Selective Opening Security in the Quantum Random Oracle Model, Revisited

Jiaxin Pan

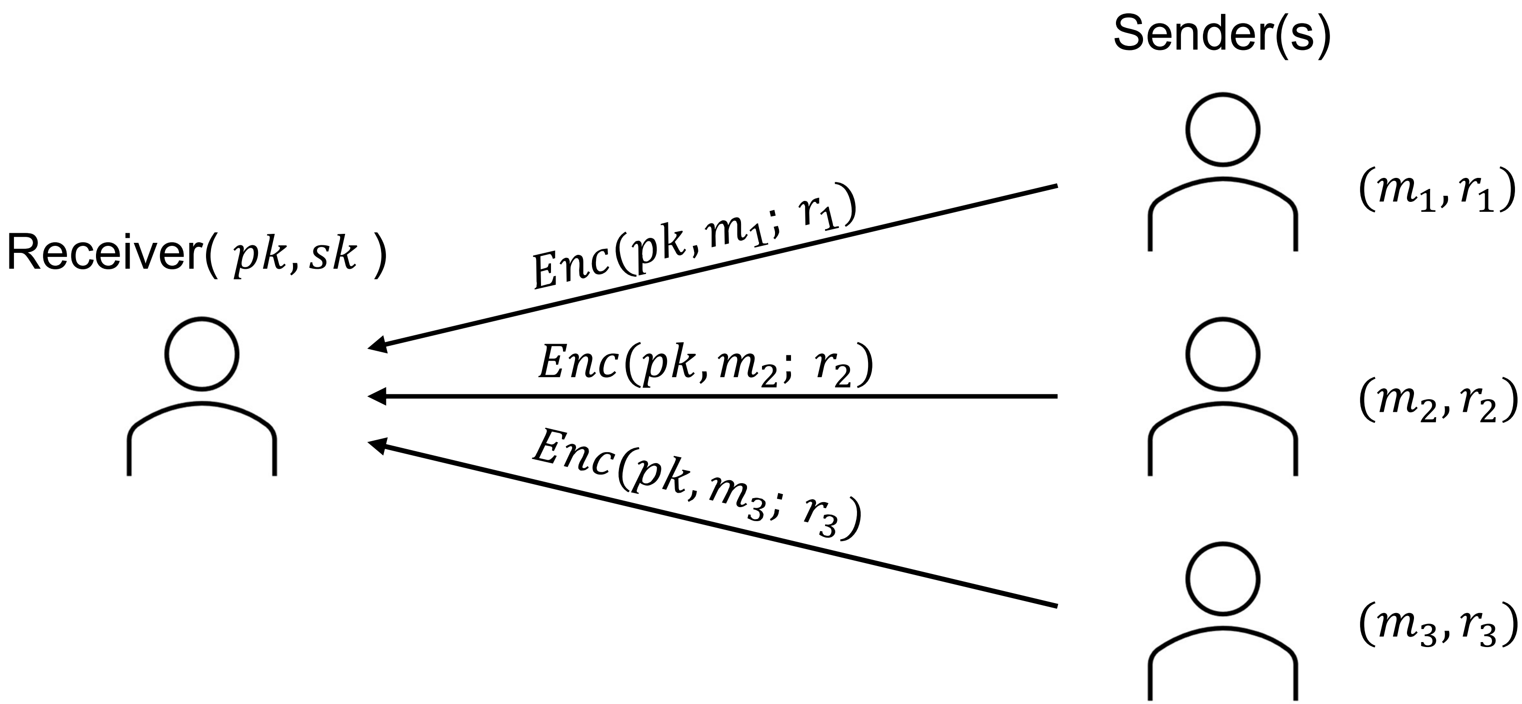
UNIKASSEL VERSITÄT

Runzhi Zeng

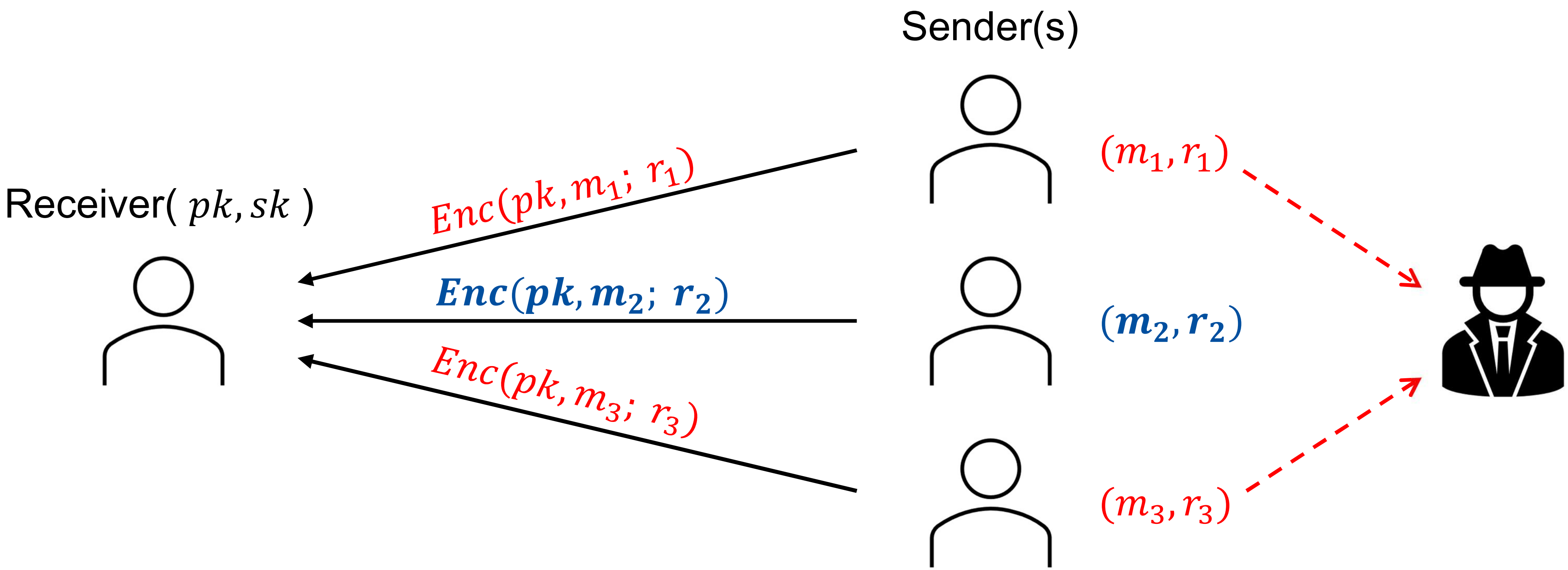
NTNU

Norwegian University of Science and Technology

Selective Opening Security

