Almost Tight Multi-User Security under Adaptive Corruptions from LWE in the Standard Model

Shuai Han, **Shengli Liu**, Zhedong Wang, Dawu Gu

Shanghai Jiao Tong University

Crypto 2023, Santa Barbara, USA

Contents

1 Almost Tight MU^c Security & Our Contributions

2 Technical Tool: Probabilistic Hash Proof System

3 Overview of Our PKE and SIG Constructions

Contents

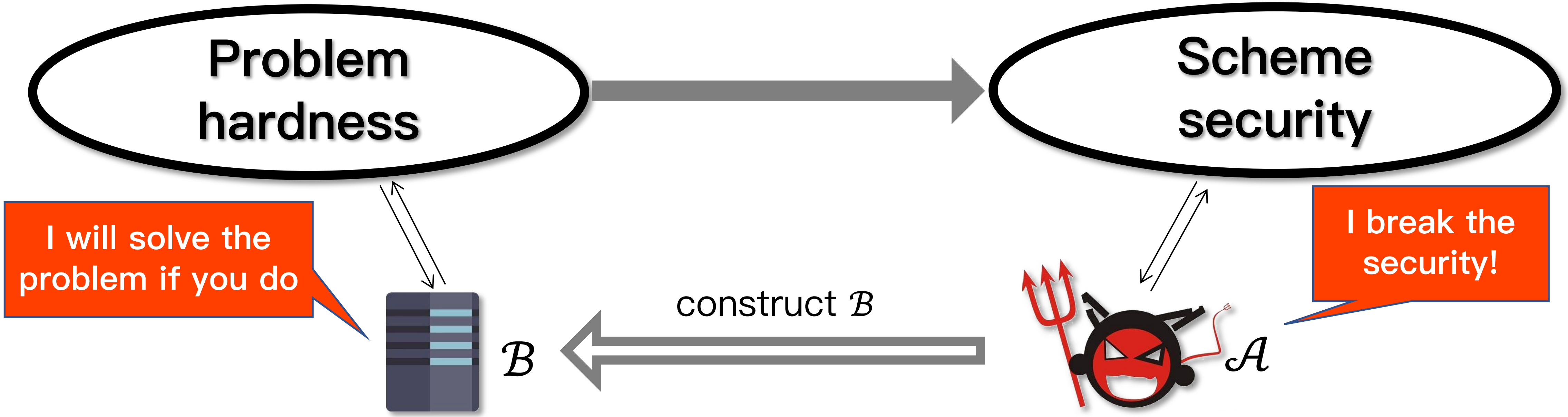
1 Almost Tight MU^c Security & Our Contributions

2 Technical Tool: Probabilistic Hash Proof System

3 Overview of Our PKE and SIG Constructions

Almost Tight Security

Security of a cryptographic **Scheme** based on a hard **Problem**.



Problem hardness

Scheme security

I will solve the problem if you do

I break the security!

construct $\mathcal{B}$

$\mathcal{B}$

$\mathcal{A}$

solving **Problem** in time $t_\mathcal{B}$ with advantage $\epsilon_\mathcal{B}$

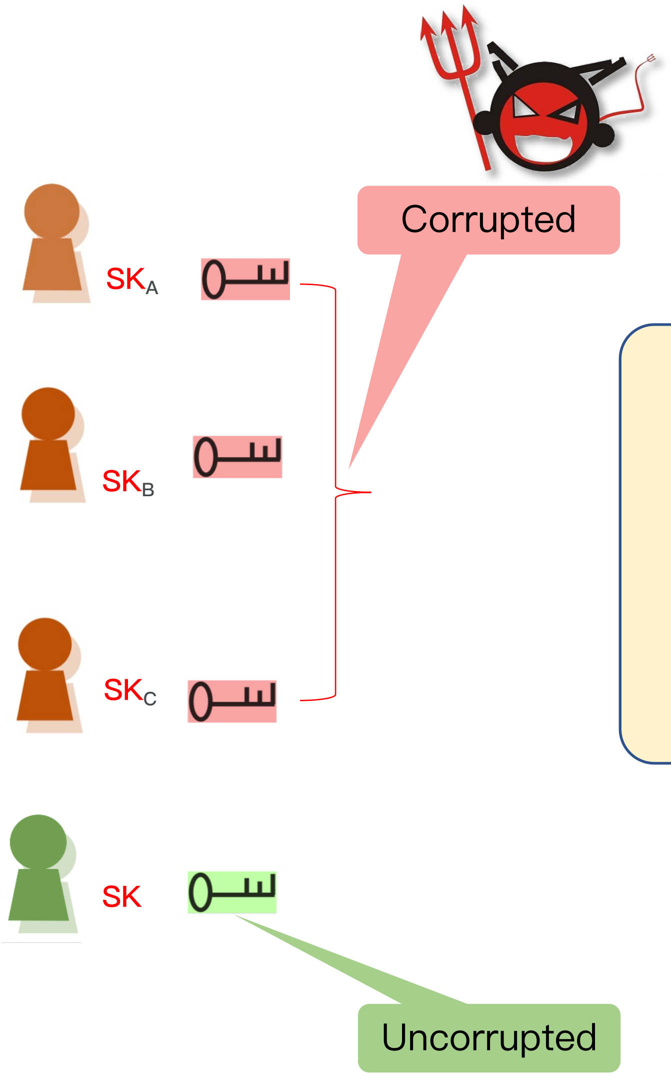
attacking **Scheme** in time $t_\mathcal{A}$ with advantage $\epsilon_\mathcal{A}$

$$\frac{t_\mathcal{B}}{\epsilon_\mathcal{B}} \leq \frac{t_\mathcal{A}}{\epsilon_\mathcal{A}} \cdot \ell$$

(Almost) Tight Security: ℓ = O(1) or poly(λ),

where λ = security parameter

Multi-User Security under Adaptive Corruptions (MU^c Security)

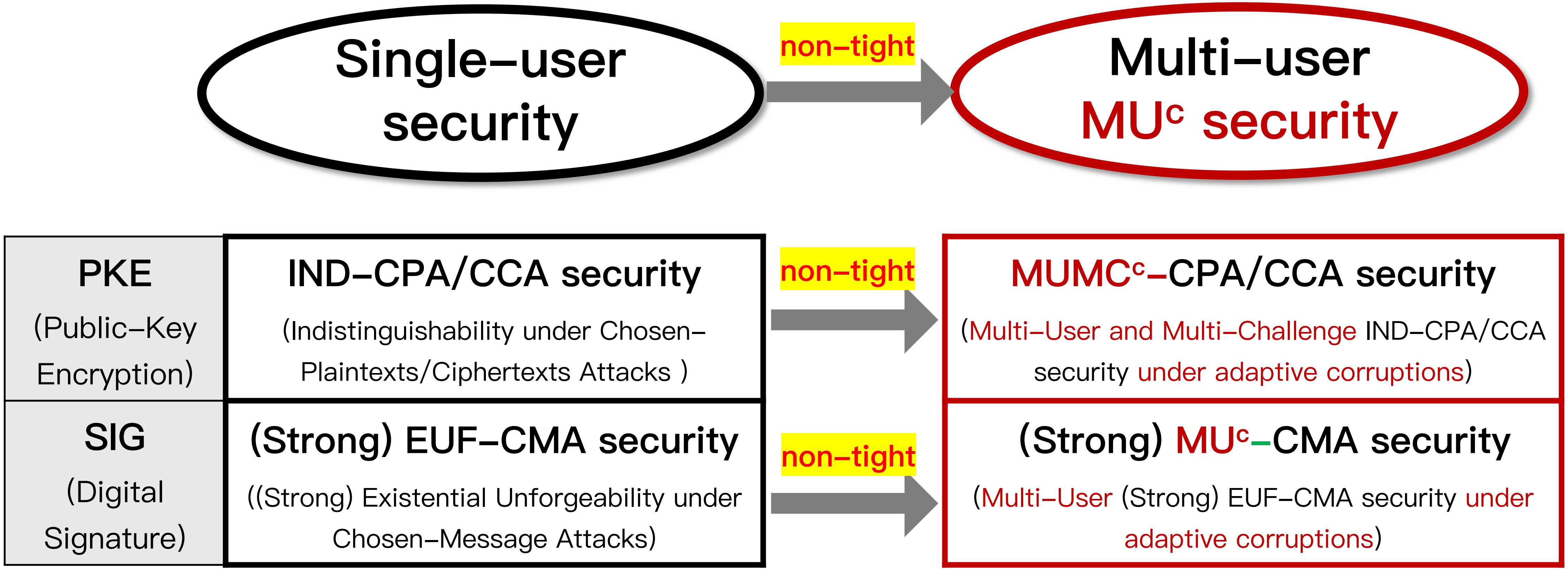


Given the adaptably corrupted keys

Protect security of the uncorrupted users

MU^c security

On Achieving Tight MU^c Security



Single–user security → non–tight → Multi–user MU^c security

PKE (Public–Key Encryption)	IND–CPA/CCA security (Indistinguishability under Chosen–Plaintexts/Ciphertexts Attacks)	non–tight →	MUMC^c–CPA/CCA security (Multi–User and Multi–Challenge IND–CPA/CCA security under adaptive corruptions)
SIG (Digital Signature)	(Strong) EUF–CMA security ((Strong) Existential Unforgeability under Chosen–Message Attacks)	non–tight →	(Strong) MU^c–CMA security (Multi–User (Strong) EUF–CMA security under adaptive corruptions)

Non–tight reduction!

$\ell \geqslant$ #users, #ciphertexts, or #signatures

On Achieving Tight MU^c Security: Impossibility Results

- Public–Key Encryption (PKE): **Tight MUMC^c–CPA/CCA security**
  - the relation (pk, sk) is "unique”
  - the relation (pk, sk) is "re–ran

Impossible !

- Digital Signature (SIG): **Tight (Strong) MU^c–CMA securit**
  - the signing algorithm is deterministic

Impossible !

On Achieving Tight MU^c Security: Possibility Results

PKE	Std/RO model?	MU^c Security?	Security Loss	Assumption	Post-Quantum?
[LLP20, DCC]	classical RO	✓	O(1)	CDH	×
[HLG23, EC]	Std	✓	O(log λ)	MDDH	×

- based on number-theoretic assumptions.

