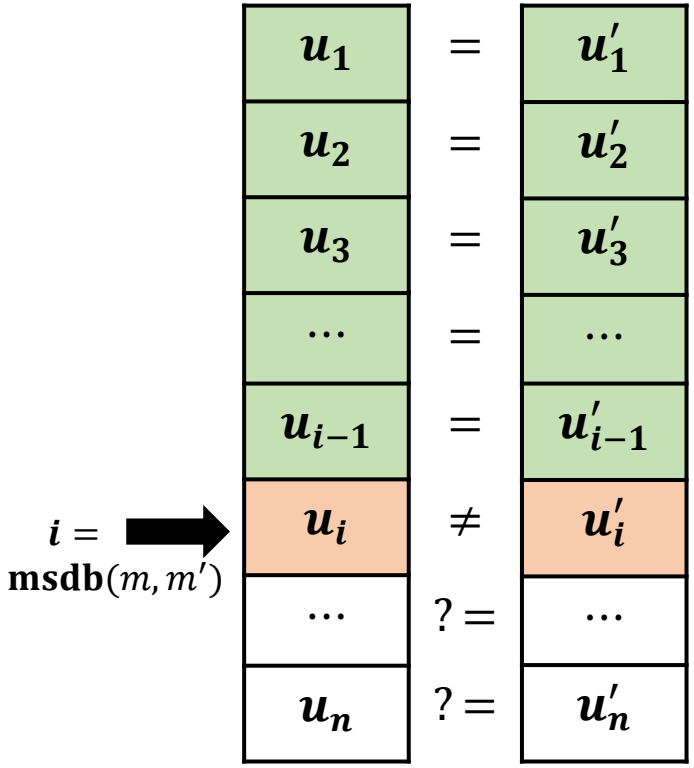
- Green nodes in subtrees are unknown to the comparator as the positions of msdb are not determined.
- Also, the position of gray nodes can be moved up or down.
- So, one can see that **{0, 4, 5, 10, 11}** and **{1, 2, 3, 5, 6}** leak the same information under $\mathcal{L}_1$.

ORE Technique——Bit-wise encryption

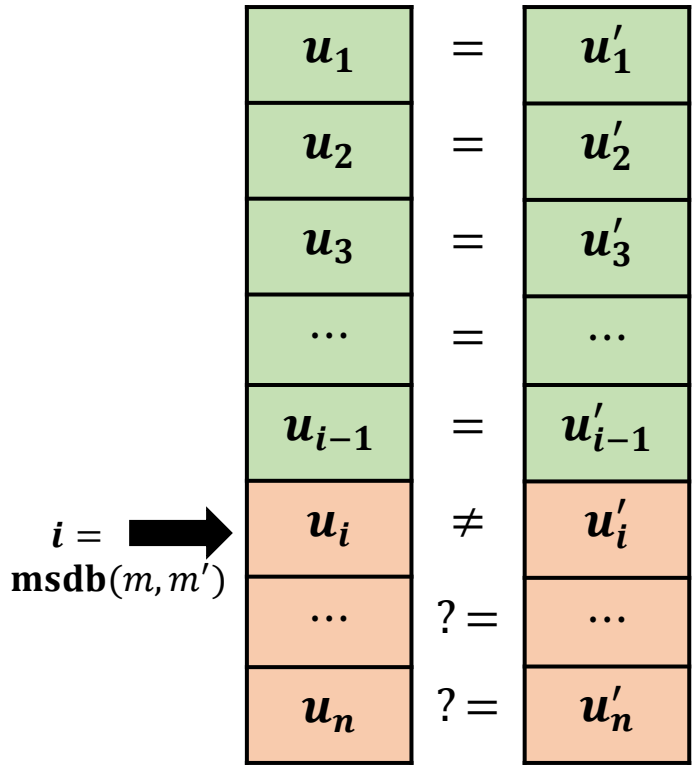
Original Message: m and m'



Bit-wise encryption: $\{u_i\}$ and $\{u'_i\}$



$$u_i = \mathrm{PRF}\left(s, m_{[:i-1]}\right) + m_i \quad \textit{mod } 3$$

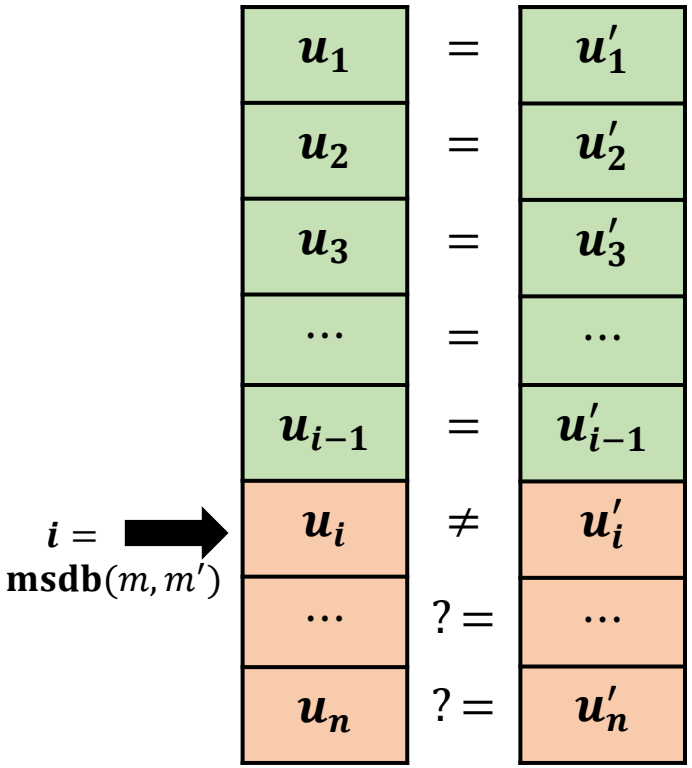


$$u_i = \mathrm{PRF}\left(s, m_{[:i-1]}\right) + m_i \quad \textit{mod } M\ (M \geq 2^{\lambda})$$

Equal Not equal May equal

ORE Technique——Random permutation

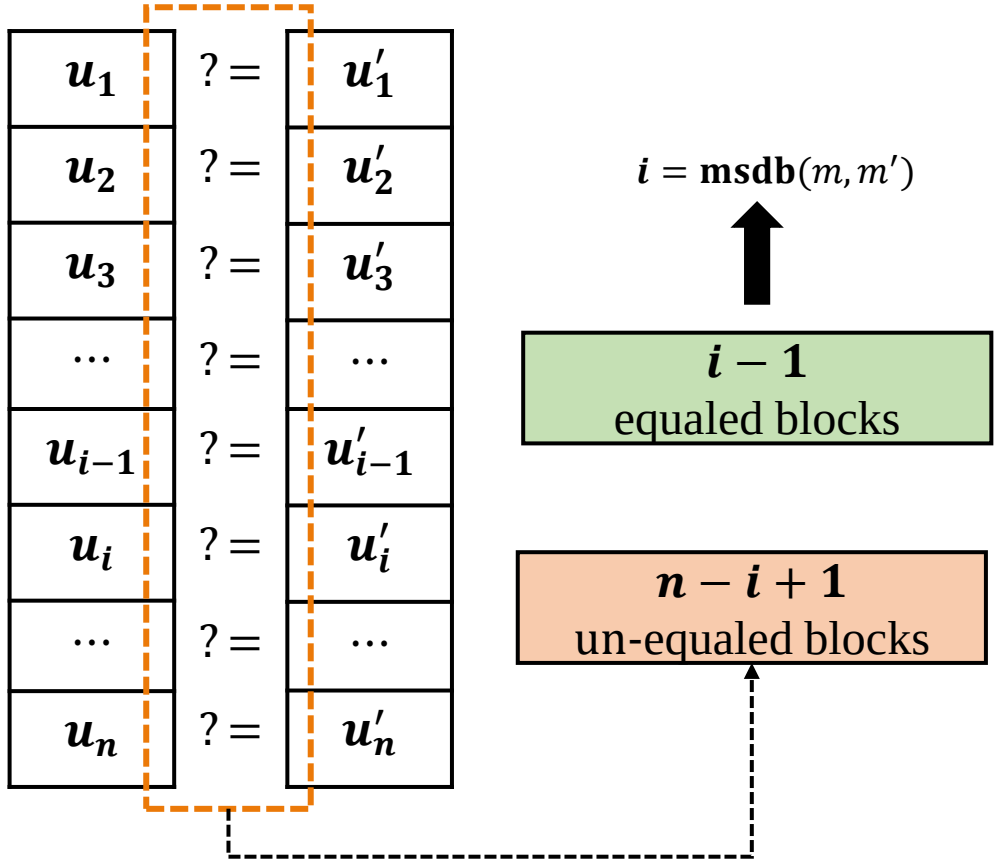
Bit-wise encryption: $\{u_i\}$ and $\{u'_i\}$



$$u_i = \mathcal{E}(s, m, \pi(i))$$

$$u'_i = \mathcal{E}(s, m, \pi'(i))$$

Random Permutation: π and π': $[n] \to [n]$



$$\mathcal{E}(s, m, i) = \text{PRF}\left(s, i || m_{[:i-1]} || 0_{[i:]}\right) + m_i \bmod M \ (M \geq 2^{\lambda})$$