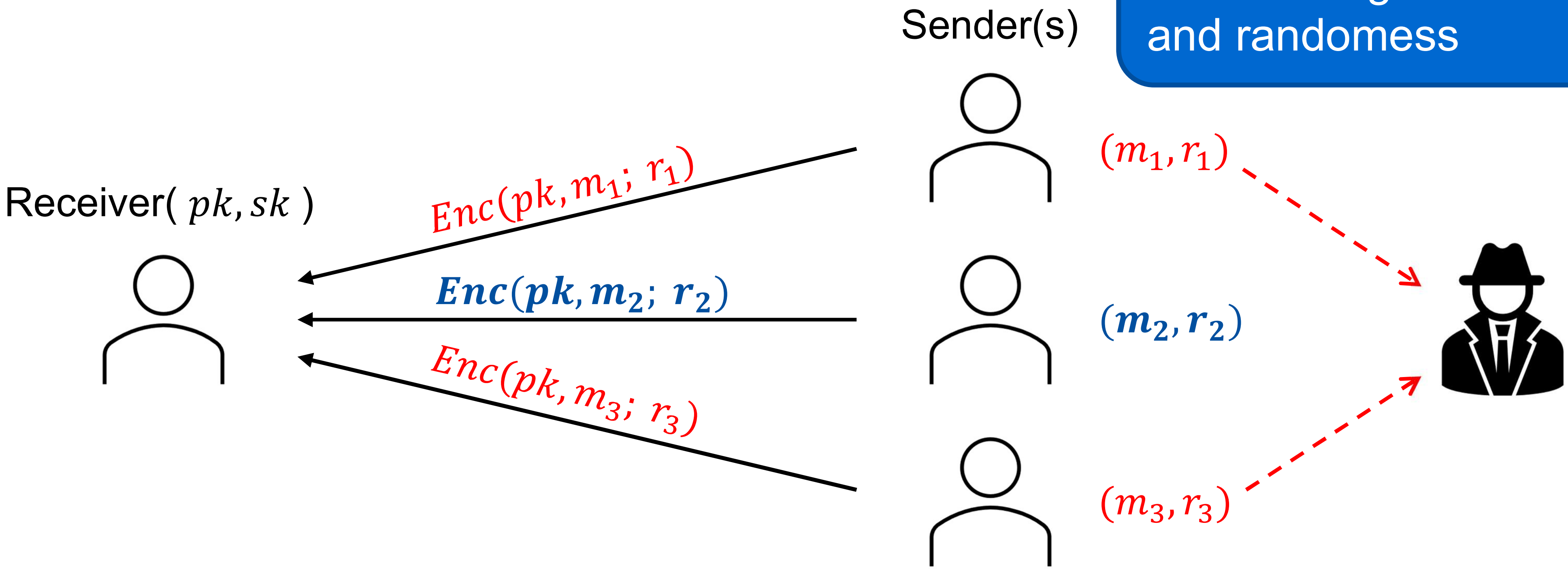


Selective Opening Security



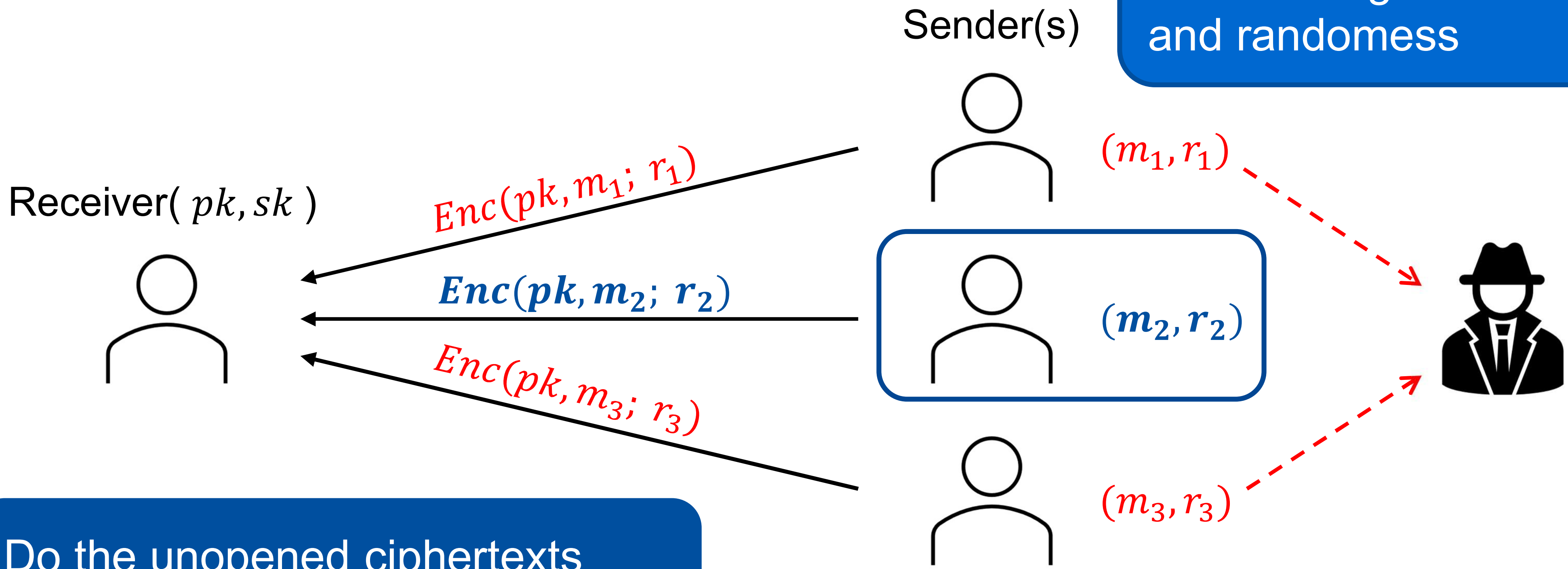
Selective Opening Security

Opening a ciphertext
reveals msg
and randomess



Selective Opening Security

Opening a ciphertext reveals msg and randomess



Do the unopened ciphertexts remain secure?