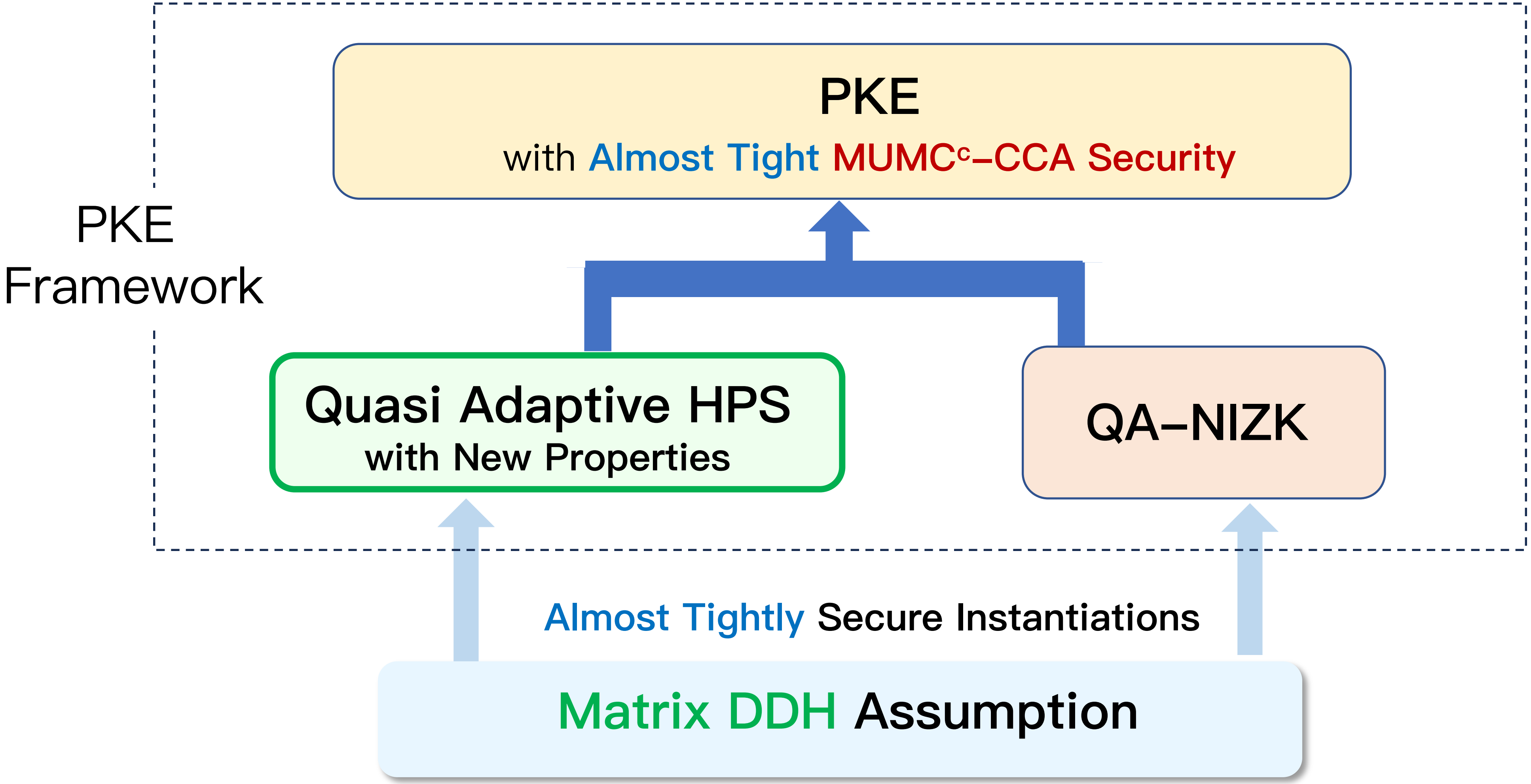
SIG	Std/RO model?	MU^c Security?	Security Loss	Assumption	Post-Quantum?
[BHJKL15, TCC]	Std	✓	O(1)	MDDH	×
[GJ18, C]	classical RO	✓	O(1)	DDH	×
[DGJL21, PKC]	classical RO	✓	O(1)	DDH/Φ-hiding	×
[HJKLPRS21, C]	Std	✓	O(λ)	MDDH	×
[PW22, PKC]	classical RO	✓	O(1)	LWE	✓
[HLG23, EC]	Std	✓	O(log λ)	MDDH	×

- either based on number-theoretic assumptions,
- or in classical RO model.

Vulnerable to Quantum

Can we achieve (almost) tight MU^c security based on LWE in the standard model?

[HLG23]: PKE with Almost Tight MU^c Security from MDDH in the Std Model



PKE

with Almost Tight MUMC^c–CCA Security

PKE Framework

Quasi Adaptive HPS

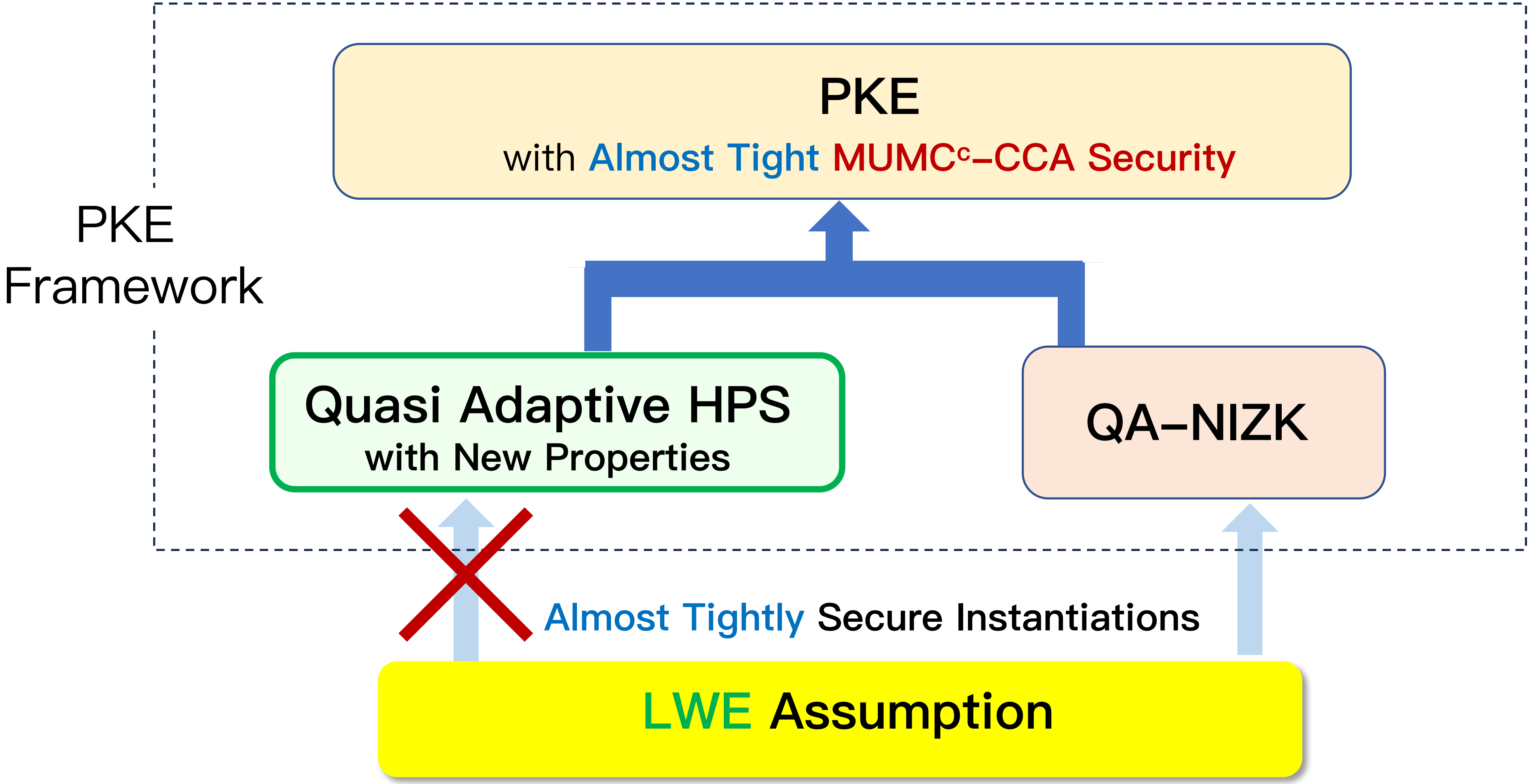
with New Properties

QA–NIZK

Almost Tightly Secure Instantiations

Matrix DDH Assumption

PKE with Almost Tight MU^c Security from LWE in the Std Model ?