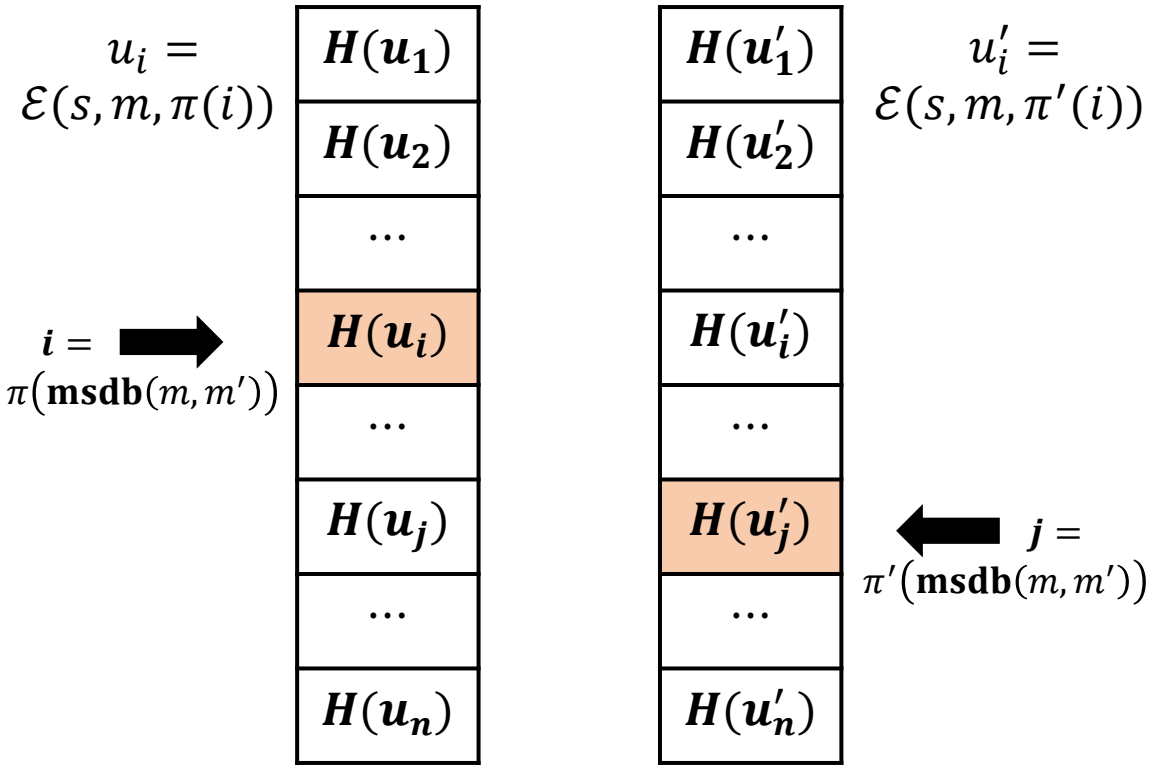
$$\mathcal{L}_2(m_1, \cdots, m_q) := (\forall 1 \leq i, j \leq q, \mathbf{1}(m_i < m_j), \text{msdb}(m_i, m_j)),$$

Equal Not equal May equal

ORE Technique——Predicate relation

Random commitment: $H(x)$

$$\mathcal{L}_1(m_1, \cdots, m_q) := (\forall 1 \le i, j, k \le q, \mathbf{1}(m_i < m_j), \mathbf{1}(\mathsf{msdb}(m_i, m_j) = \mathsf{msdb}(m_i, m_k)))$$



- If $m > m'$, we have $u_i = u'_j + 1$
- If $m < m'$, we have $u_i = u'_j - 1$

Question

1. Randomness
   Make the number of equaled blocks secret?
   Need the function H output random value, i.e.
   $$H(x) \neq H(y) \text{ if } x = y.$$
2. Property-Preserving
   Preserve a testable predicate relation $\mathcal{P}$, such as
   $$x = y \pm 1$$
   Need an algorithm satisfying
   $$\mathrm{Test}\big(H(x), H(y)\big) = \mathcal{P}(x, y)$$
3. Collision Resistance
   The probability
   $$\Pr\big[\mathrm{Test}\big(H(x), H(y)\big) \neq \mathcal{P}(x, y)\big] < \mathbf{negl}(\lambda)$$
   is negligible.

Only $\mathcal{P}(u_i, u'_j) = 1$, otherwise 0

ORE Technique——Parameter-Hiding ORE (at Asiacrypt 2018)

New Property Preserving Hash (PPH) construction: $\mathcal{H}(\mathsf{hk}, x) = (g^{r_1}, g^{r_1 \cdot F(k,x)}, \hat{g}^{r_2}, \hat{g}^{r_2 \cdot F(k,x+1)})$.

$g^{r_{10}}$	$g^{r_{10} \cdot F(k,u_1)}$	$\hat{g}^{r_{11}}$	$\hat{g}^{r_{11} \cdot F(k,u_1+1)}$	$\neq$	$g^{r'_{10}}$	$g^{r'_{10} \cdot F(k,u'_1)}$	$\hat{g}^{r'_{11}}$	$\hat{g}^{r'_{11} \cdot F(k,u'_1+1)}$
…	…	…	…	$\neq$	…	…	…	…
$g^{r_{i0}}$	$g^{r_{i0} \cdot F(k,u_i)}$	$\hat{g}^{r_{i1}}$	$\hat{g}^{r_{i1} \cdot F(k,u_i+1)}$	$\neq$	$g^{r'_{i0}}$	$g^{r'_{i0} \cdot F(k,u'_i)}$	$\hat{g}^{r'_{i1}}$	$\hat{g}^{r'_{i1} \cdot F(k,u'_i+1)}$
…	…	…	…	$\neq$	…	…	…	…
$g^{r_{j0}}$	$g^{r_{j0} \cdot F(k,u_j)}$	$\hat{g}^{r_{j1}}$	$\hat{g}^{r_{j1} \cdot F(k,u_j+1)}$	$\neq$	$g^{r'_{j0}}$	$g^{r'_{j0} \cdot F(k,u'_j)}$	$\hat{g}^{r'_{j1}}$	$\hat{g}^{r'_{j1} \cdot F(k,u'_j+1)}$
…	…	…	…	$\neq$	…	…	…	…
$g^{r_{n0}}$	$g^{r_{n0} \cdot F(k,u_n)}$	$\hat{g}^{r_{n1}}$	$\hat{g}^{r_{n1} \cdot F(k,u_n+1)}$	$\neq$	$g^{r'_{n0}}$	$g^{r'_{n0} \cdot F(k,u'_n)}$	$\hat{g}^{r'_{n1}}$	$\hat{g}^{r'_{n1} \cdot F(k,u'_n+1)}$

- ElGamal Encryption
- Bit-wise randomness
- **No efficient map between g and $\hat{g}$**

Find a tuple satisfying $e(A_1, D_2) = e(A_2, D_1)$

$\mathcal{L}_1(m_1, \cdots, m_q) := (\forall 1 \le i, j, k \le q, \mathbf{1}(m_i < m_j), \mathbf{1}(\mathsf{msdb}(m_i, m_j) = \mathsf{msdb}(m_i, m_k)))$

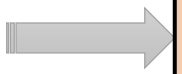
Complexity

$O(n^2)$, for check whether $m > m'$;
$O(n^2)$, for check whether $m < m'$;

ORE Technique——First Optimization

$O(n^2)$ times pairing operations are decided by bit-wise randomness in encryption

$g^{r_{10}}$	$g^{r_{10}\cdot F(k,u_1)}$
…	…
$g^{r_{i0}}$	$g^{r_{i0}\cdot F(k,u_i)}$
…	…
$g^{r_{j0}}$	$g^{r_{j0}\cdot F(k,u_j)}$
…	…
$g^{r_{n0}}$	$g^{r_{n0}\cdot F(k,u_n)}$



g^{r_0}	$g^{r_0\cdot F(k,u_1)}$	$\hat{g}^{r_1}$	$\hat{g}^{r_1\cdot F(k,u_1+1)}$	$\neq$	$g^{r'_0}$	$g^{r'_0\cdot F(k,u'_1)}$	$\hat{g}^{r'_1}$	$\hat{g}^{r'_1\cdot F(k,u'_1+1)}$
	…		…	$\neq$		…		…
	$g^{r_0\cdot F(k,u_i)}$		$\hat{g}^{r_1\cdot F(k,u_i+1)}$	$\neq$		$g^{r'_0\cdot F(k,u'_i)}$		$\hat{g}^{r'_1\cdot F(k,u'_i+1)}$
	…		…	$\neq$		…		…
	$g^{r_0\cdot F(k,u_j)}$		$\hat{g}^{r_1\cdot F(k,u_j+1)}$	$\neq$		$g^{r'_0\cdot F(k,u'_j)}$		$\hat{g}^{r'_1\cdot F(k,u'_j+1)}$
	…		…	$\neq$		…		…
	$g^{r_0\cdot F(k,u_n)}$		$\hat{g}^{r_1\cdot F(k,u_n+1)}$	$\neq$		$g^{r'_0\cdot F(k,u'_n)}$		$\hat{g}^{r'_1\cdot F(k,u'_n+1)}$

The security reduce to whether vector ElGamal encryption with the same randomness is IND-CCA1 security?