Selective Opening Security

- (Sender) Selective Opening Security
  - Date back to [DNS99]
  - Why SO? Sender corruptions, randomness leakage, ...

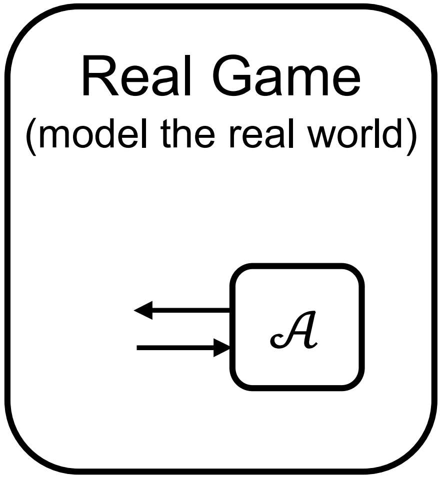
Selective Opening Security

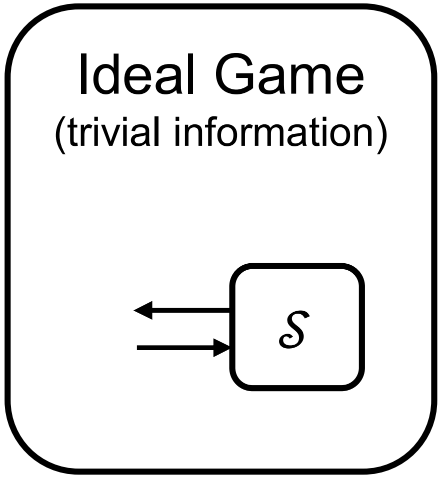
- (Sender) Selective Opening Security
  - Date back to [DNS99]
  - Why SO? Sender corruptions, randomness leakage, ...

- Definitions for SO security [DNRS99, BHY09, HLOV11, BHK12,...]
- Two flavors of SO security
  - Indistinguishability-based SO (IND-SO) [BHY09, BHK12, …]
  - Simulation-based SO (SIM-SO) [DNRS99, BHY09, ...]
  - SIM-SO => IND-SO [BHK12]

Simulation-based SO Security

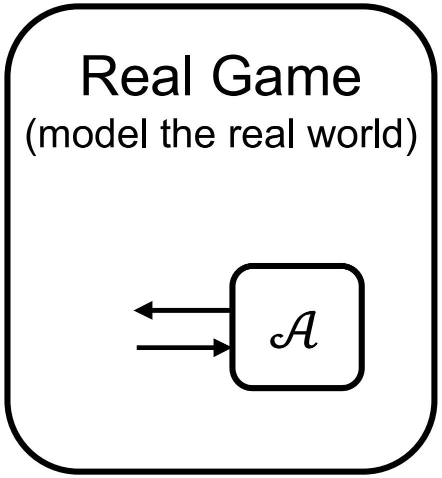
- Real game and Ideal game
  - Real game models a real-world adversary $\mathcal{A}$ attacks in the real-world scenario
  - Ideal game models a simulator $\mathcal{S}$ attacks in an ideal world with all trivial information





Simulation-based SO Security

- Real game and Ideal game
  - Real game models a real-world adversary $\mathcal{A}$ attacks in the real-world scenario
  - Ideal game models a simulator $\mathcal{S}$ attacks in an ideal world with all trivial information



In Real Game, $\mathcal{A}$ is allowed to:

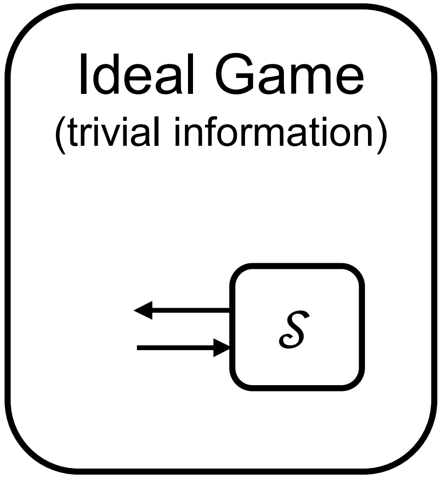
- Choose the message distribution
- Open any challenge ciphertext and get
  - The **decrypted message**
  - The **randomness** used for generating the ciphertext
- Decryption Oracle (for CCA security)

Simulation-based SO Security

- Real game and Ideal game
  - Real game models a real-world adversary $\mathcal{A}$ attacks in the real-world scenario
  - Ideal game models a simulator $\mathcal{S}$ attacks in an ideal world with all trivial information

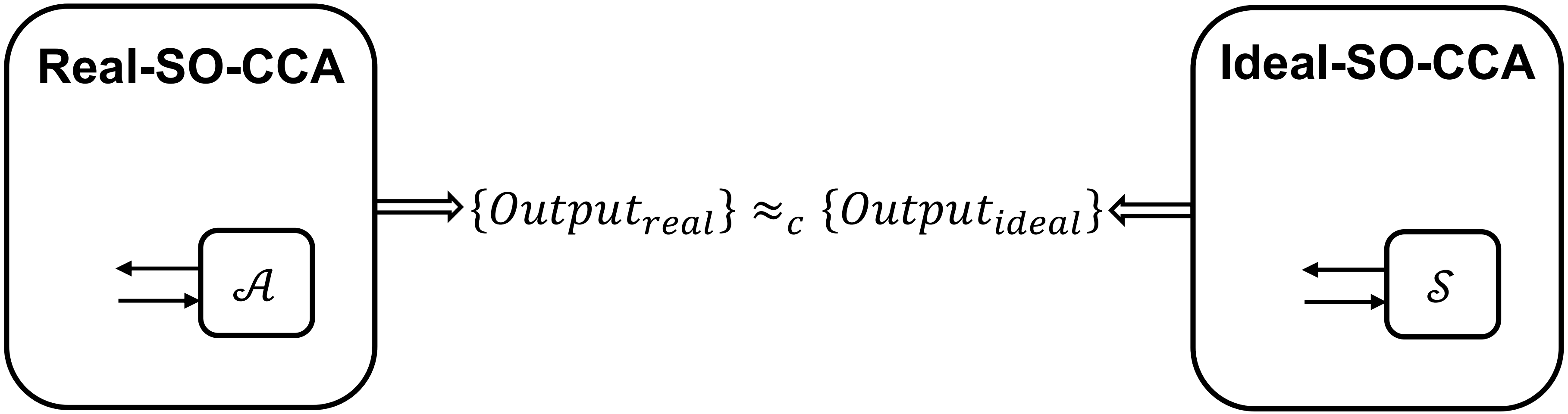
In Ideal Game, $\mathcal{S}$ is allowed to:

- Choose the message distribution
- Open **dummy** messages
  - No public key, no challenge ciphertexts, no randomness…
- Decryption Oracle (for CCA security)



Simulation-based SO Security

- **SIM-SO-CCA Security:** $\forall \mathcal{A}$**,** there exists a simulator $\mathcal{S}$ (both are PPT) such that…



- $\mathcal{S}$ simulates all “behaviors” of $\mathcal{A}$ (e.g., they choose the same messages distribution, open the same ciphertexts, produce the same output)

Why is hard to achieve SIM-SO

- SIM-SO-CCA is strictly stronger than (multi-challenge) IND-CCA [BDWY11]

Why is hard to achieve SIM-SO

- SIM-SO-CCA is strictly stronger than (multi-challenge) IND-CCA [BDWY11]
- A naïve “hybrid argument + IND-CCA” approach does not work

c_1 c_2 … c_i … c_n

Cannot open c_i (since the IND-CCA experiment does not provide randomness)…

Why is hard to achieve SIM-SO

- SIM-SO-CCA is strictly stronger than (multi-challenge) IND-CCA [BDWY11]
- A naïve "hybrid argument + IND-CCA" approach does not work

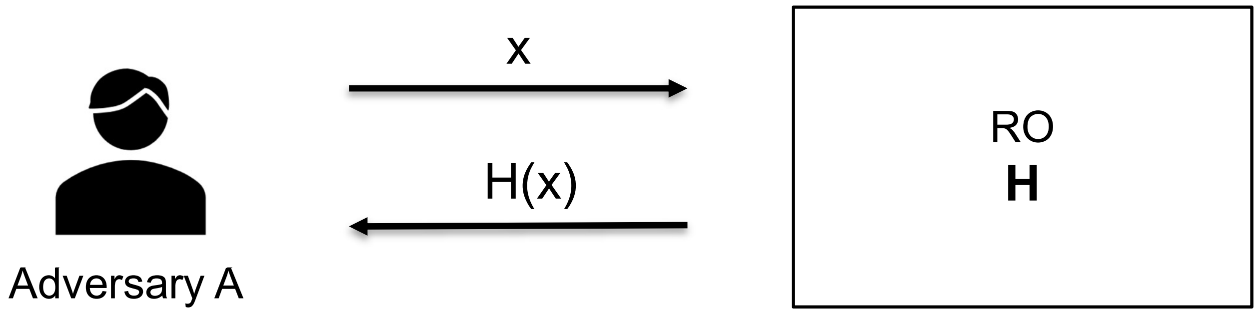
$$\boxed{c_1} \quad \boxed{c_2} \quad \cdots \quad \boxed{c_i} \quad \cdots \quad \boxed{c_n}$$

Cannot open c_i (since the IND-CCA experiment does not provide randomness)…

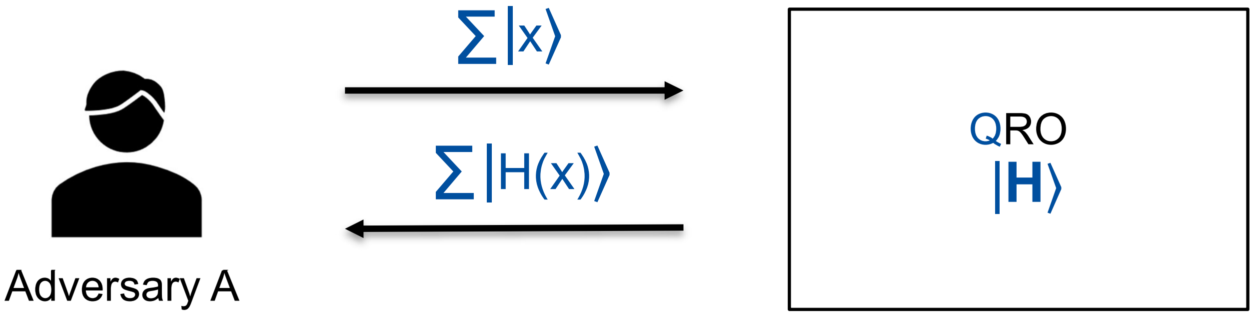
- A trivial guessing technique does not work

  (namely, guess which ciphertext will not be opened, security loss $1/2^n$)

Random Oracle Model



Quantum Random Oracle Model



Compact and Efficient SO-CCA Construction

- Constructions with compact ciphertext
  - |ct| / |pk| = constant, |ct| / |msg| = constant
  - More practical and efficient

Compact and Efficient SO-CCA Construction

- Based on Fujisaki-Okamoto's Transformation (FOT) [HJR16, SS19, PWZ23]
  - [HJR16]: FOT KEM + OTP and FO PKE, in the classical ROM
  - [HP16]: (FOT KEM) + DEM, in the classical ROM
  - **[SS19]: FOT KEM + OTP/DEM, in the QROM**
  - [PWZ23]: (modified) FOT KEM + OTP, in the classical ROM

Compact and Efficient SO-CCA Construction

- Based on Fujisaki-Okamoto's Transformation (FOT) [HJR16, SS19, PWZ23]
  - [HJR16]: FOT KEM + OTP and FO PKE, in the classical ROM
  - [HP16]: (FOT KEM) + DEM, in the classical ROM
  - **[SS19]: FOT KEM + OTP/DEM, in the QROM (…a subtle gap in the proof…)**
  - [PWZ23]: (modified) FOT KEM + OTP, in the classical ROM