PKE

with Almost Tight MUMC^c–CCA Security

PKE Framework

Quasi Adaptive HPS

with New Properties

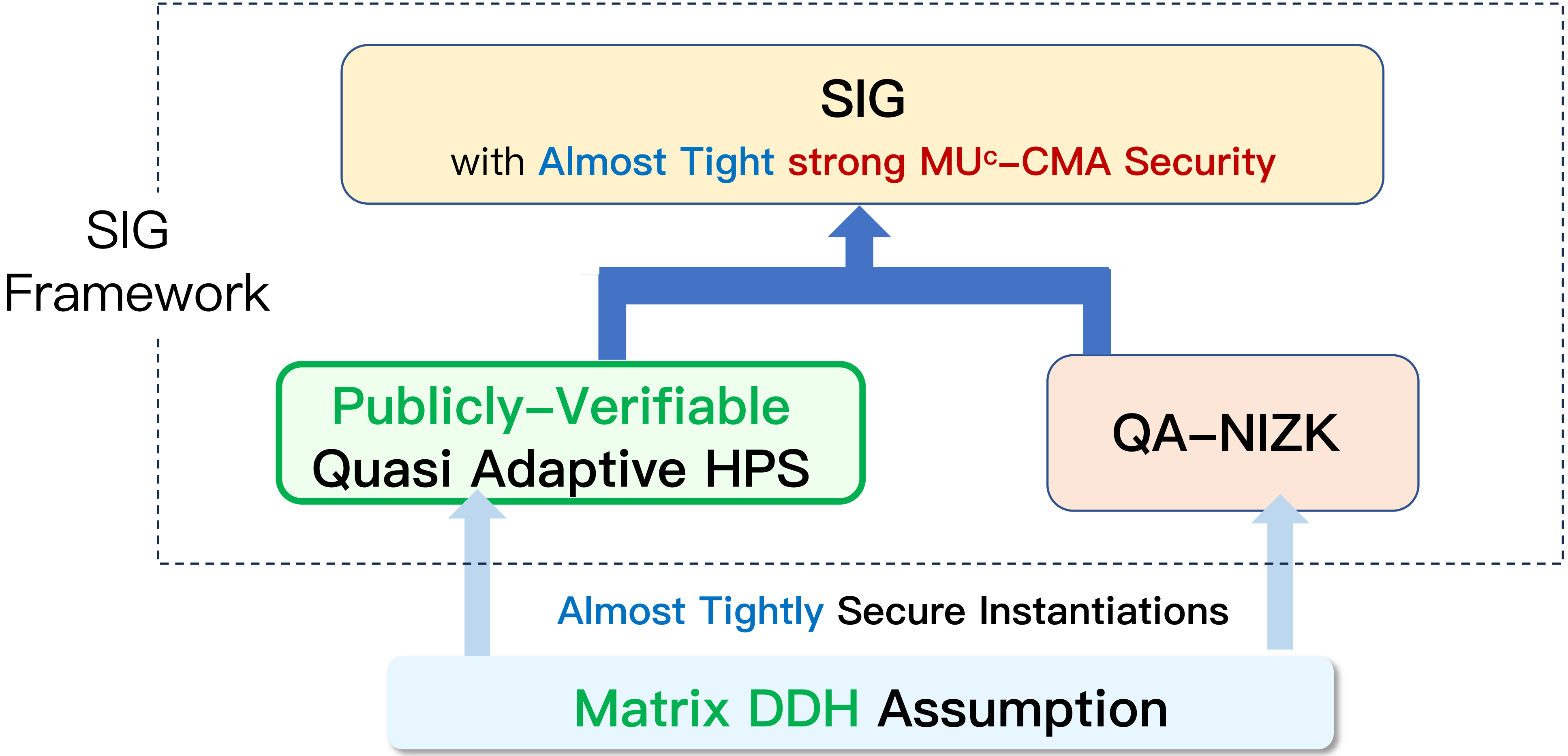
QA–NIZK

Almost Tightly Secure Instantiations

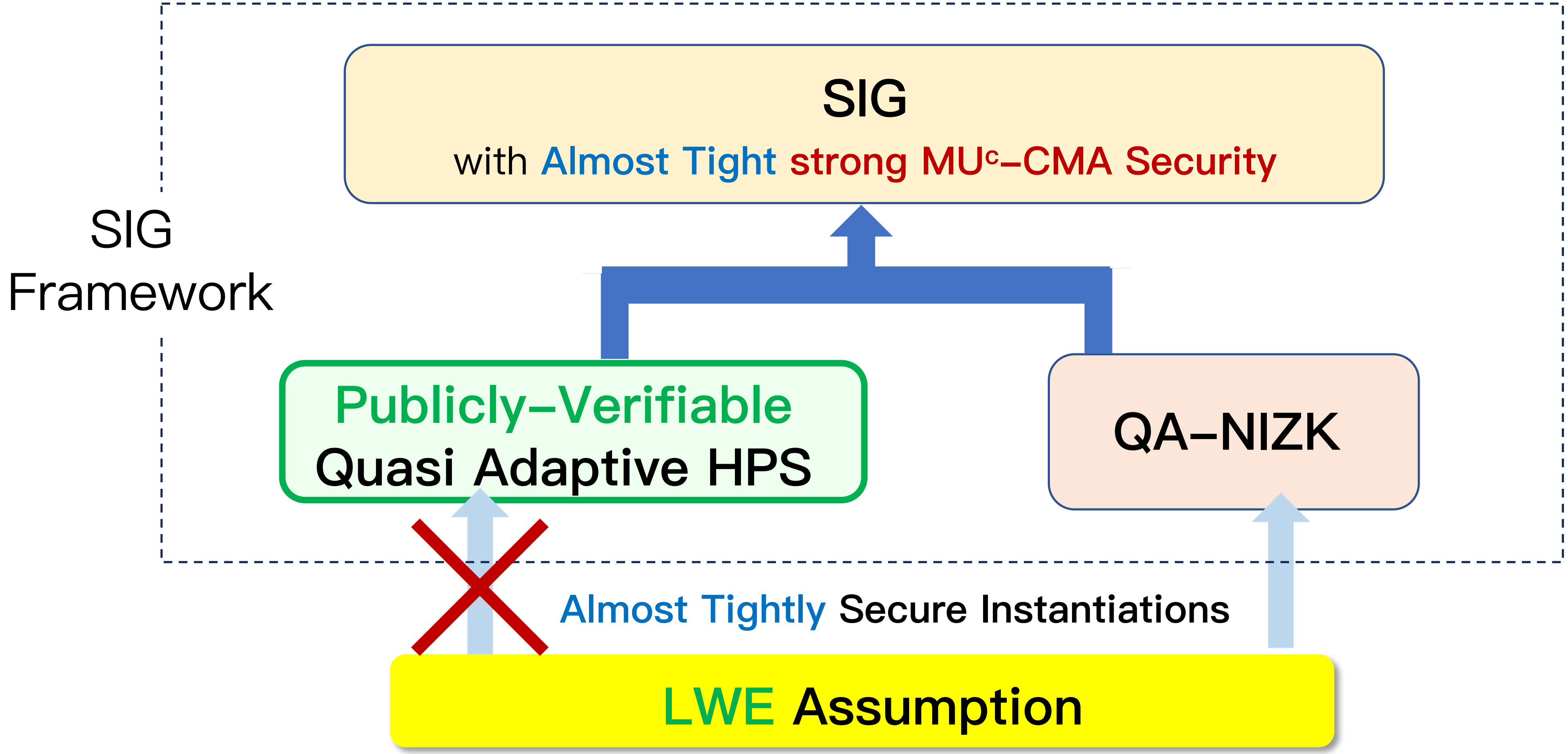
LWE Assumption

Can we achieve (almost) tight MU^c security based on LWE in the standard model?

[HLG23]: SIG with Almost Tight MU^c Security from MDDH in the Std Model


SIG
with Almost Tight strong MU^c–CMA Security
SIG Framework
Publicly–Verifiable Quasi Adaptive HPS
QA–NIZK
Almost Tightly Secure Instantiations
Matrix DDH Assumption

SIG with Almost Tight MU^c Security from LWE in the Std Model ?



SIG

with Almost Tight strong MU^c–CMA Security

SIG Framework

Publicly–Verifiable Quasi Adaptive HPS

QA–NIZK

Almost Tightly Secure Instantiations

LWE Assumption

Can we achieve (almost) tight MU^c security based on LWE in the standard model?

Contribution: Almost Tight MU^c Security from LWE in the Std Model

PKE	Std/RO model?	MU^c Security?	Security Loss	Assumption	Post-Quantum?
[LLP20, DCC]	classical RO	✓	O(1)	CDH	×
[HLG23, EC]	Std	✓	O(log λ)	MDDH	×
Ours	**Std**	✓	**O(λ^2)**	**LWE**	✓

- The *first* LWE-based PKE scheme with almost tight MUMC^c-CCA security in the standard model

SIG	Std/RO model?	MU^c Security?	Security Loss	Assumption	Post-Quantum?
[BHJKL15, TCC]	Std	✓	O(1)	MDDH	×
[GJ18, C]	classical RO	✓	O(1)	DDH	×
[DGJL21, PKC]	classical RO	✓	O(1)	DDH/Φ-hiding	×
[HJKLPRS21, C]	Std	✓	O(λ)	MDDH	×
[PW22, PKC]	classical RO	✓	O(1)	LWE	✓
[HLG23, EC]	Std	✓	O(log λ)	MDDH	×
Ours	**Std**	✓	**O(λ^2)**	**LWE**	✓

- The *first* LWE-based SIG scheme with almost tight MU^c-CMA security in the standard model

Contents

1 Almost Tight MU^c Security & Our Contributions

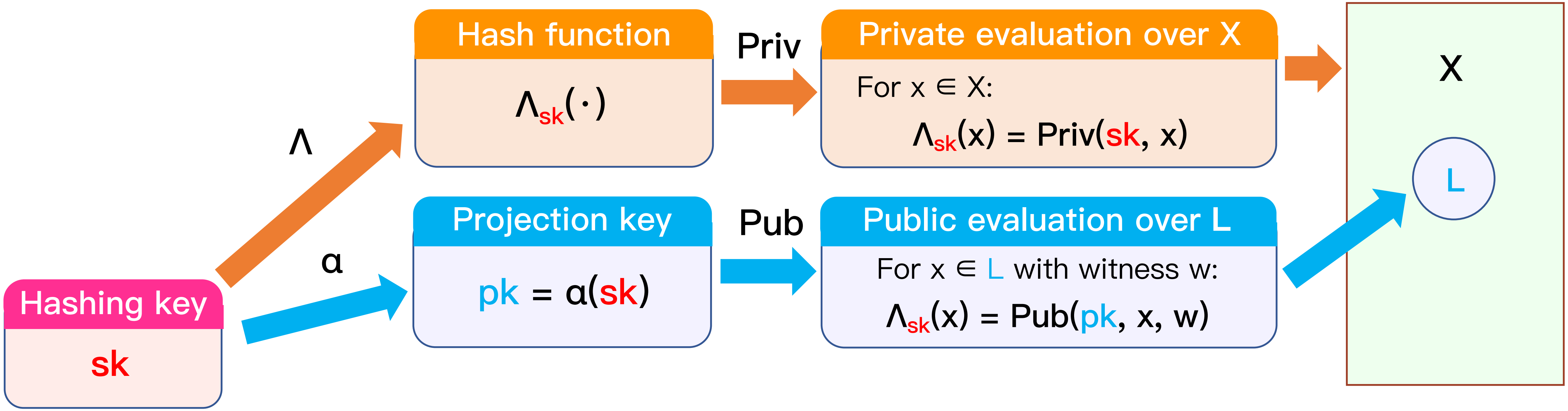
2 Technical Tool: Probabilistic Hash Proof System

3 Overview of Our PKE and SIG Constructions

Recap: Hash Proof System [Cramer–Shoup, EC02]

HPS = (Λ, α, Priv, Pub, X, L)