$\left(g^{r_0}, g^{r_0\cdot F(k,u_1)}, \dots, g^{r_0\cdot F(k,u_i)}, \dots, g^{r_0\cdot F(k,u_n)}\right)$ VS $\left(g^{r'_0}, g^{r'_0\cdot F(k,u'_1)}, \dots, g^{r'_0\cdot F(k,u'_i)}, \dots, g^{r'_0\cdot F(k,u'_n)}\right)$

Motivating Question & Our Contributions

WUHAN UNIVERSITY

Is it possible to design a parameter − hiding order − revealing encryption scheme without pairings?

Contributions

- Design **a generic PPH construction from a special type of identification schemes** which is of additional property called map-invariance} and prove its restricted-chosen-input secure w.r.t. the predicate

$$\mathcal{P}(x, y) = \pm 1 \text{ if and only if } x = y \pm 1.$$

- Provide **a generic ORE construction** with smoothed CLWW leakage $\mathcal{L}_1$ **from identification protocols**, and prove that the proposed scheme is $\mathcal{L}_1$ non-adaptive-simulation secure with respect to the predicate

$$\mathcal{O}(x, y) = \pm 1 \text{ if and only if } x > y \text{ or } x < y.$$

- Instanced with Schnorr identification, we presented an ORE scheme, which can be converted to parameter-hiding ORE via Cash et al.'s framework. The results demonstrate that our scheme outperforms the current parameter-hiding ORE at the same security level, in which the **ciphertext length reduced more than 31.25%**, the encryption efficiency increased by **nearly 2.6 times** and the comparison efficiency increased by **more than 3 times**.

Attractive Properties of Schnorr Identification

$$\mathsf{pk} := (g, y = g^x) \in \mathbb{G}^2$$

Prover

$\mathsf{sk} := x \leftarrow_{\$} \mathbb{Z}_p^*$

$r_0 \in \mathbb{Z}_p^*$

Verifier

$\text{commitment } \mathsf{cmt} := w = g^{r_0}$ →

$\text{challenge } \xi \in \mathbb{Z}_p^*$ ←

$\text{response } \mathsf{rsp} := z = r_0 - \xi \cdot x$ →

checks whether $w = g^z \cdot y^{\xi}$

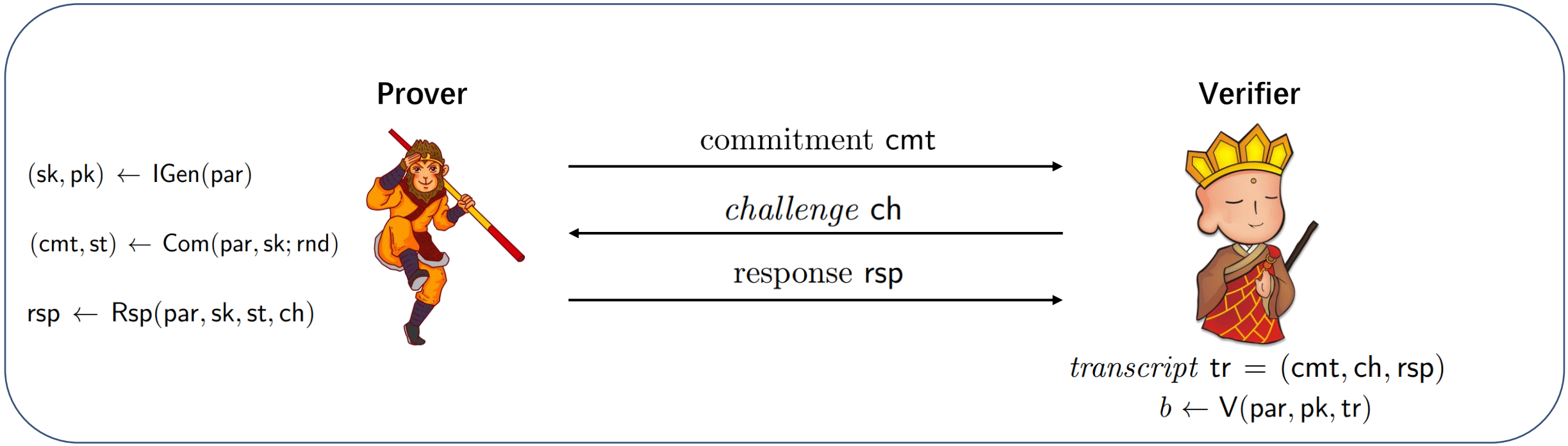
Basic Properties

- Completeness
- Soundness
- Special honest verifier zero-knowledge

Attractive Properties

- Commitment-Independency
- Commitment-Recoverability
- Commitment-Augmentability
- Response-Indistinguishability

Attractive Properties of Schnorr Identification

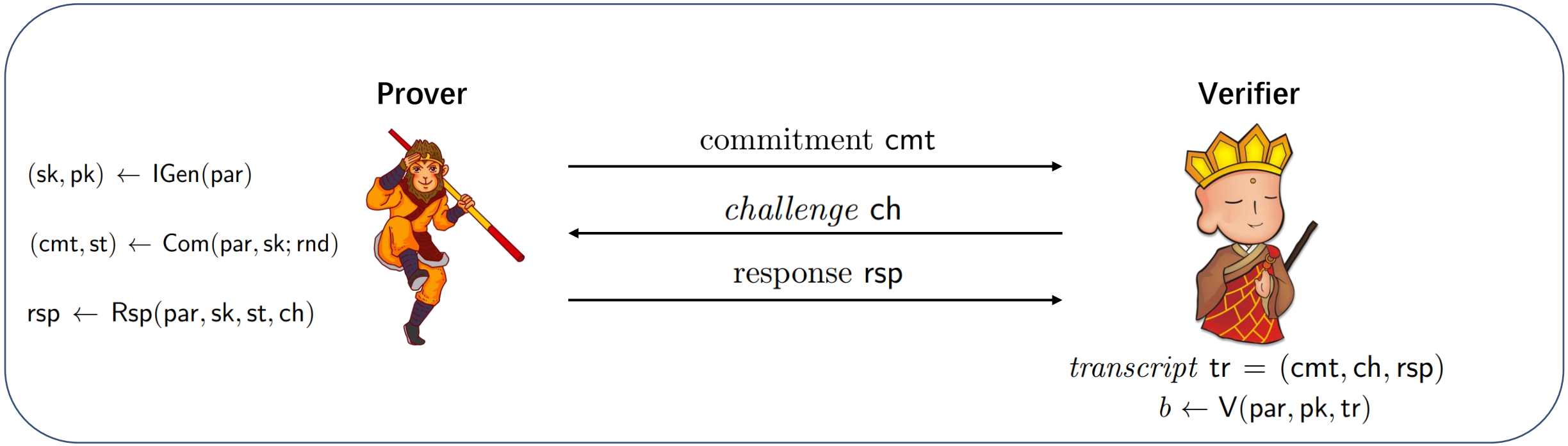


- ***Commitment – Independency***
  - $(\mathsf{cmt}, \mathsf{st})$ *are uniquely determined by* par *and* rnd, *and independent of the key-pair* sk *and* pk*;*
  - $\mathsf{Com}_1(\mathsf{par}, \mathsf{sk}; \mathsf{rnd}) = \mathsf{Com}_1(\mathsf{par}, \mathsf{sk}; \mathsf{rnd}')$ *if and only if* $\mathsf{rnd} = \mathsf{rnd}'$

Schnorr Instance

- g^{r_0} *and* r_0 are determined by g *and* $|\mathbb{G}|$
- $g^{r_0} = g^{r_0'} \Leftrightarrow r_0 = r_0'$

Attractive Properties of Schnorr Identification



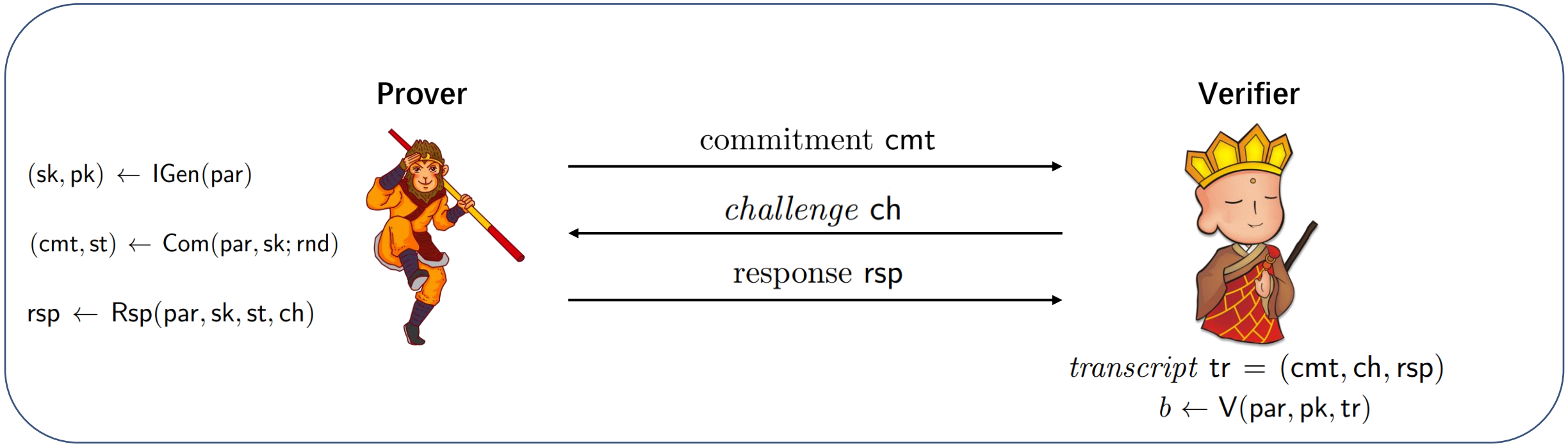
- ***Commitment – Recoverability***
  - *The unique commitment* cmt *can be publicly computed by a commitment recovery algorithm* Rec*, i.e.,*

$$\mathsf{cmt} = \mathsf{Rec}(\mathsf{par}, \mathsf{pk}, \mathsf{ch}, \mathsf{rsp}).$$