The hybrid-argument issue

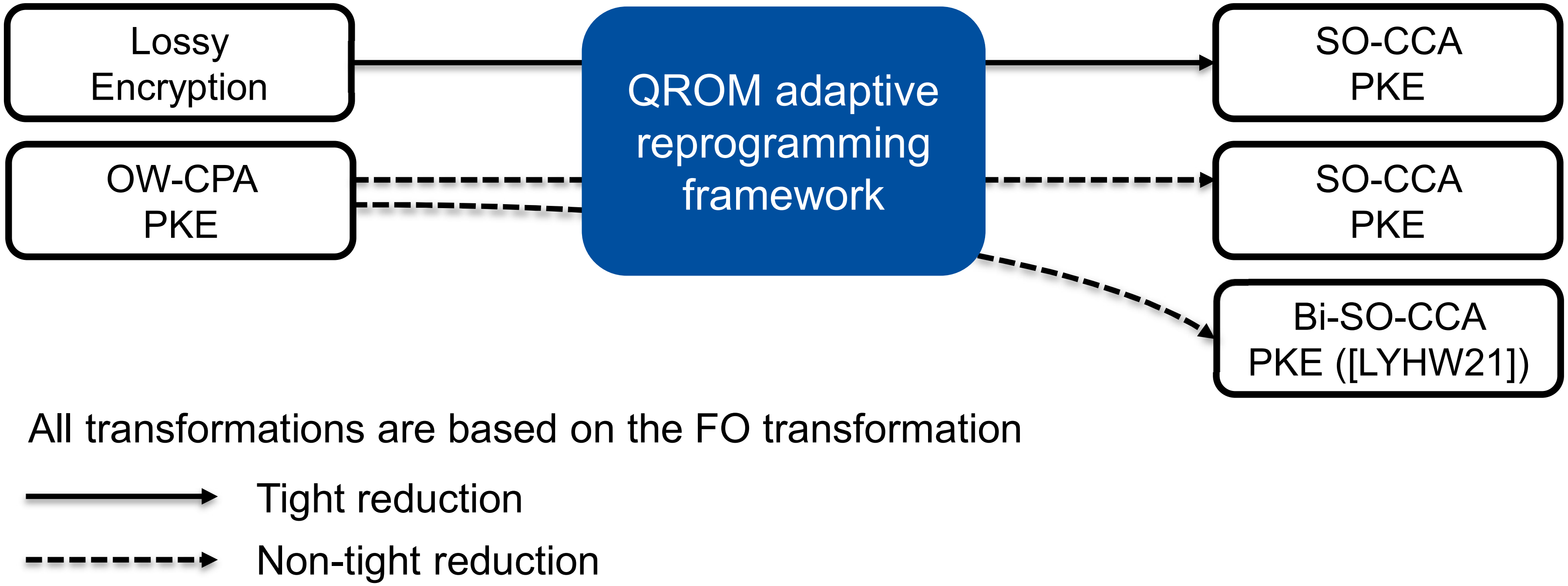
Compact and Efficient SO-CCA Construction

- Based on Fujisaki-Okamoto's Transformation (FOT) [HJR16, SS19, PWZ23]
  - [HJR16]: FOT KEM + OTP and FO PKE, in the classical ROM
  - [HP16]: (FOT KEM) + DEM, in the classical ROM
  - [SS19]: FOT KEM + OTP/DEM, in the QROM (…a subtle gap in the proof…)
  - [PWZ23]: (modified) FOT KEM + OTP, in the classical ROM

- FOT is widely used in post-quantum KEM/PKE (e.g., Crystal-Kyber…)
- Analyses in the classical ROM may not be sufficient for full post-quantum security

Goal: SO security of FOT-based constructions in the QROM

Contribution



NTNU

Contribution

Lossy Encryption

QROM adaptive

SO-CCA PKE

Schemes	Construction	Assumptions	Model
HJKS15	FOT	DH, …	ROM
PWZ23	(Modified) FOT	DH, …	ROM
Ours	**FOT**	**PQ assumptions, …**	**QROM**

Tight reduction

Non-tight reduction

NTNU

A FOT Construction

- FO-Enc(pk, m):
  - $r \leftarrow_{\$} MSP$
  - c := OW-Enc(pk, r; G(r)) // FO derandomization
  - (K, K^{mac}) := H(r, c) // Derive two keys
  - $d := K \oplus m$ // One-time pad
  - τ := MAC.Sign(K^{mac}, (c, d)) // One-time MAC
  - return (c, d, τ)

A FOT Construction

- FO-Enc(pk, m):
  - $r \leftarrow_{\$}$ MSP
  - c := OW-Enc(pk, r; G(r))
  - (K, K^{mac}) := H(r, c)
  - d := K $\oplus$ m
  - τ := MAC.Sign(K^{mac}, (c, d))
  - return (c, d, τ)

In Reduction:

- FO-Enc(pk, m):
  - $r \leftarrow_{\$}$ MSP
  - **c^* ← OW-CPA.Challenge**
  - K^{mac} ← **\$** (K, K^{mac}) := H(m, c)
  - **d ← \$** d := K $\oplus$ m
  - τ := MAC.Sign(K^{mac}, (c, d))
  - return **(c, d, τ)**

A FOT Construction

In Reduction:

- FO-Enc(pk, m):
  - $r \leftarrow_{\$} MSP$
  - $c^* \leftarrow$ **OW-CPA.Challenge**
  - $K^{mac} \leftarrow \$$ $(K, K^{mac}) := H(r, c)$
  - $d \leftarrow \$$ $d := K \oplus m$
  - $\tau := \text{MAC.Sign}(K^{mac}, (c, d))$
  - return (c, d, τ)

$(c_1, d_1, \tau_1), \ldots, (c_i, d_i, \tau_i), \ldots, (c_n, d_n, \tau_n)$

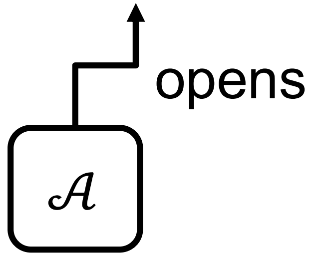
$\boxed{\mathcal{A}}$

A FOT Construction

In Reduction:

- FO-Enc(pk, m):
  - $\boxed{r \leftarrow_{\$} \mathrm{MSP}}$
  - **$c^* \leftarrow$ OW-CPA.Challenge**
  - $K^{mac} \leftarrow \$$ $(K, K^{mac}) := H(r, c)$
  - **$d \leftarrow \$$** $d := K \oplus m$
  - $\tau := \mathrm{MAC.Sign}(K^{mac}, (c, d))$
  - return **(c, d, τ)**