SMP： $X \approx_c L$

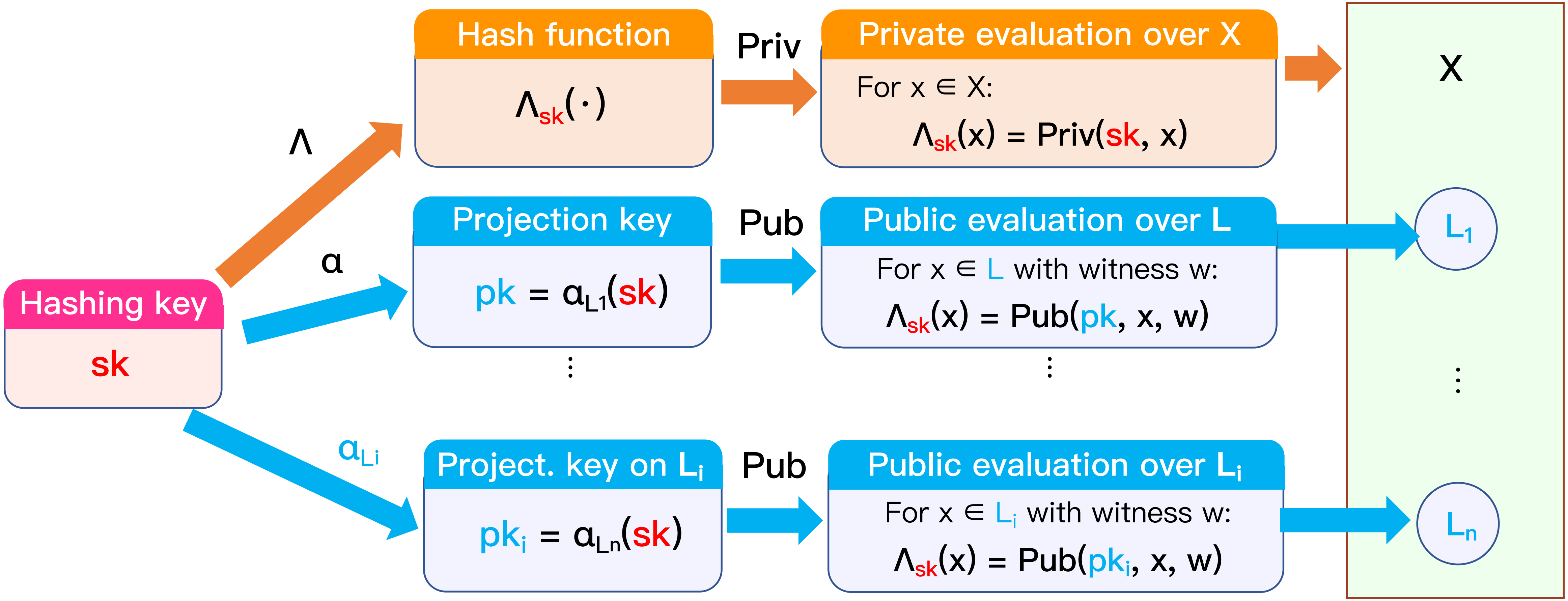


Hash function
$\Lambda_{sk}(\cdot)$
Priv
Private evaluation over X
For $x \in X$:
$\Lambda_{sk}(x) = \text{Priv}(sk, x)$
X
L
Λ
α
Hashing key
sk
Projection key
$pk = \alpha(sk)$
Pub
Public evaluation over L
For $x \in L$ with witness w:
$\Lambda_{sk}(x) = \text{Pub}(pk, x, w)$

- **(Exact) Correctness:** requires $\text{Priv}(sk, x) = \text{Pub}(pk, x, w)$ for $x \in L$.

Recap: Quasi–Adaptive HPS [Han–Liu–Lyu–Gu, C19]

QA–HPS = (Λ, $\alpha_{(\cdot)}$, Priv, Pub, X, $\{L_i\}$) SMP： X $\approx_c$ L_i



Hashing key

sk

Λ

α

α_{Li}

Hash function

$\Lambda_{sk}(\cdot)$

Priv

Private evaluation over X

For x ∈ X:

$\Lambda_{sk}(x)$ = Priv(sk, x)

Projection key

pk = α_{L1}(sk)

Pub

Public evaluation over L

For x ∈ L with witness w:

$\Lambda_{sk}(x)$ = Pub(pk, x, w)

⋮

Project. key on L_i

pk_i = α_{Ln}(sk)

Public evaluation over L_i

For x ∈ L_i with witness w:

$\Lambda_{sk}(x)$ = Pub(pk_i, x, w)

- **(Exact) Correctness:** requires Priv(sk, x) = Pub(pk, x, s) for x ∈ L .
- **Key Switching**: (α_{L0}(sk), α_{L1}(sk)) $\approx_s$ (α_{L0}(sk), α_{L1}(sk’))

Obstacle: No LWE-based HPS with Exact Correctness

$$\mathcal{X} = \{\mathbf{c} \mid \mathbf{c} \in \mathbb{Z}_q^m\}$$

Languages are LWE samples:

$$\mathcal{L}_{\mathbf{A}} := \{\mathbf{c} = \mathbf{A}^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\mathcal{L}_{\mathbf{A}_1} := \{\mathbf{c} = \mathbf{A}_1^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\mathcal{L}_{\mathbf{A}_2} := \{\mathbf{c} = \mathbf{A}_2^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\cdots \qquad \cdots$$

Secret&Projection Key: $sk = \mathbf{k} \in \{0,1\}^m, \quad pk_{\mathbf{A}} := \alpha_{\mathbf{A}}(\mathbf{k}) = \mathbf{A}^\top \mathbf{k}.$

Private evaluation: $\mathsf{Priv}(\mathbf{k}, \mathbf{c}) = \Lambda_{\mathbf{k}}(\mathbf{c}) := \mathbf{c}^\top \mathbf{k} \in \mathbb{Z}_q$

$$= (\mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top)\mathbf{k} \;=\; \boxed{\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}} + \mathbf{e}^\top \mathbf{k} \quad \text{for } \mathbf{c} \in \mathcal{L}_{\mathbf{A}}$$

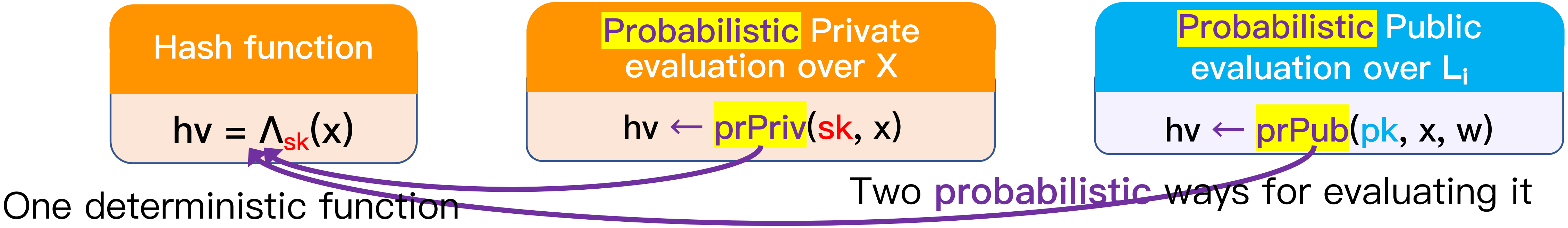
Public evaluation: $\mathsf{Pub}(pk_{\mathbf{A}}, \mathbf{c}, \mathbf{s}, \mathbf{e}) = \mathbf{s}^\top \cdot pk_{\mathbf{A}} = \boxed{\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}}$

Priv(sk, x) ≈ Pub(pk, x, s) but Priv(sk, x) ≠ Pub(pk, x, s) !

Our Solution to the Obstacle: pr–QA–HPS

Probabilistic QA–HPS:

- Probabilistic public evaluation: prPriv(sk, x)
- Probabilistic private evaluation: prPub(pk, x, s)



Hash function

$hv = \Lambda_{sk}(x)$

One deterministic function

Probabilistic Private evaluation over X

$hv \leftarrow \text{prPriv}(sk, x)$

Probabilistic Public evaluation over L_i

$hv \leftarrow \text{prPub}(pk, x, w)$

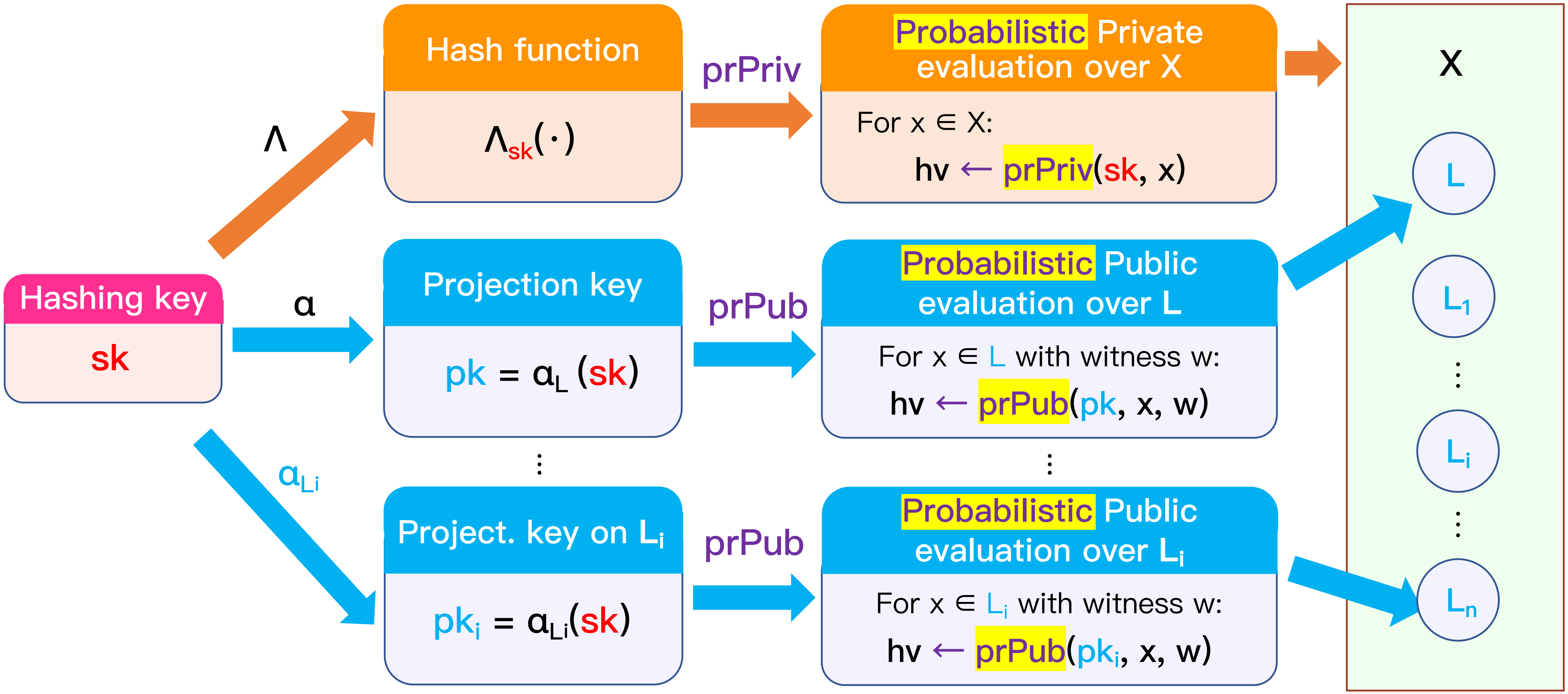
Two probabilistic ways for evaluating it

- **Approximate Correctness:** $\text{prPriv}(sk, x) \approx \Lambda_{sk}(x) \approx \text{prPub}(pk\ x, w)$
- **Evaluation st. Indistinguishability:** $\text{prPriv}(sk, x) \approx_s \text{prPub}(pk, x, w)$ given sk
- **Key Switching**: $(\alpha_{L0}(sk), \alpha_{L1}(sk)) \approx_s (\alpha_{L0}(sk), \alpha_{L1}(sk'))$

Our New Tool: Probabilistic QA–HPS

$\text{pr–QA–HPS} = (\Lambda, \alpha_{(\cdot)}, \text{prPriv}, \text{prPub}, X, \{L_i\})$