Schnorr Instance

- The commitment $\mathbf{cmt} = g^{r_0}$
- $\mathbf{Rec(par, pk, ch, rsp)} = g^z y^{\xi} = g^{r_0}$
  - $\mathbf{sk} = x$, $\mathbf{pk} = (g, y = g^x)$
  - $\mathbf{ch} = \xi$
  - $\mathbf{rsp} = z = r_0 - \xi \cdot x$

Attractive Properties of Schnorr Identification



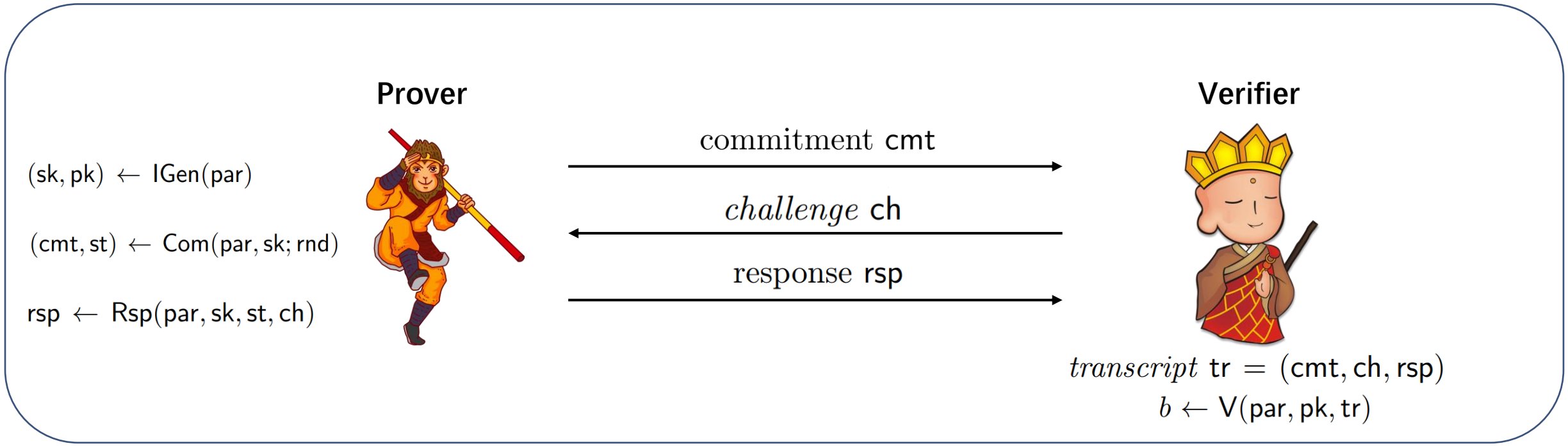
- ***Commitment – Augmentability***
  - *There exists an* append algorithm Apd *that takes as input* cmt *and randomness* $\mathsf{rnd}' \in \mathcal{R}$ *and outputs* cmt' *satisfying*

$$\mathsf{cmt}' \leftarrow \mathsf{Com}_1(\mathsf{par}, \mathsf{sk}; \mathsf{rnd} \cdot \mathsf{rnd}'),$$

Schnorr Instance

- The commitment $\mathbf{cmt} = g^{r_0}$
- $\mathbf{Apd(cmt, rnd')} = (g^{r_0})^{r'} = g^{r_0 \cdot r'}$
- $\mathbf{Com_1(par,sk;rnd \cdot rnd')} = g^{r_0 \cdot r'}$

Attractive Properties of Schnorr Identification



- ***Response – Indistinguishability***

Game $\mathsf{IND}^{\mathsf{RSP}}_{\mathsf{ID},\mathcal{A}}(\lambda)$:

1: $\mathsf{par} \leftarrow \mathsf{Setup}(1^\lambda)$; $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{IGen}(\mathsf{par})$; $\mathsf{rnd} \leftarrow_\$ \mathcal{R}$;
2: $(\mathsf{cmt}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{par}, \mathsf{sk}; \mathsf{rnd})$; $\mathsf{ch} \leftarrow_\$ \mathcal{CH}$;
3: $\mathsf{rsp}_0 \leftarrow \mathsf{Rsp}(\mathsf{par}, \mathsf{sk}, \mathsf{st}, \mathsf{ch})$; $\mathsf{rsp}_1 \leftarrow_\$ \mathcal{Z}$;
4: $b \leftarrow_\$ \{0,1\}; b' \leftarrow \mathcal{A}(\mathsf{par}, \mathsf{sk}, \mathsf{ch}, \mathsf{rsp}_b)$
5: **return** $b \stackrel{?}{=} b'$

negligible

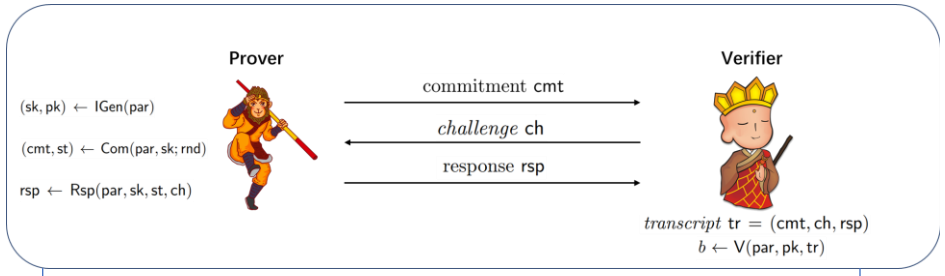
$$\mathsf{Adv}^{\mathsf{RSP}}_{\mathsf{ID},\mathcal{A}}(\lambda) = 2\Pr\left[\mathsf{IND}^{\mathsf{RSP}}_{\mathsf{ID},\mathcal{A}}(\lambda) = 1\right] - 1$$

Schnorr Instance

- $\mathbf{rsp} = z = r_0 - \xi \cdot x$

The response is indistinguish with any random element in $\mathbb{Z}_p$

Main Observation: Map-Invariance



- ✓ Commitment-Independency
- ✓ Commitment-Augmentability
- ✓ Commitment-Recoverability
- ✓ Response-Indistinguishability

$\mathsf{par} \leftarrow \mathsf{Setup}(1^\lambda)$ —rnd'→ $\mathsf{rpar} \leftarrow \mathsf{ParMap}(\mathsf{par}, \mathsf{rnd}')$

$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{IGen}(\mathsf{par})$ —rnd'→ $\mathsf{rpk} \leftarrow \mathsf{PkMap}(\mathsf{pk}, \mathsf{rnd}')$

$\mathsf{rsp} \leftarrow \mathsf{Rsp}(\mathsf{par}, \mathsf{sk}, \mathsf{st}, \mathsf{ch})$