$(c_1, d_1, \tau_1), \ldots, (c_i, d_i, \tau_i), \ldots, (c_n, d_n, \tau_n)$

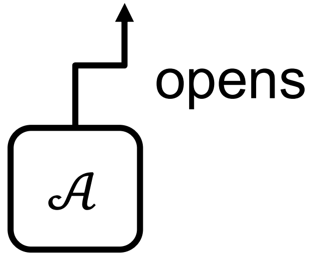


A FOT Construction

In Reduction:

- FO-Enc(pk, m):
  - $\boxed{r \leftarrow_{\$} \mathrm{MSP}}$
  - **$c^* \leftarrow$ OW-CPA.Challenge**
  - $K^{mac} \leftarrow$ **\$** $(K, K^{mac}) := H(r, c)$
  - **d** $\leftarrow$ **\$** $d := K \oplus m$
  - $\tau := \mathrm{MAC.Sign}(K^{mac}, (c, d))$
  - return **(c, d,** τ**)**

$(c_1, d_1, \tau_1), \ldots, (c_i, d_i, \tau_i), \ldots, (c_n, d_n, \tau_n)$



Solution: Reprogram ROs [HJKS15]

- $K_i := \mathbf{d}_i \oplus m_i$
- Reprogram $H(*, c_i) := (K_i, K_i^{mac})$
- By One-wayness, $\mathcal{A}$ detects such reprogramming within a negl probability

A FOT Construction

In Reduction:

- FO-Enc(pk, m):
  - $\boxed{r \leftarrow_{\$} \mathrm{MSP}}$
  - **$c^* \leftarrow$ OW-CPA.Challenge**
  - $K^{mac} \leftarrow \$$ $(K, K^{mac}) := H(r, c)$
  - **d** $\leftarrow \$$ $d := K \oplus m$
  - $\tau := \mathrm{MAC.Sign}(K^{mac}, (c, d))$
  - return **(c, d,** τ**)**

No tools for **computational** QROM adaptive reprogramming

$(c_1, d_1, \tau_1), \ldots, (c_i, d_i, \tau_i), \ldots, (c_n, d_n, \tau_n)$

opens

$\mathcal{A}$

RO adaptive reprogramming

Solution: Reprog

- $K_i := d_i \oplus m_i$
- Reprogram $H(*, c_i) := (K_i, K_i^{mac})$
- By One-wayness, $\mathcal{A}$ detects such reprogramming within a negl probability

QROM Adaptive Reprogramming Framework

Repro Game (for two-stage adversaries)

1. $\mathrm{inp}_0 \leftarrow$ Initialize
2. $\mathrm{out}_0 \leftarrow \mathcal{A}^{H_0}(\mathrm{inp}_0)$ // First stage
3. $(\mathrm{inp}_1, \mathrm{aux}_1) := \mathbf{F}(\mathrm{out}_0)$
4. $H_1 := \mathbf{Repro}(\mathrm{aux}_1, H_0)$

// Reprogramming

5. $\mathrm{out}_1 \leftarrow \mathcal{A}^{H_1}(\mathrm{inp}_1)$ // Second stage
6. return $(\mathrm{out}_0, \mathrm{out}_1)$

QROM Adaptive Reprogramming Framework

Repro Game (for two-stage adversa

1. $\text{inp}_0 \leftarrow$ Initialize
2. $\text{out}_0 \leftarrow \mathcal{A}^{H_0}(\text{inp}_0)$ // Fi stage
3. $(\text{inp}_1, \text{aux}_1) :=$ **F**(out_0)
4. $H_1 :=$ **Repro**(aux_1, H_0)
5. $\text{out}_1 \leftarrow \mathcal{A}^{H_1}(\text{inp}_1)$ // Second stage
6. return $(\text{out}_0, \text{out}_1)$

// Reprogramming

- **F**: Algorithm that generates
  - the input for the next stage of $\mathcal{A}$
  - auxiliary information for reprogramming

- **Repro**: Algorithm that captures reprogramming operations

- We consider **all outputs** from $\mathcal{A}$ during the game

NTNU

QROM Adaptive Reprogramming Framework

Repro Game (for two-stage adversaries)

1. $\mathsf{inp}_0 \leftarrow$ Initialize
2. $\mathsf{out}_0 \leftarrow \mathcal{A}^{H_0}(\mathsf{inp}_0)$
3. $(\mathsf{inp}_1, \mathsf{aux}_1) := \mathbf{F}(\mathsf{out}_0)$
4. $H_1 := \mathbf{Repro}(\mathsf{aux}_1, H_0)$
5. $\mathsf{out}_1 \leftarrow \mathcal{A}^{H_1}(\mathsf{inp}_1)$
6. return $(\mathsf{out}_0, \mathsf{out}_1)$

No-repro Game (for two-stage adversaries)

1. $\mathsf{inp}_0 \leftarrow$ Initialize
2. $\mathsf{out}_0 \leftarrow \mathcal{A}^{H_0}(\mathsf{inp}_0)$
3. $(\mathsf{inp}_1, \mathsf{aux}_1) := \mathbf{F}(\mathsf{out}_0)$
4. $H_1 := H_0$ // no reprogramming
5. $\mathsf{out}_1 \leftarrow \mathcal{A}^{H_1}(\mathsf{inp}_1)$
6. return $(\mathsf{out}_0, \mathsf{out}_1)$

QROM Adaptive Reprogramming Framework

Repro Game (for two-stage adversaries)

1. $\text{inp}_0 \leftarrow$ Initialize
2. $\text{out}_0 \leftarrow \mathcal{A}^{H_0}(\text{inp}_0)$
3. $(\text{inp}_1, \text{aux}_1) := \mathbf{F}(\text{out}_0)$
4. $H_1 := \textbf{Repro}(\text{aux}_1, H_0)$
5. $\text{out}_1 \leftarrow \mathcal{A}^{H_1}(\text{inp}_1)$
6. return $(\text{out}_0, \text{out}_1)$

No-repro Game (for two-stage adversaries)

1. $\text{inp}_0 \leftarrow$ Initialize
2. $\text{out}_0 \leftarrow \mathcal{A}^{H_0}(\text{inp}_0)$
3. $(\text{inp}_1, \text{aux}_1) := \mathbf{F}(\text{out}_0)$
4. $H_1 := H_0$ // no reprogramming
5. $\text{out}_1 \leftarrow \mathcal{A}^{H_1}(\text{inp}_1)$
6. return $(\text{out}_0, \text{out}_1)$

- Let S be the differ set of H_0 and H_1, namely, $\forall x \notin S, H_0(\mathrm{x}) = H_1(x)$.