SMP： $X \approx_c L_i$



Hashing key

sk

Λ

α

α_{L_i}

Hash function

$\Lambda_{sk}(\cdot)$

Projection key

$pk = \alpha_L(sk)$

⋮

Project. key on L_i

$pk_i = \alpha_{L_i}(sk)$

prPriv

prPub

prPub

Probabilistic Private evaluation over X

For $x \in X$:

$hv \leftarrow \text{prPriv}(sk, x)$

Probabilistic Public evaluation over L

For $x \in L$ with witness w:

$hv \leftarrow \text{prPub}(pk, x, w)$

⋮

Probabilistic Public evaluation over L_i

For $x \in L_i$ with witness w:

$hv \leftarrow \text{prPub}(pk_i, x, w)$

pr–QA–HPS from LWE

Languages are LWE samples:
with
Subset Mempership Problem

$$\mathcal{L}_{\mathbf{A}} := \{\mathbf{c} = \mathbf{A}^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\mathcal{L}_{\mathbf{A}_1} := \{\mathbf{c} = \mathbf{A}_1^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\mathcal{L}_{\mathbf{A}_2} := \{\mathbf{c} = \mathbf{A}_2^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$x \leftarrow_{\$} \mathcal{L}_{\mathbf{A}_1} \approx_c x \leftarrow_{\$} \mathcal{X} \approx_c x \leftarrow_{\$} \mathcal{L}_{\mathbf{A}_2}$$

L

pr–QA–HPS from LWE

Languages are LWE samples:
with
Subset Mempership Problem

$$\mathcal{L}_{\mathbf{A}} := \{\mathbf{c} = \mathbf{A}^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\mathcal{L}_{\mathbf{A}_1} := \{\mathbf{c} = \mathbf{A}_1^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\mathcal{L}_{\mathbf{A}_2} := \{\mathbf{c} = \mathbf{A}_2^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\cdots \qquad \cdots$$

Secret&Projection Key:

$$sk = \mathbf{k} \in \{0,1\}^m, \quad pk_{\mathbf{A}} := \alpha_{\mathbf{A}}(\mathbf{k}) = \mathbf{A}^\top \mathbf{k}.$$

$$\Lambda_{\mathbf{k}}(\mathbf{c}) := \mathbf{c}^\top \mathbf{k} = \boxed{\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}}$$

Error smuging

Private evaluation:

$$\mathsf{Priv}(\mathbf{k}, \mathbf{c}) = \mathbf{c}^\top \mathbf{k} + \boxed{e'} = \boxed{\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}} + \mathbf{e}^\top \mathbf{k} + \boxed{e'}$$

Public evaluation:

$$\mathsf{Pub}(pk_{\mathbf{A}}, \mathbf{c}, \mathbf{s}, \mathbf{e}) = \mathbf{s}^\top \cdot pk_{\mathbf{A}} + \boxed{e'} = \boxed{\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}} + \boxed{e'}$$

Close & Evaluation Indistinguishability

pr–QA–HPS from LWE

Languages are LWE samples:
with
Subset Mempership Problem

$$\mathcal{L}_{\mathbf{A}} := \{\mathbf{c} = \mathbf{A}^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\mathcal{L}_{\mathbf{A}_1} := \{\mathbf{c} = \mathbf{A}_1^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\mathcal{L}_{\mathbf{A}_2} := \{\mathbf{c} = \mathbf{A}_2^\top \mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}.$$

$$\cdots \qquad \cdots$$

Secret&Projection Key: $sk = \mathbf{k} \in \{0,1\}^m, \quad pk_{\mathbf{A}} := \alpha_{\mathbf{A}}(\mathbf{k}) = \mathbf{A}^\top \mathbf{k}.$

$$\Lambda_{\mathbf{k}}(\mathbf{c}) := \mathbf{c}^\top \mathbf{k} = \boxed{\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}}$$

Private evaluation: $\mathsf{Priv}(\mathbf{k}, \mathbf{c}) = \mathbf{c}^\top \mathbf{k} + \boxed{\boxed{e'}} = \boxed{\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}} + \mathbf{e}^\top \mathbf{k} + \boxed{\boxed{e'}}$

Public evaluation: $\mathsf{Pub}(pk_{\mathbf{A}}, \mathbf{c}, \mathbf{s}, \mathbf{e}) = \mathbf{s}^\top \cdot pk_{\mathbf{A}} + \boxed{\boxed{e'}} = \boxed{\mathbf{s}^\top \mathbf{A}^\top \mathbf{k}} + \boxed{\boxed{e'}}$

Key Switching:

$$(\alpha_{\mathbf{A}_1}(\mathbf{k}), \alpha_{\mathbf{A}_2}(\mathbf{k})) = (\mathbf{A}_1^\top \mathbf{k}, \mathbf{A}_2^\top \mathbf{k}) \approx_s (\mathbf{A}_1^\top \mathbf{k}, \$) \approx_s (\mathbf{A}_1^\top \mathbf{k}, \mathbf{A}_2^\top \mathbf{k}') = (\alpha_{\mathbf{A}_1}(\mathbf{k}), \alpha_{\mathbf{A}_2}(\mathbf{k}'))$$

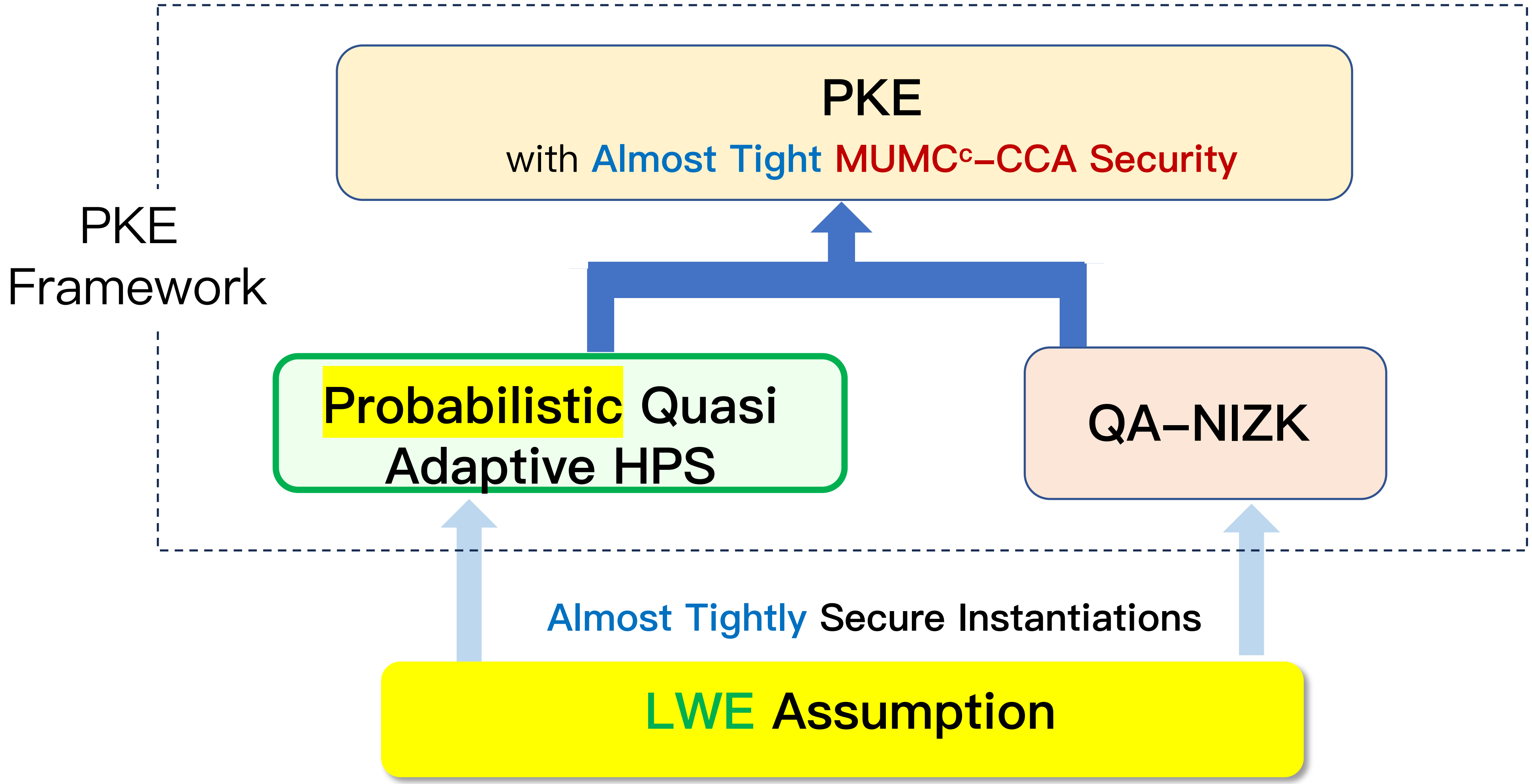
Contents

1 Almost Tight MU^{c} Security & Our Contributions

2 Technical Tool: Probabilistic Hash Proof System

3 Overview of Our PKE and SIG Constructions

PKE with Almost Tight MU^c Security from LWE in the Std Model


PKE
with Almost Tight MUMC^c–CCA Security
PKE Framework
Probabilistic Quasi Adaptive HPS
QA–NIZK
Almost Tightly Secure Instantiations
LWE Assumption

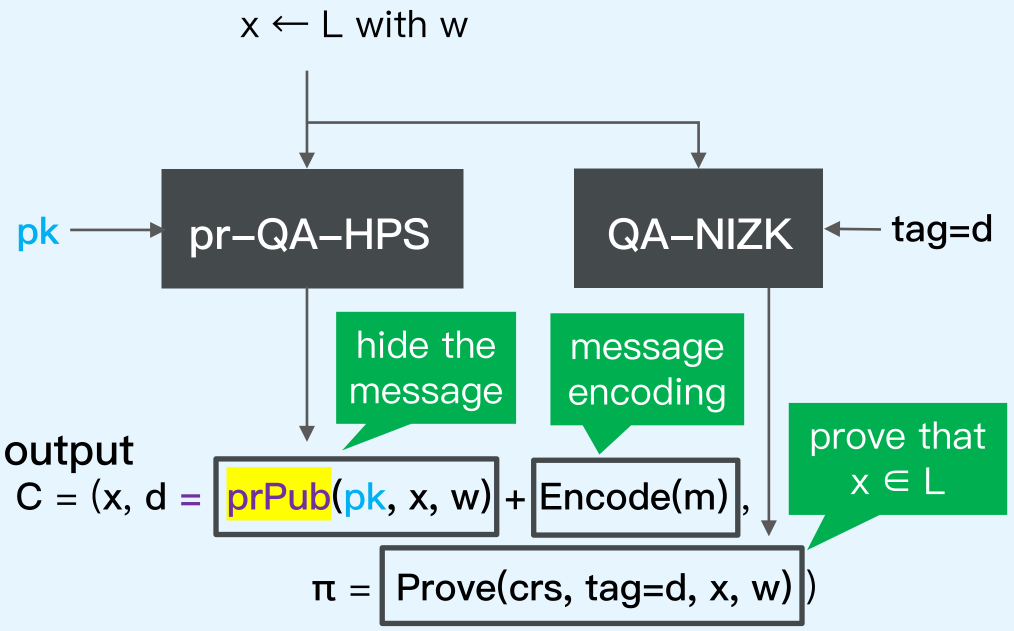
Yes, we can achieve (almost) tight MU^c security for PKE based on LWE in the standard model?