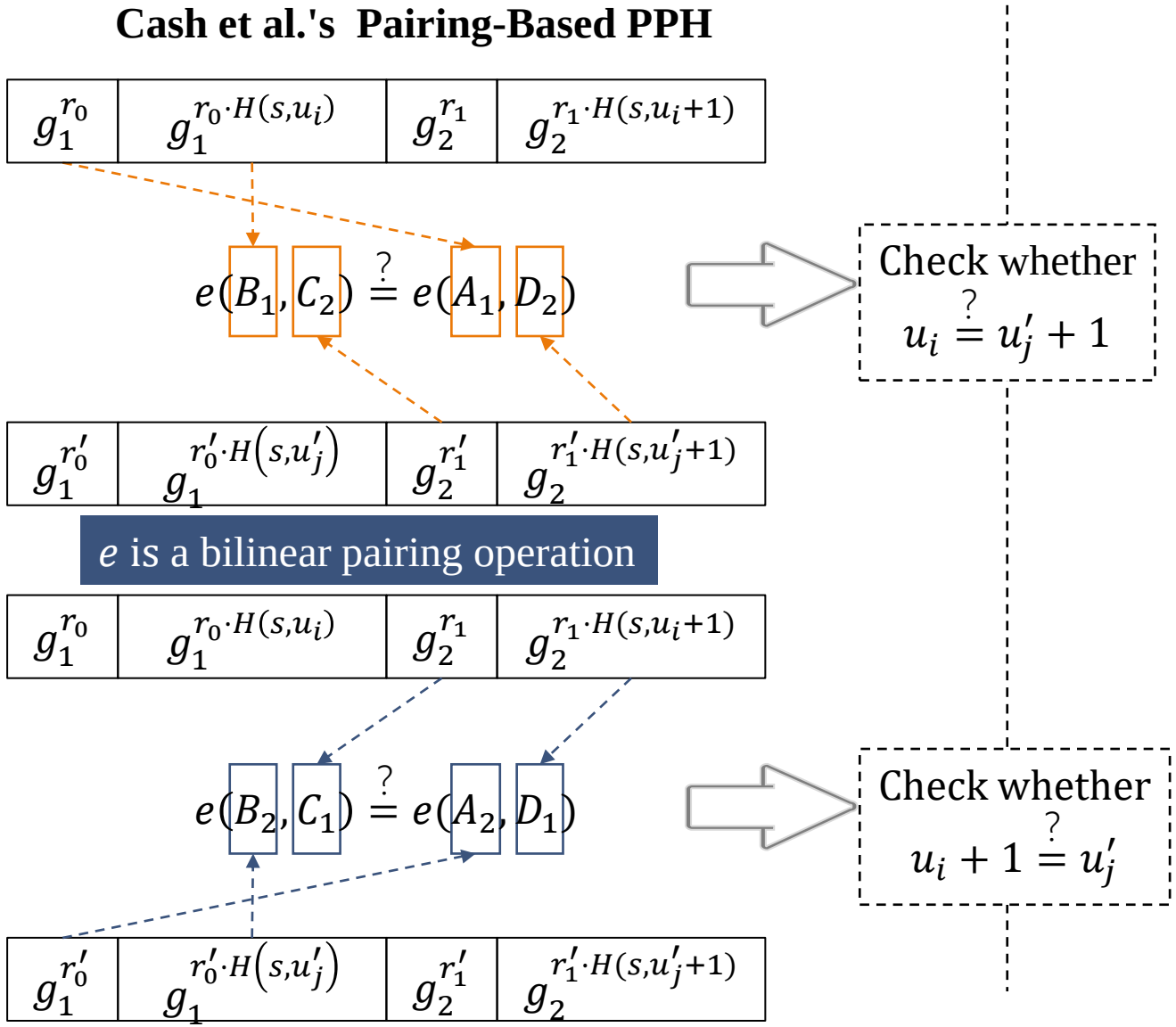
$(\mathsf{cmt}, \mathsf{st}) \leftarrow \mathsf{Com}(\mathsf{par}; \mathsf{rnd})$

$\mathsf{cmt} = \mathsf{Rec}(\mathsf{par}, \mathsf{pk}, \mathsf{ch}, \mathsf{rsp})$

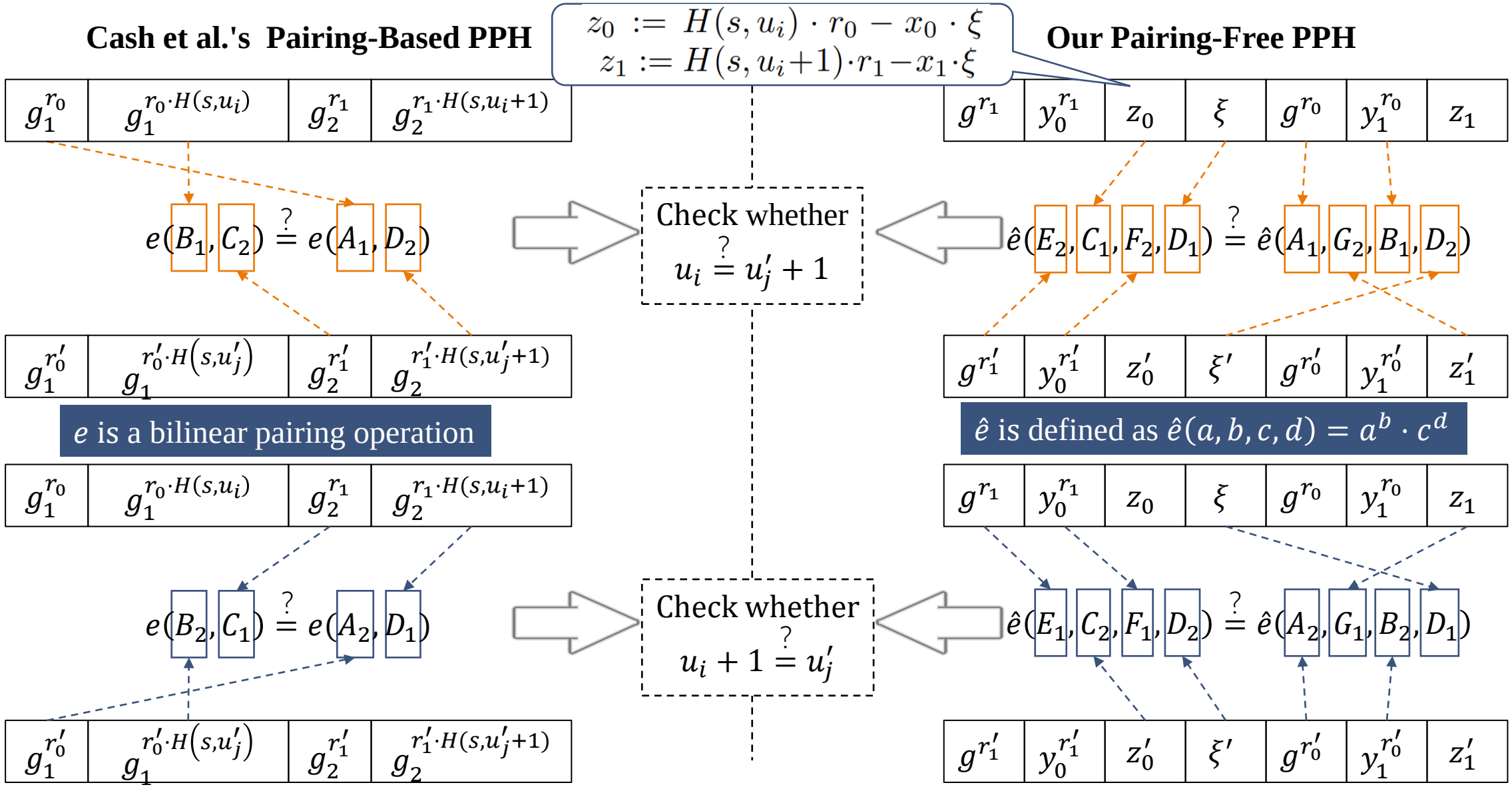
Parameters、public keys and commitments are mapped by **rnd′**, **but the verification equation holds.**

$$\mathsf{Apd}(\mathsf{cmt}, \mathsf{rnd}') = \mathsf{cmt}' = \mathsf{Rec}(\mathsf{rpar}, \mathsf{rpk}, \mathsf{ch}, \mathsf{rsp})$$

Our Efficient PPH from Schnorr Identification



Our Efficient PPH from Schnorr Identification



Our Efficient PPH from Schnorr Identification

$$z_0 := H(s, u_i) \cdot r_0 - x_0 \cdot \xi$$
$$z_1 := H(s, u_i + 1) \cdot r_1 - x_1 \cdot \xi$$

Our Pairing-Free PPH

g^{r_1}	$y_0^{r_1}$	z_0	ξ	g^{r_0}	$y_1^{r_0}$	z_1

$\hat{e}(E_2, C_1, F_2, D_1) \stackrel{?}{=} \hat{e}(A_1, G_2, B_1, D_2)$

$g^{r_1'}$	$y_0^{r_1'}$	z_0'	ξ'	$g^{r_0'}$	$y_1^{r_0'}$	z_1'

$\hat{e}$ is defined as $\hat{e}(a, b, c, d) = a^b \cdot c^d$

Check whether $u_i = u_j' + 1$

$$\hat{e}(E_2, C_1, F_2, D_1) = g^{r_0 \cdot r_1' \cdot H(s, u_i)}$$
$$\hat{e}(A_1, G_2, B_1, D_2) = g^{r_0 \cdot r_1' \cdot H(s, u_i + 1)}$$

Complexity

$O(4n)$, for check whether $u_i = u_j' + 1$;
$O(4n)$, for check whether $u_i + 1 = u_j'$;

g^{r_1}	$y_0^{r_1}$	z_0	ξ	g^{r_0}	$y_1^{r_0}$	z_1

$\hat{e}(E_1, C_2, F_1, D_2) \stackrel{?}{=} \hat{e}(A_2, G_1, B_2, D_1)$

$g^{r_1'}$	$y_0^{r_1'}$	z_0'	ξ'	$g^{r_0'}$	$y_1^{r_0'}$	z_1'

Check whether $u_i + 1 \stackrel{?}{=} u_j'$

$$\hat{e}(E_1, C_2, F_1, D_2) = g^{r_1 \cdot r_0' \cdot H(s, u_i + 1)}$$
$$\hat{e}(A_2, G_1, B_2, D_1) = g^{r_1 \cdot r_0' \cdot H(s, u_j)}$$