QROM Adaptive Reprogramming Framework

Repro Game (for two-stage adversaries)

1. $\text{inp}_0 \leftarrow$ Initialize
2. $\text{out}_0 \leftarrow \mathcal{A}^{H_0}(\text{inp}_0)$
3. $(\text{inp}_1, \text{aux}_1) := \mathbf{F}(\text{out}_0)$
4. $H_1 := \mathbf{Repro}(\text{aux}_1, H_0)$
5. $\text{out}_1 \leftarrow \mathcal{A}^{H_1}(\text{inp}_1)$
6. return $(\text{out}_0, \text{out}_1)$

No-repro Game (for two-stage adversaries)

1. $\text{inp}_0 \leftarrow$ Initialize
2. $\text{out}_0 \leftarrow \mathcal{A}^{H_0}(\text{inp}_0)$
3. $(\text{inp}_1, \text{aux}_1) := \mathbf{F}(\text{out}_0)$
4. $H_1 := H_0$ // no reprogramming
5. $\text{out}_1 \leftarrow \mathcal{A}^{H_1}(\text{inp}_1)$
6. return $(\text{out}_0, \text{out}_1)$

- Let S be the differ set of H_0 and H_1, namely, $\forall x \notin S, H_0(\mathrm{x}) = H_1(x)$.
- $\mathcal{A}$ cannot distinguish two games unless it ``queries'' any points in S

  (Can be bounded by adaptive OW2H [Unr14])

NTNU

QROM Adaptive Reprogramming Framework

Repro Game

1. $(H_0$, inp$_0) \leftarrow$ Initialize
2. out$_0 \leftarrow \mathcal{A}^{H_0}$(inp$_0$)
3. For i from 1 to n

   (inp$_i$, aux$_i$) := **F**(out$_{i-1}$)

   H_i := **Repro**(aux$_i$, H_{i-1})

   out$_i \leftarrow \mathcal{A}^{H_i}$(inp$_i$)
4. return (out$_0$, out$_1$, …, out$_n$)

No-repro Game

1. $(H$, inp$_0) \leftarrow$ Initialize
2. out$_0 \leftarrow \mathcal{A}^{H}$(inp$_0$)
3. For i from 1 to n

   (inp$_i$, aux$_i$) := **F**(out$_{i-1}$)

   H := H // no reprogramming

   out$_i \leftarrow \mathcal{A}^{H}$(inp$_i$)
4. return (out$_0$, out$_1$, …, out$_n$)

- Let S_i be the differ sets of H_0 and H_i, namely, $\forall x \notin S_i, H(\mathrm{x}) = H_i(x)$.

QROM Adaptive Reprogramming Framework

Repro Game

1. $(H_0$, $\text{inp}_0) \leftarrow$ Initialize
2. $\text{out}_0 \leftarrow \mathcal{A}^{H_0}(\text{inp}_0)$
3. For i from 1 to n

   $(\text{inp}_i, \text{aux}_i) := \mathbf{F}(\text{out}_{i-1})$

   $H_i \coloneqq \mathbf{Repro}(\text{aux}_i, H_{i-1})$

   $\text{out}_i \leftarrow \mathcal{A}^{H_i}(\text{inp}_i)$
4. return $(\text{out}_0, \text{out}_1, \ldots, \text{out}_n)$

No-repro Game

1. $(H$, $\text{inp}_0) \leftarrow$ Initialize
2. $\text{out}_0 \leftarrow \mathcal{A}^{H}(\text{inp}_0)$
3. For i from 1 to n

   $(\text{inp}_i, \text{aux}_i) := \mathbf{F}(\text{out}_{i-1})$

   $H \coloneqq H$ // no reprogramming

   $\text{out}_i \leftarrow \mathcal{A}^{H}(\text{inp}_i)$
4. return $(\text{out}_0, \text{out}_1, \ldots, \text{out}_n)$

- Let S_i be the differ sets of H_0 and H_i, namely, $\forall x \notin S_i, H(\mathrm{x}) = H_i(x)$.
- **A straight-forward "Hybrid argument + adaptive OW2H" proof does not work…**
  - …since S_i's can be co-related, the game consider all outputs from $\mathcal{A}$,…

QROM Adaptive Reprogramming Framework

Repro Game

1. $(H_0, \text{inp}_0) \leftarrow$ Initialize
2. $\text{out}_0 \leftarrow \mathcal{A}^{H_0}(\text{inp}_0)$
3. For i from 1 to n

   $(\text{inp}_i, \text{aux}_i) := \mathbf{F}(\text{out}_{i-1})$

   $H_i := \mathbf{Repro}(\text{aux}_i, H_{i-1})$

   $\text{out}_i \leftarrow \mathcal{A}^{H_i}(\text{inp}_i)$
4. return $(\text{out}_0, \text{out}_1, \ldots, \text{out}_n)$

No-repro Game

1. $(H, \text{inp}_0) \leftarrow$ Initialize
2. $\text{out}_0 \leftarrow \mathcal{A}^{H}(\text{inp}_0)$
3. For i from 1 to n

   $(\text{inp}_i, \text{aux}_i) := \mathbf{F}(\text{out}_{i-1})$

   $H := H$ // no reprogramming

   $\text{out}_i \leftarrow \mathcal{A}^{H}(\text{inp}_i)$
4. return $(\text{out}_0, \text{out}_1, \ldots, \text{out}_n)$

- Reprogram n times, $\mathcal{A}$ queries QRO q times
- $|\Pr\{\mathcal{A}'s \text{ "behavior" in Repro Game}\} - \Pr\{\mathcal{A}'s \text{ "behavior" in No-repro Game}\}|$

$$\leq \boldsymbol{O}(\boldsymbol{n}^2\boldsymbol{q}) \cdot \sqrt{\Pr\{\ldots \text{ a reduction "extracts" a point in } S_i \ldots\}}$$

QROM Adaptive Reprogramming Framework

Repro Game

1. $(H_0$, inp$_0$) ← Initialize
2. out$_0$ ← $\mathcal{A}^{H_0}$(inp$_0$)
3. For i from 1 to n