Our PKE with Almost Tight MUMC$^{c\&l}$–CCA security

Probabilistic QA–HPS + QA–NIZK + Error Correcting Code

Gen → ($pk = \alpha_L(sk)$, sk) : Projection key on L and Hashing key of QA–HPS

Enc(pk, m):

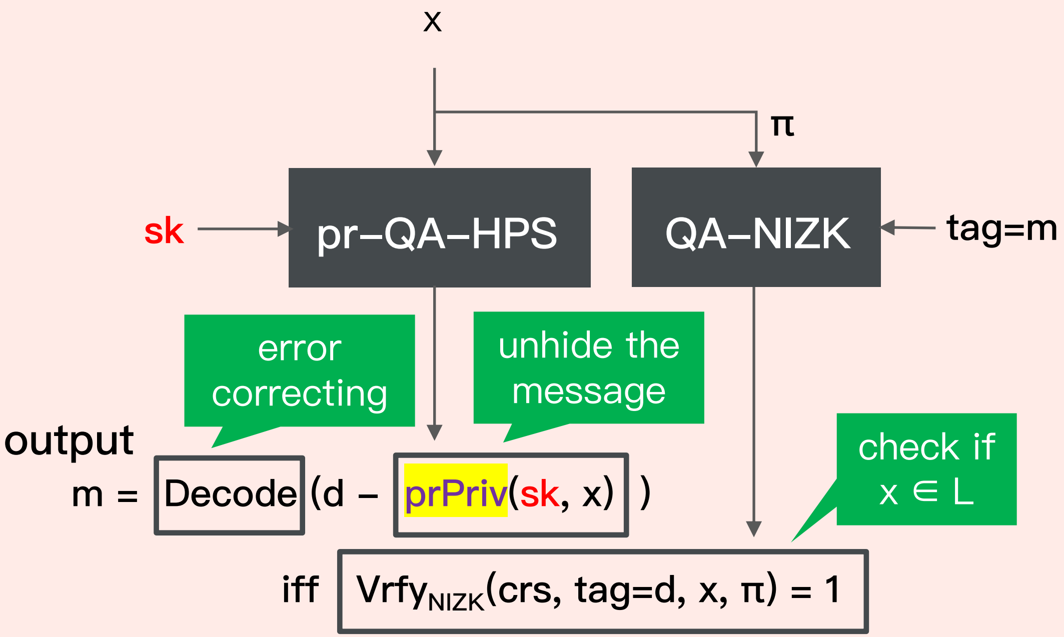


output

C = (x, d = prPub(pk, x, w) + Encode(m),

π = Prove(crs, tag=d, x, w))

Dec(sk, C = (x, d, π)):

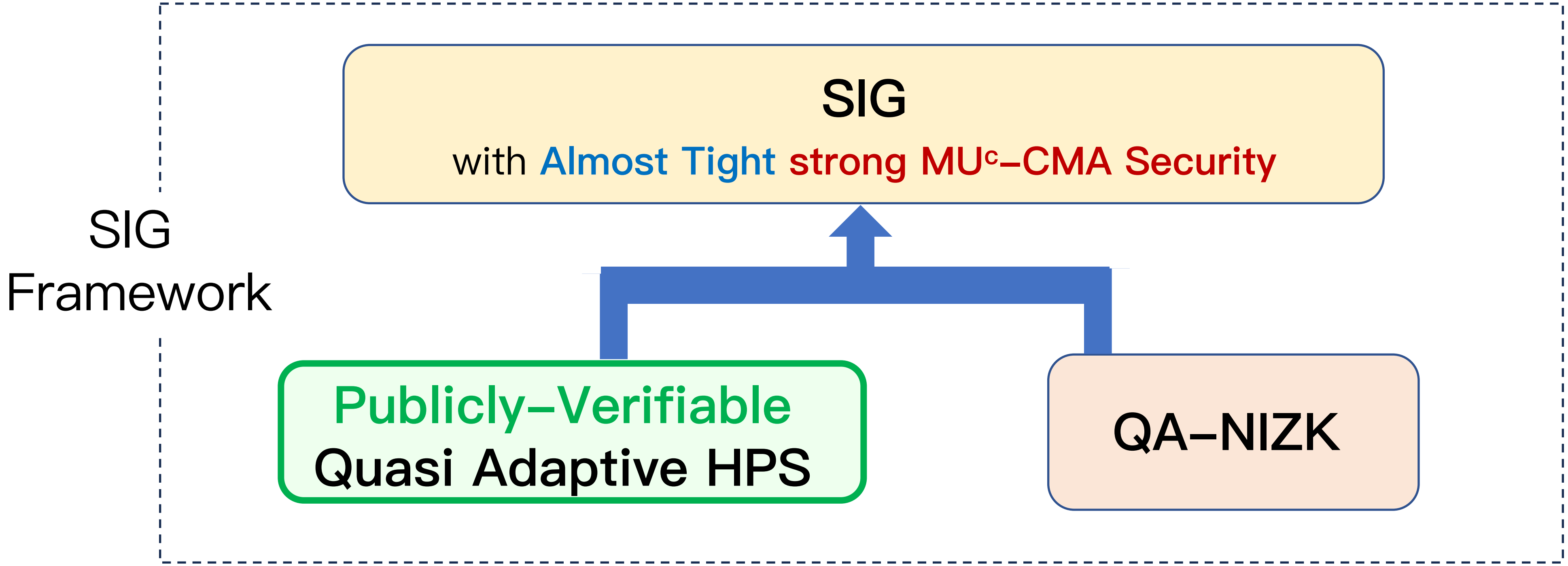


output

m = Decode (d − prPriv(sk, x))

iff Vrfy_{NIZK}(crs, tag=d, x, π) = 1

[HLG23]: SIG with Almost Tight MU^c Security from MDDH in the Std Model



SIG Framework

SIG

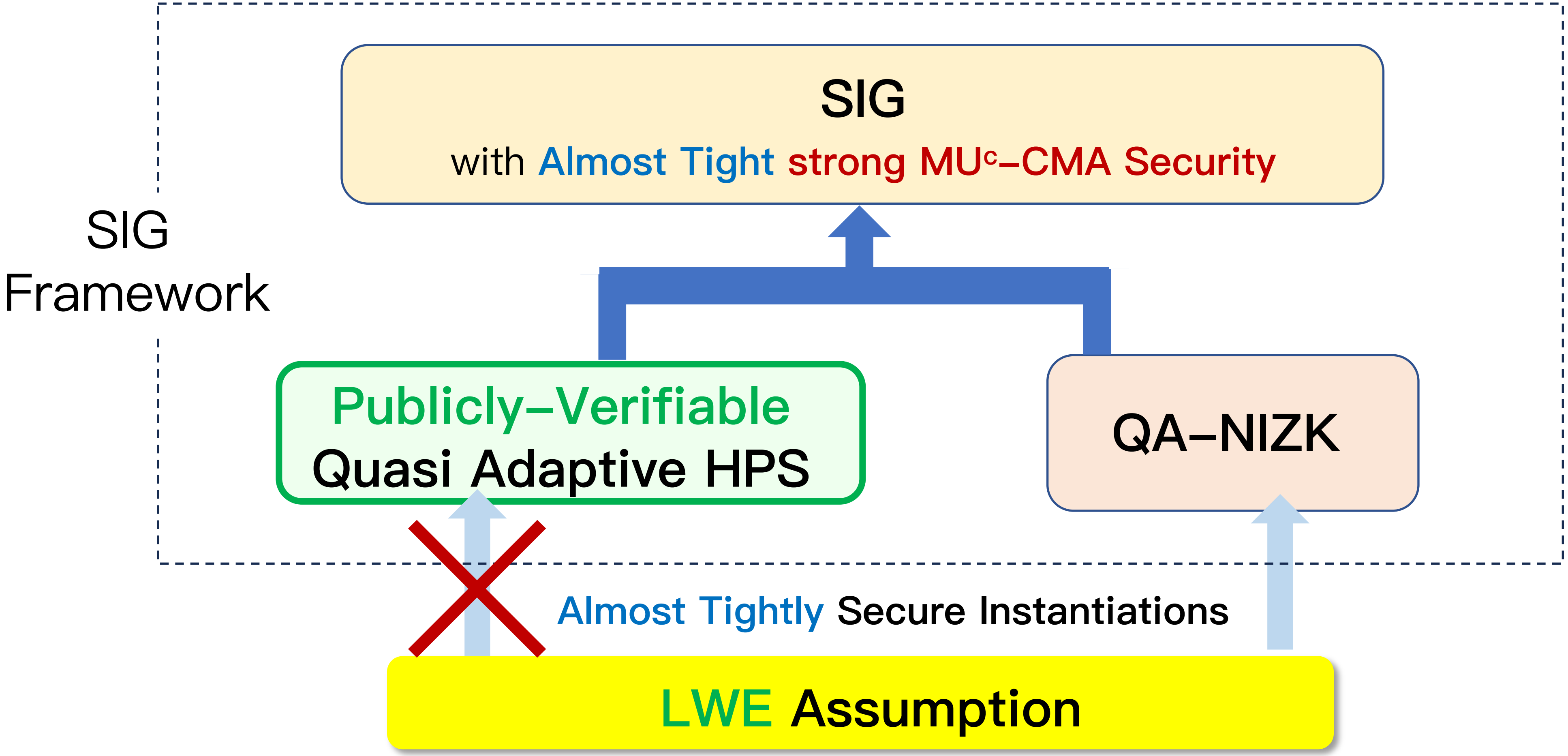
with Almost Tight strong MU^c–CMA Security