Our Efficient PPH from Schnorr Identification

WUHAN UNIVERSITY

Generic PPH Construction

- $\mathsf{PPH.KeyGen}(1^\lambda)$:
  Two key pairs $(\mathsf{sk}_i, \mathsf{pk}_i)$ $(i \in \{0,1\})$. Two keyed hash functions
  - $H_r : \{0,1\}^\lambda \times \{0,1\}^* \to \mathcal{R}$
  - $H_c : \{0,1\}^\lambda \times \{0,1\}^* \to \mathcal{CH}$

  Two keys $s, \kappa \leftarrow_\$ \{0,1\}^\lambda$
- $\mathsf{PPH.Hash}(\mathsf{hk}, u)$:
  Two randomness $r_0, r_1 \leftarrow_\$ \mathcal{R}$
  Two values $\hat{u}_0 \leftarrow H_r(s,u)$, $\hat{u}_1 \leftarrow H_r(s,u+1)$
  The commitments $(\mathsf{cmt}_i, \mathsf{st}_i) \leftarrow \mathsf{Com}(\mathsf{par}_{\mathsf{ID}}; \hat{u}_i \cdot r_i)$
  The challenge $\mathsf{ch} \leftarrow H_c(\kappa, \hat{u}_0 || \hat{u}_1 || \mathsf{cmt}_0 || \mathsf{cmt}_1)$
  The responses $\mathsf{rsp}_i \leftarrow \mathsf{Rsp}(\mathsf{par}_{\mathsf{ID}}, \mathsf{sk}_i, \mathsf{st}_i, \mathsf{ch})$ for $i \in \{0,1\}$.

$$\mathsf{rpar}_0 \leftarrow \mathsf{ParMap}(\mathsf{par}_{\mathsf{ID}}, r_1), \mathsf{rpk}_0 \leftarrow \mathsf{PkMap}(\mathsf{pk}_0, r_1),$$

$$\mathsf{rpar}_1 \leftarrow \mathsf{ParMap}(\mathsf{par}_{\mathsf{ID}}, r_0), \mathsf{rpk}_1 \leftarrow \mathsf{PkMap}(\mathsf{pk}_1, r_0),$$

  Encode the challenge by $\mathsf{ech} = \mathsf{ch} \oplus H_c(\kappa, \mathsf{rsp}_0 || \mathsf{rsp}_1)$.
- $\mathsf{PPH.Test}(\mathsf{tk}, h, h')$:
  Two hash values

$$h = \{\mathsf{rpar}_0, \mathsf{rpar}_1, \mathsf{rpk}_0, \mathsf{rpk}_1, \mathsf{ech}, \mathsf{rsp}_0, \mathsf{rsp}_1\},$$

$$h' = \{\mathsf{rpar}'_0, \mathsf{rpar}'_1, \mathsf{rpk}'_0, \mathsf{rpk}'_1, \mathsf{ech}', \mathsf{rsp}'_0, \mathsf{rsp}'_1\},$$

  Recover the challenges $\mathsf{ch} = \mathsf{ech} \oplus H_c(\kappa, \mathsf{rsp}_0 || \mathsf{rsp}_1)$ and $\mathsf{ch}' = \mathsf{ech}' \oplus H_c(\kappa, \mathsf{rsp}'_0 || \mathsf{rsp}'_1)$.
  Finally, it outputs a flag $b = 1$ if $\mathsf{rec}_0 = \mathsf{rec}'_1$, or $b = -1$ if $\mathsf{rec}_1 = \mathsf{rec}'_0$, otherwise $b = 0$.

$$\begin{cases} \mathsf{rec}_0 \leftarrow \mathsf{Rec}(\mathsf{rpar}'_0, \mathsf{rpk}'_0, \mathsf{ch}, \mathsf{rsp}_0), \\ \mathsf{rec}'_1 \leftarrow \mathsf{Rec}(\mathsf{rpar}_1, \mathsf{rpk}_1, \mathsf{ch}', \mathsf{rsp}'_1), \\ \mathsf{rec}_1 \leftarrow \mathsf{Rec}(\mathsf{rpar}'_1, \mathsf{rpk}'_1, \mathsf{ch}, \mathsf{rsp}_1), \\ \mathsf{rec}'_0 \leftarrow \mathsf{Rec}(\mathsf{rpar}_0, \mathsf{rpk}_0, \mathsf{ch}', \mathsf{rsp}'_0), \end{cases}$$

PPH Construction from Schnorr Identification

$\mathsf{PPH.KeyGen}(1^\lambda)$

1: Pick keyed hash functions $H_r : \{0,1\}^\lambda \times \{0,1\}^* \to \mathbb{Z}_p^*$ and $H_c : \{0,1\}^\lambda \times \{0,1\}^* \to \mathbb{Z}_p$.
2: $g \leftarrow_\$ \mathbb{G}$, $s, \kappa, x_0, x_1 \leftarrow_\$ \mathbb{Z}_p$
3: $y_0 := g^{x_0}$, $y_1 := g^{x_1}$
4: $\mathsf{par}_{\mathsf{PPH}} := \{g, H_r, H_c\}$
5: $\mathsf{hk} := \{s, \kappa, x_0, x_1, g, y_0, y_1\}$, $\mathsf{tk} := \{\kappa\}$
6: **return** $(\mathsf{par}_{\mathsf{PPH}}, \mathsf{hk}, \mathsf{tk})$