   (inp$_i$, aux$_i$) := **F**(out$_{i-1}$)

   H_i ≔ **Repro**(aux$_i$, H_{i-1})

   out$_i$ ← $\mathcal{A}^{H_i}$(inp$_i$)
4. return (out$_0$, out$_1$, …, out$_n$)

No-repro Game

1. (H, inp$_0$) ← Initialize
2. out$_0$ ← $\mathcal{A}^{H}$(inp$_0$)
3. For i from 1 to n

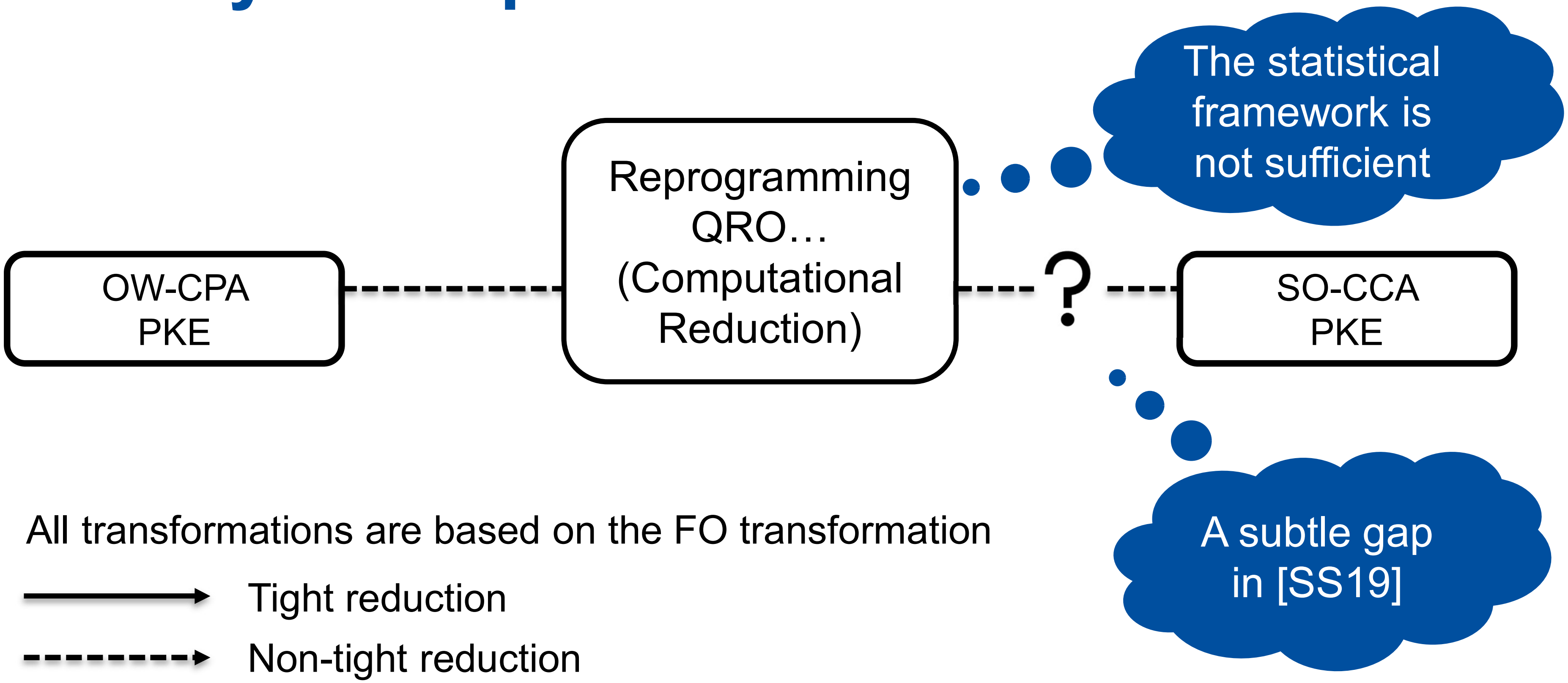
   (inp$_i$, aux$_i$) := **F**(out$_{i-1}$)

   H ≔ H // no reprogramming

   out$_i$ ← $\mathcal{A}^{H}$(inp$_i$)
4. return (out$_0$, out$_1$, …, out$_n$)

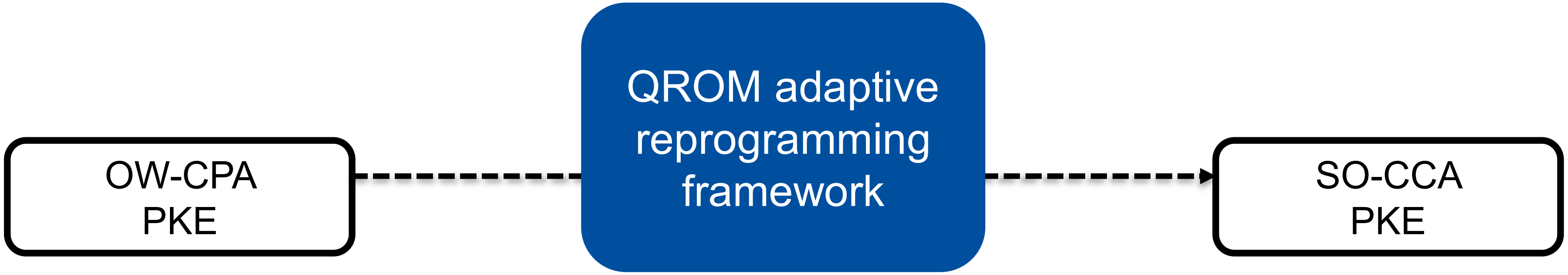
- Such reduction has a similar running time with $\mathcal{A}$…
  - …which allows us to construct **computational reductions**…
  - … v.s. the framework in [GHHM21]: Computational (ours) v.s. Statistical

Summary and Open Problems



NTNU

Summary and Open Problems



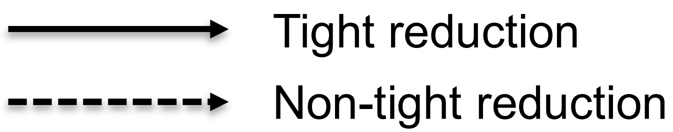
All transformations are based on the FO transformation

Tight reduction

Non-tight reduction

Summary and Open Problems

All transformations are based on the FO transformation



Summary and Open Problems