Publicly–Verifiable Quasi Adaptive HPS

QA–NIZK

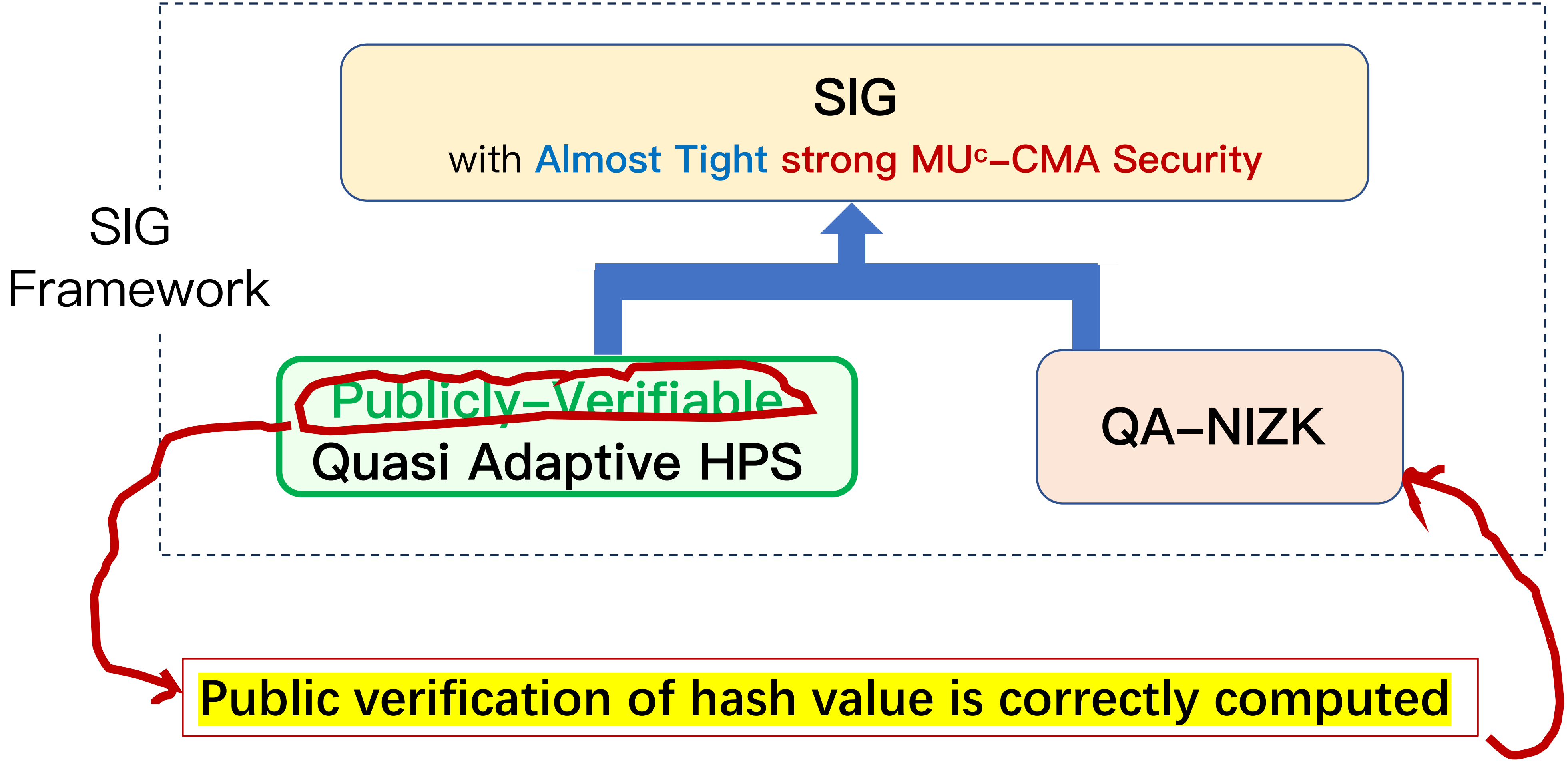
Can we achieve (almost) tight MU^c security based on LWE in the standard model?

Obstacle: No LWE-based HPS with Public Verification


SIG
with Almost Tight strong MU^c-CMA Security
SIG
Framework
Publicly-Verifiable
Quasi Adaptive HPS
QA-NIZK
Almost Tightly Secure Instantiations
LWE Assumption

Can we achieve (almost) tight MU^c security based on LWE in the standard model?

Our Solution: New Framework for Constructing SIG



SIG Framework

SIG

with Almost Tight strong MU^c–CMA Security

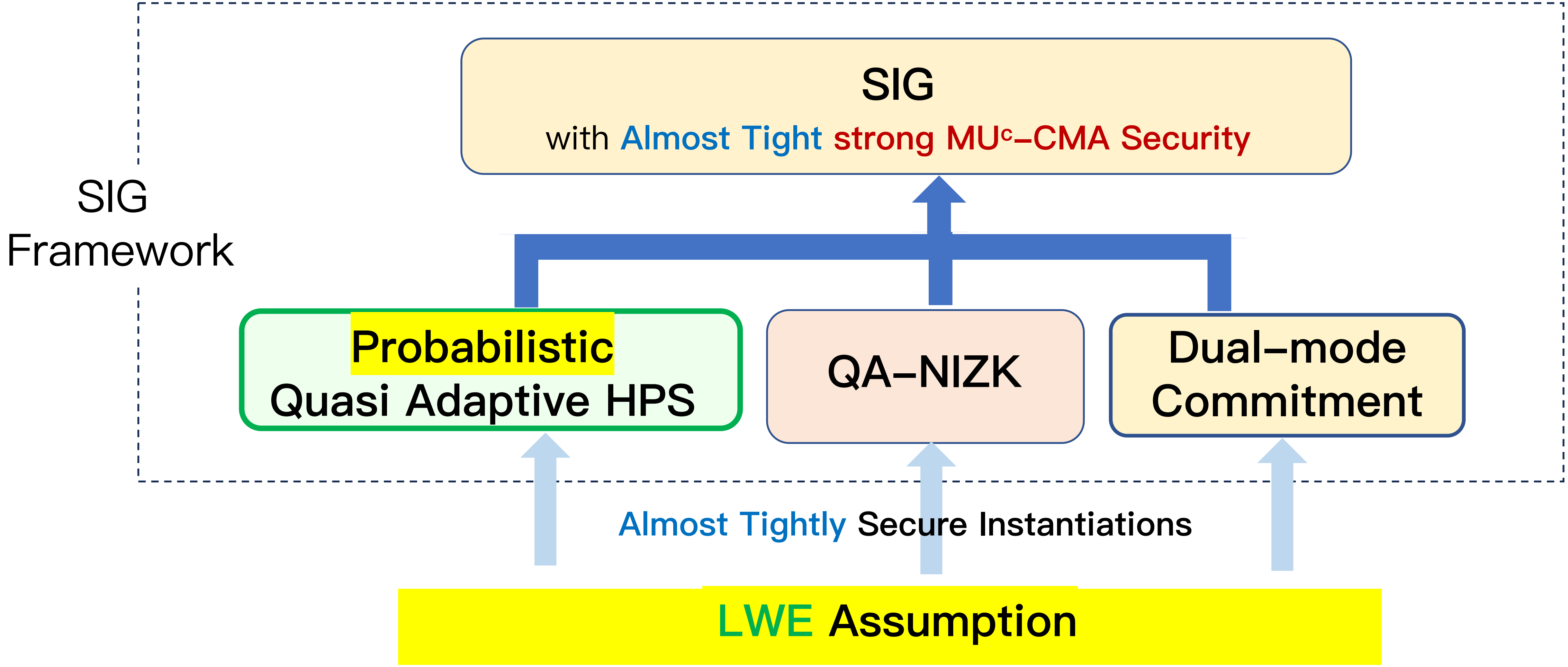
Publicly–Verifiable

Quasi Adaptive HPS

QA–NIZK

Public verification of hash value is correctly computed

Our SIG with Almost Tight MU^c–CMA security



SIG

with Almost Tight strong MU^c–CMA Security

SIG Framework

Probabilistic Quasi Adaptive HPS

QA–NIZK

Dual–mode Commitment

Almost Tightly Secure Instantiations

LWE Assumption

Our SIG with Almost Tight MU^c–CMA security

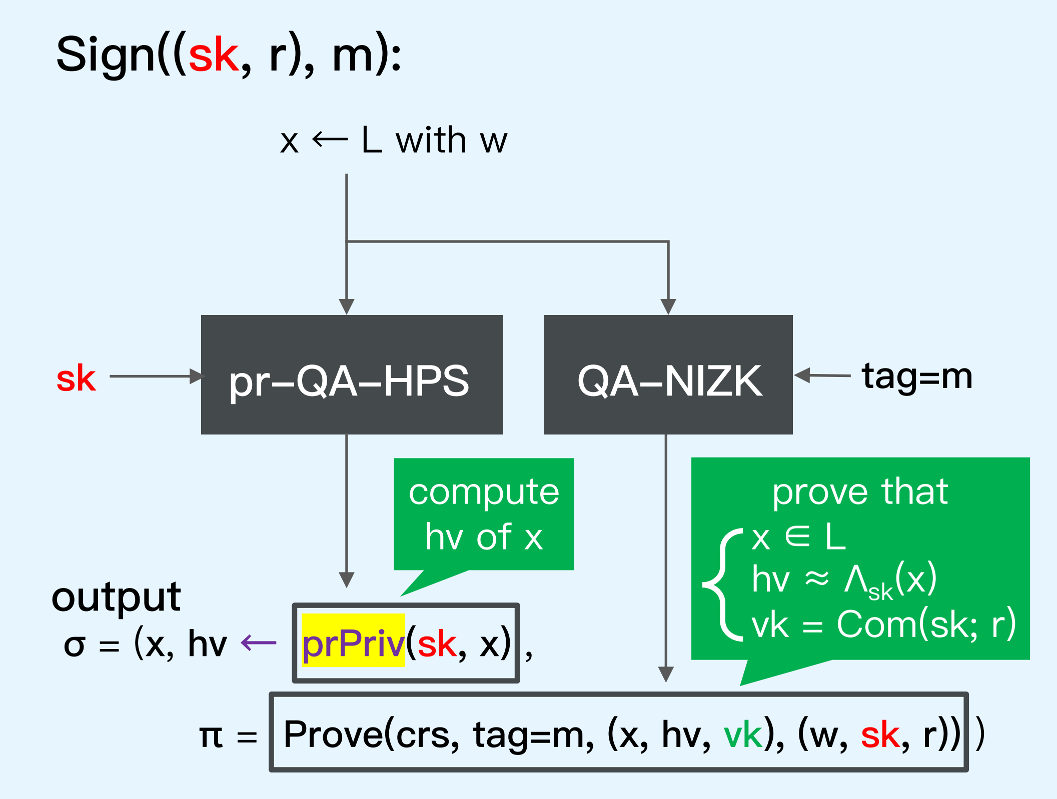
Probabilistic QA–HPS + QA–NIZK + Dual–mode Commitment

Gen → (vk = Com(sk; r), (sk, r)) : Verification key is a commitment of Hashing key

Our SIG with Almost Tight $\mathrm{MU^c}$–CMA security

Probabilistic QA–HPS + QA–NIZK + Dual–mode Commitment

Gen → (vk = Com(sk; r), (sk, r)) : Verification key is a commitment of Hashing key