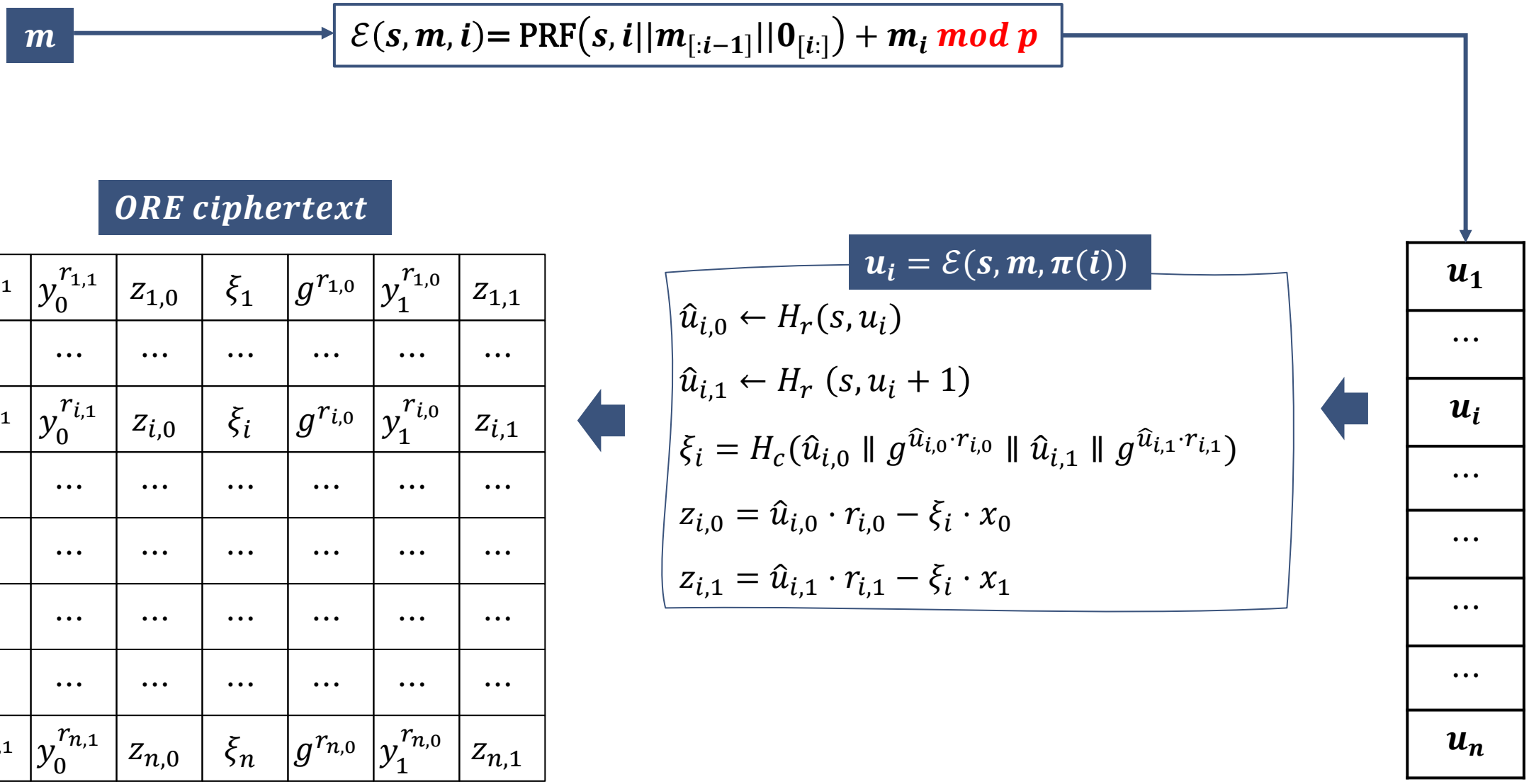
$\mathsf{PPH.Hash}(\mathsf{hk}, u)$

1: $\{s, \kappa, x_0, x_1, g, y_0, y_1\} \leftarrow \mathsf{hk}$
2: $\hat{u}_0 \leftarrow H_r(s,u)$, $\hat{u}_1 \leftarrow H_r(s,u+1)$
3: $r_0, r_1 \leftarrow_\$ \mathbb{Z}_p$, $w_0 := g^{\hat{u}_0 \cdot r_0}$, $w_1 := g^{\hat{u}_1 \cdot r_1}$
4: $\xi \leftarrow H_c(\hat{u}_0 || \hat{u}_1 || w_0 || w_1)$
5: $z_0 := \hat{u}_0 \cdot r_0 - \xi \cdot x_0$, $z_1 := \hat{u}_1 \cdot r_1 - \xi \cdot x_1$
6: $\hat{g}_0 := g^{r_1}$, $\hat{y}_0 := y_0^{r_1}$, $\hat{g}_1 := g^{r_0}$, $\hat{y}_1 := y_1^{r_0}$
7: $\hat{\xi} := \xi \oplus H_c(\kappa || z_0 || z_1)$
8: **return** $H := \{\hat{g}_0, \hat{g}_1, \hat{y}_0, \hat{y}_1, \hat{\xi}, z_0, z_1\}$

$\mathsf{PPH.Test}(\mathsf{tk}, h, h')$

1: $\{\hat{g}_0, \hat{g}_1, \hat{y}_0, \hat{y}_1, \hat{\xi}, z_0, z_1\} \leftarrow h$
2: $\{\hat{g}'_0, \hat{g}'_1, \hat{y}'_0, \hat{y}'_1, \hat{\xi}', z'_0, z'_1\} \leftarrow h'$
3: $\xi := \hat{\xi} \oplus H_c(\kappa || z_0 || z_1)$, $\xi' := \hat{\xi}' \oplus H_c(\kappa || z'_0 || z'_1)$
4: $\mathsf{rec}_0 := (\hat{g}'_0)^{z_0} \cdot (\hat{y}'_0)^{\xi}$, $\mathsf{rec}'_1 := (\hat{g}_1)^{z'_1} \cdot (\hat{y}_1)^{\xi'}$
5: **if** $\mathsf{rec}_0 = \mathsf{rec}'_1$ **then**
6: **return** 1
7: $\mathsf{rec}_1 := (\hat{g}'_1)^{z_1} \cdot (\hat{y}'_1)^{\xi}$, $\mathsf{rec}'_0 := (\hat{g}_0)^{z'_0} \cdot (\hat{y}_0)^{\xi'}$
8: **if** $\mathsf{rec}_1 = \mathsf{rec}'_0$ **then**
9: **return** -1
10: **return** 0

From PPH to Parameter-Hiding ORE



Reduce ORE Ciphertext Size

ORE ciphertext

$g^{r_{1,1}}$	$y_0^{r_{1,1}}$	$z_{1,0}$	ξ_1	$g^{r_{1,0}}$	$y_1^{r_{1,0}}$	$z_{1,1}$
…	…	…	…	…	…	…
$g^{r_{i,1}}$	$y_0^{r_{i,1}}$	$z_{i,0}$	ξ_i	$g^{r_{i,0}}$	$y_1^{r_{i,0}}$	$z_{i,1}$
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
…	…	…	…	…	…	…
$g^{r_{n,1}}$	$y_0^{r_{n,1}}$	$z_{n,0}$	ξ_n	$g^{r_{n,0}}$	$y_1^{r_{n,0}}$	$z_{n,1}$

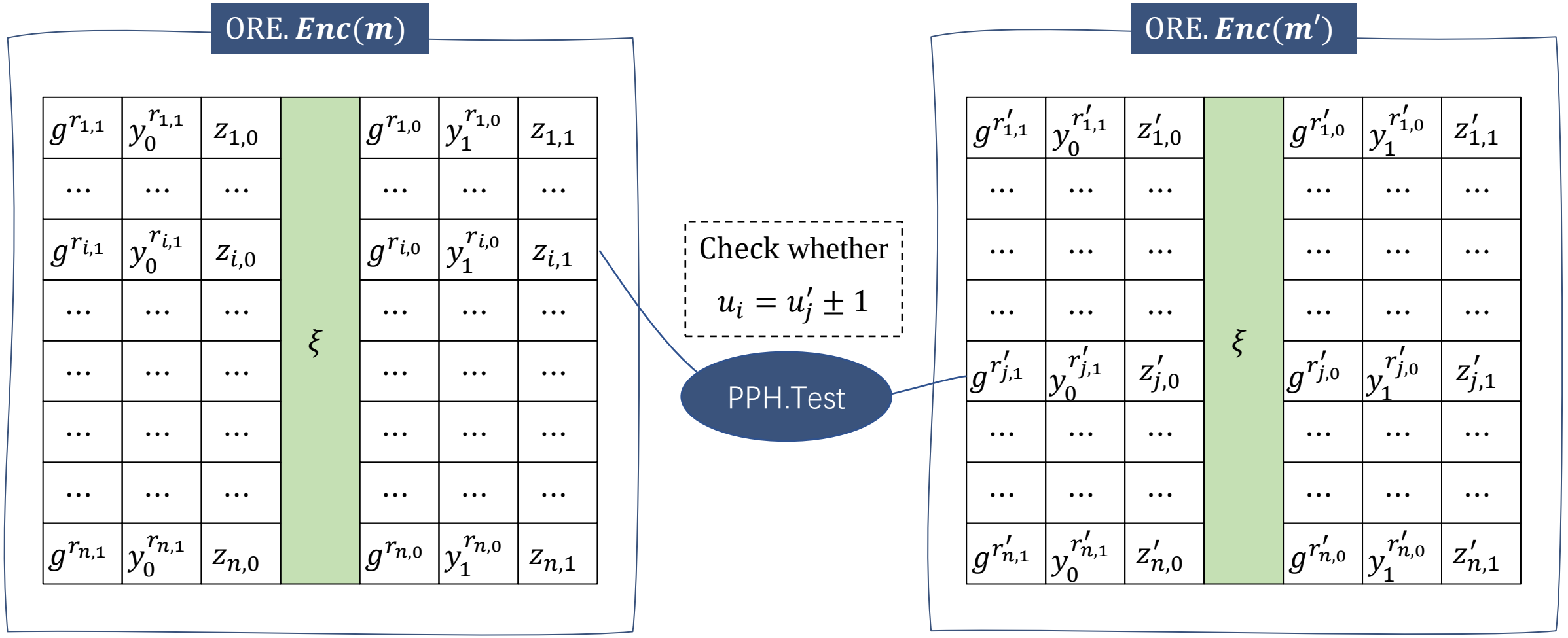
$$\xi_i = H_c(\hat{u}_{i,0} \parallel g^{\hat{u}_{i,0} \cdot r_{i,0}} \parallel \hat{u}_{i,1} \parallel g^{\hat{u}_{i,1} \cdot r_{i,1}})$$

Optimized ORE ciphertext

$g^{r_{1,1}}$	$y_0^{r_{1,1}}$	$z_{1,0}$		$g^{r_{1,0}}$	$y_1^{r_{1,0}}$	$z_{1,1}$
…	…	…		…	…	…
$g^{r_{i,1}}$	$y_0^{r_{i,1}}$	$z_{i,0}$		$g^{r_{i,0}}$	$y_1^{r_{i,0}}$	$z_{i,1}$
…	…	…		…	…	…
…	…	…	ξ	…	…	…
…	…	…		…	…	…
…	…	…		…	…	…
$g^{r_{n,1}}$	$y_0^{r_{n,1}}$	$z_{n,0}$		$g^{r_{n,0}}$	$y_1^{r_{n,0}}$	$z_{n,1}$

$$\xi = H_c(\{\hat{u}_{i,0} \parallel g^{\hat{u}_{i,0} \cdot r_{i,0}} \parallel \hat{u}_{i,1} \parallel g^{\hat{u}_{i,1} \cdot r_{i,1}}\}_{1 \le i \le n})$$

From PPH to Parameter-Hiding ORE



At most $8n^2$ group exponentiation operations

Improving Efficiency with $\mathcal{L}_1'$-Leakage

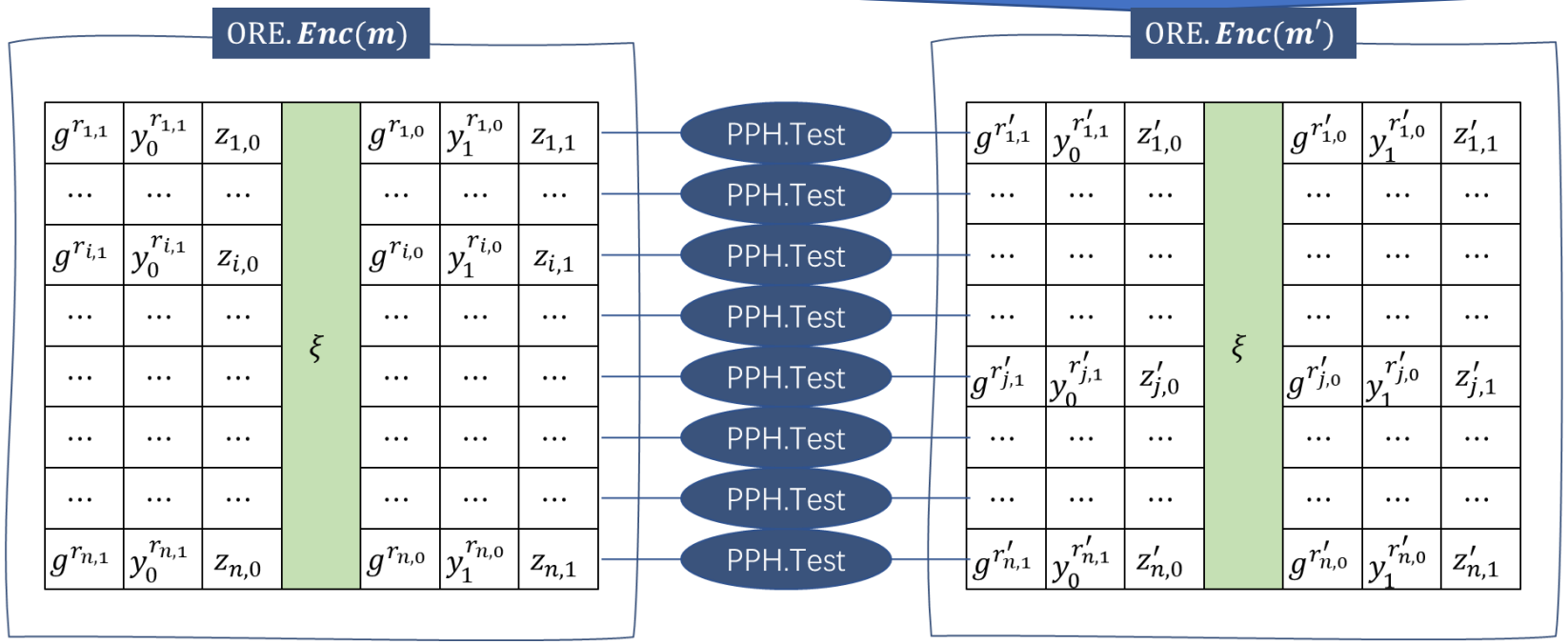
- $\mathcal{L}_1$ - Smooth CLWW leakage

$$\mathcal{L}_1(m_1, \cdots, m_q) := \left(\forall 1 \le i, j, k \le q, \mathbf{1}(m_i < m_j), \mathbf{1}(\mathbf{msdb}(m_i, m_j) = \mathbf{msdb}(m_i, m_k))\right)$$

- $\mathcal{L}_1'$- Smooth CLWW leakage

$$\mathcal{L}_1'(m_1, \cdots, m_q) := \left(\forall 1 \le i, j, k, l \le q, \mathbf{1}(m_i < m_j), \mathbf{1}(\mathbf{msdb}(m_i, m_j) = \mathbf{msdb}(m_k, m_l))\right)$$

Fix the permutation π in msk to replace randomized permutation in Hash

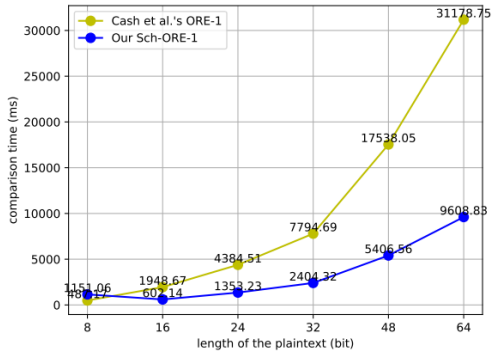


At most $8n$ group exponentiation operations

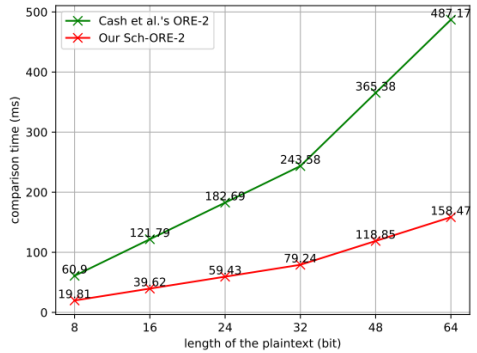
Experimental Evaluation

ORE Scheme	Ciphertext size	Encryption cost	Comparison Cost	Leakage
Cash et al.'s ORE [10]	$2n\|\mathbb{G}_1\| + 2n\|\mathbb{G}_2\|$	$2n\mathsf{PM}_1 + 2n\mathsf{PM}_2 + 3n\mathsf{PRF}$	$4n^2\mathsf{BP}$	$\mathcal{L}_1$
Cash et al.'s ORE* [10]	$2n\|\mathbb{G}_1\| + 2n\|\mathbb{G}_2\|$	$2n\mathsf{PM}_1 + 2n\mathsf{PM}_2 + 3n\mathsf{PRF}$	$4n\mathsf{BP}$	$\mathcal{L}_1'$
Our Sch-ORE	$4n\|\mathbb{G}_1\| + 3n\|\mathbb{Z}_p\|$	$6n\mathsf{PM}_1 + 5n\mathsf{PRF}$	$8n^2\mathsf{PM}_1 + 2n\mathsf{PRF}$	$\mathcal{L}_1$
Our Sch-ORE*	$4n\|\mathbb{G}_1\| + 3n\|\mathbb{Z}_p\|$	$6n\mathsf{PM}_1 + 5n\mathsf{PRF}$	$8n\mathsf{PM}_1 + 2n\mathsf{PRF}$	$\mathcal{L}_1'$

Table 1. Comparison with existing ORE schemes with smoothed CLWW leakage. λ is the security parameter, n is the bit-length of the plaintext; $|\mathbb{G}_1|$ and $|\mathbb{G}_2|$ is the size of elements in groups $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively; PM_1, PM_2 and BP is the group exponentiation in $\mathbb{G}_1$ and $\mathbb{G}_2$ and the bilinear pairing operation, respectively; PRF is the pseudorandom function operation. ORE* refers to the ORE scheme with fixed permutation.



(c) Comparison time in ORE with randomized permutations



(d) Comparison time in ORE with fixed permutations

n	Scheme	Ciphertext size (KB)	Encryption cost (ms)	Comparison Cost (ms)	Leakage
8	Cash et al.'s ORE [10]	2.50	43.52	487.17	$\mathcal{L}_1$
	Cash et al.'s ORE* [10]	2.50	43.52	60.91	$\mathcal{L}_1'$
	Our Sch-ORE	1.72	16.71	151.06	$\mathcal{L}_1$
	Our Sch-ORE	**1.72**	**16.71**	**19.81**	$\mathcal{L}_1'$
16	Cash et al.'s ORE [10]	5.00	87.04	1948.67	$\mathcal{L}_1$
	Cash et al.'s ORE* [10]	5.00	87.04	121.97	$\mathcal{L}_1'$
	Our Sch-ORE	3.44	33.42	602.14	$\mathcal{L}_1$
	Our Sch-ORE	**3.44**	**33.42**	**39.62**	$\mathcal{L}_1'$
24	Cash et al.'s ORE [10]	7.50	130.56	4384.51	$\mathcal{L}_1$
	Cash et al.'s ORE* [10]	7.50	130.56	182.69	$\mathcal{L}_1'$
	Our Sch-ORE	5.16	50.12	1353.23	$\mathcal{L}_1$
	Our Sch-ORE	**5.16**	**50.12**	**59.43**	$\mathcal{L}_1'$
32	Cash et al.'s ORE [10]	10.00	174.08	7794.69	$\mathcal{L}_1$
	Cash et al.'s ORE* [10]	10.00	174.08	243.58	$\mathcal{L}_1'$
	Our Sch-ORE	6.88	66.83	2404.32	$\mathcal{L}_1$
	Our Sch-ORE	**6.88**	**66.83**	**79.24**	$\mathcal{L}_1'$
48	Cash et al.'s ORE [10]	15.00	261.12	17538.05	$\mathcal{L}_1$
	Cash et al.'s ORE* [10]	15.00	261.12	365.38	$\mathcal{L}_1'$
	Our Sch-ORE	10.31	100.25	5406.56	$\mathcal{L}_1$
	Our Sch-ORE	**10.31**	**100.25**	**118.85**	$\mathcal{L}_1'$
64	Cash et al.'s ORE [10]	20.00	348.16	31178.75	$\mathcal{L}_1$
	Cash et al.'s ORE* [10]	20.00	348.16	487.17	$\mathcal{L}_1'$
	Our Sch-ORE	13.75	133.67	9608.83	$\mathcal{L}_1$
	Our Sch-ORE	**13.75**	**133.67**	**158.47**	$\mathcal{L}_1'$

Table 2. Ciphertext size in KB and running time in milliseconds of ORE schemes with different message bit-length n in {8, 16, 24, 32, 48, 64}.ORE* refers to the ORE scheme with fixed permutation.

Our implementation is available at https://github.com/cpeng-crypto/pORE

Thanks for Your Attention!

Any Questions?