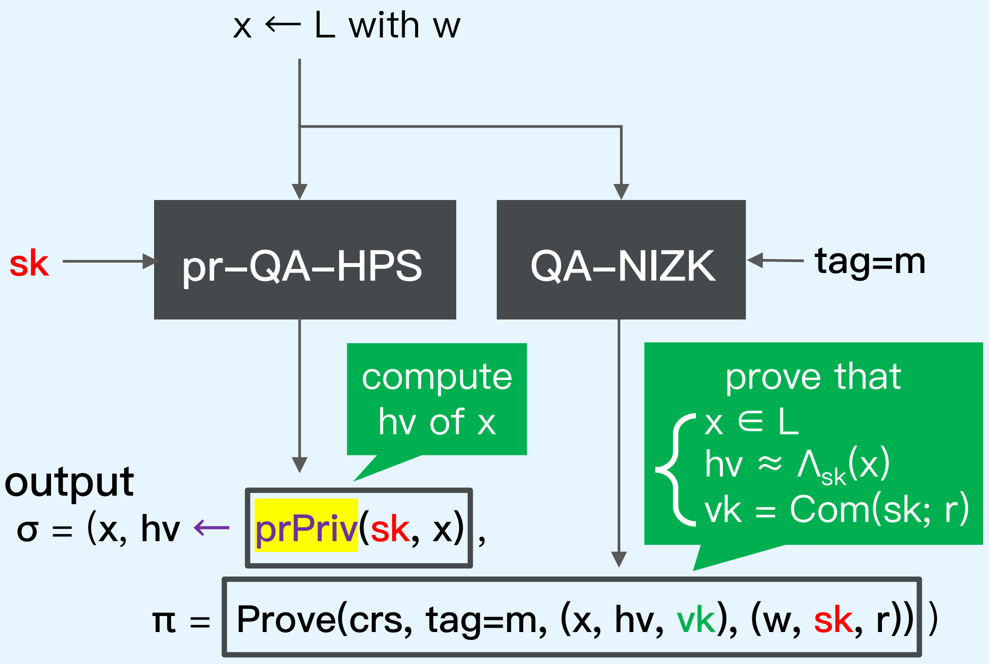


Our SIG with Almost Tight MU^{c}–CMA security

Probabilistic QA–HPS + QA–NIZK + Dual–mode Commitment

Gen → (vk = Com(sk; r), (sk, r)) : Verification key is a commitment of Hashing key

Sign((sk, r), m):



x ← L with w

sk → pr–QA–HPS

QA–NIZK ← tag=m

compute hv of x

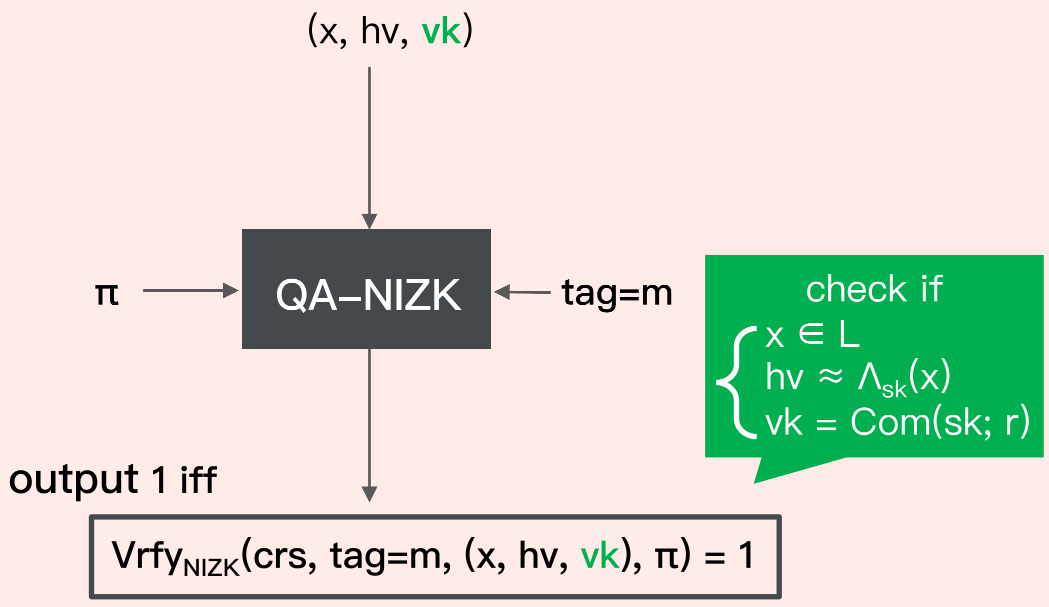
prove that
$$\begin{cases} x \in L \\ hv \approx \Lambda_{sk}(x) \\ vk = \text{Com}(sk; r) \end{cases}$$

output

σ = (x, hv ← $\boxed{\text{prPriv}(sk, x)}$,

π = $\boxed{\text{Prove}(\text{crs}, \text{tag}=m, (x, hv, vk), (w, sk, r))}$)

Vrfy(vk, m, σ = (x, hv, π)):



(x, hv, vk)

π → QA–NIZK ← tag=m

check if
$$\begin{cases} x \in L \\ hv \approx \Lambda_{sk}(x) \\ vk = \text{Com}(sk; r) \end{cases}$$

output 1 iff

$\boxed{\text{Vrfy}_{\text{NIZK}}(\text{crs}, \text{tag}=m, (x, hv, vk), \pi) = 1}$

Almost Tight (strong) MU^{c}–CMA security of SIG

Signing Oracle (m):

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ \ d := \mathsf{prPriv}(\mathbf{k}, \mathbf{c}),\ \ \pi := \mathsf{Prove}(\mathrm{tag} = m, (\mathbf{c}, vk, d), (\mathbf{k}, r, e'))\ )$$

Successful forgery (m*, σ*= (x*, d*, π*)):

$$\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, (x^*, vk, d^*), \pi^*) = 1$$

Almost Tight (strong) MU^{c}–CMA security of SIG

Signing Oracle (m):

Evaluation IND

ZK of NIZK

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ d := \mathsf{prPriv}(\mathbf{k}, \mathbf{c}),\ \pi := \mathsf{Prove}(\mathsf{tag} = m, (\mathbf{c}, vk, d), (\mathbf{k}, r, e'))\ )$$

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ d := \boxed{\mathsf{prPub}(\alpha_{\mathbf{A}}(\mathbf{k}), \mathbf{c}, \mathbf{s})},\ \pi := \boxed{\mathsf{Sim}(\mathsf{tag} = m, (\mathbf{c}, vk, d))\ )}$$

Successful forgery (m*, σ*= (x*, d*, π*)):

$$\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, (x^*, vk, d^*), \pi^*) = 1$$

Almost Tight (strong) MU^c–CMA security of SIG

Signing Oracle (m):

Subset Membership Problem

$$\sigma := (\ \mathbf{c} \leftarrow_\$ \mathcal{L}_{\mathbf{A}},\ d := \mathsf{prPriv}(\mathbf{k},\mathbf{c}),\ \pi := \mathsf{Prove}(\mathsf{tag}=m,(\mathbf{c},vk,d),(\mathbf{k},r,e'))\ )$$

$$\sigma := (\ \mathbf{c} \leftarrow_\$ \mathcal{L}_{\mathbf{A}},\ d := \boxed{\mathsf{prPub}(\alpha_{\mathbf{A}}(\mathbf{k}),\mathbf{c},\mathbf{s})},\ \pi := \mathsf{Sim}(\mathsf{tag}=m,(\mathbf{c},vk,d))\ )$$

$$\sigma := (\ \boxed{\mathbf{c} \leftarrow_\$ \mathcal{L}_{\mathbf{A}_0}},\ d := \mathsf{prPub}(\alpha_{\mathbf{A}_0}(\mathbf{k}),\mathbf{c},\mathbf{s}),\ \pi := \mathsf{Sim}(\mathsf{tag}=m,(\mathbf{c},vk,d),(\mathbf{k},r,e'))\ )$$

Successful forgery (m*, σ*= (c*, d*, π*)):

$$\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs},\tau,(\mathbf{c}^*,vk,d^*),\pi^*)=1$$

Almost Tight (strong) MU^c–CMA security of SIG

Signing Oracle (m):

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ d := \mathsf{prPriv}(\mathbf{k}, \mathbf{c}),\ \pi := \mathsf{Prove}(\mathrm{tag} = m, (\mathbf{c}, vk, d), (\mathbf{k}, r, e'))\ )$$

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ d := \boxed{\mathsf{prPub}(\alpha_{\mathbf{A}}(\mathbf{k}), \mathbf{c}, \mathbf{s})},\ \pi := \mathsf{Sim}(\mathrm{tag} = m, (\mathbf{c}, vk, d))\ )$$

$$\sigma := (\ \boxed{\mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}_0}},\ d := \mathsf{prPub}(\alpha_{\mathbf{A}_0}(\mathbf{k}), \mathbf{c}, \mathbf{s}),\ \pi := \mathsf{Sim}(\mathrm{tag} = m, (\mathbf{c}, vk, d))\ )$$

Successful forgery (m*, σ*= (x*, d*, π*)):

USS-Soundness of QA-NIZK

$$\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, (\mathbf{c}^*, vk, d^*), \pi^*) = 1$$

$$\boxed{\wedge\ \mathbf{c}^* \in \mathcal{L}_{\mathbf{A}}\ \wedge\ d^* \approx \mathsf{prPriv}(\mathbf{k}, x^*) \approx \mathsf{prPub}(\alpha_{\mathbf{A}}(\mathbf{k}), x^*, w)}$$

Almost Tight (strong) MU^c–CMA security of SIG

Signing Oracle (m):

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ d := \mathsf{prPriv}(\mathbf{k}, \mathbf{c}),\ \pi := \mathsf{Prove}(\mathsf{tag} = m, (\mathbf{c}, vk, d), (\mathbf{k}, r, e'))\ )$$

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ d := \boxed{\mathsf{prPub}(\alpha_{\mathbf{A}}(\mathbf{k}), \mathbf{c}, \mathbf{s})},\ \pi := \mathsf{Sim}(\mathsf{tag} = m, (\mathbf{c}, vk, d))\ )$$

$$\sigma := (\ \boxed{\mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}_0}},\ d := \mathsf{prPub}(\alpha_{\mathbf{A}_0}(\mathbf{k}), \mathbf{c}, \mathbf{s}),\ \pi := \mathsf{Sim}(\mathsf{tag} = m, (\mathbf{c}, vk, d))\ )$$

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}_0},\ d := \mathsf{prPub}(\boxed{\alpha_{\mathbf{A}_0}(\mathbf{k}')}, \mathbf{c}, \mathbf{s}),\ \pi := \mathsf{Sim}(\mathsf{tag} = m, (\mathbf{c}, vk, d))\ )$$

Key Switching of pr-QA-HPS

Successful forgery (m*, σ*= (c*, d*, π*)):

$$\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, (\mathbf{c}^*, vk, d^*), \pi^*) = 1$$

$$\boxed{\wedge\ \mathbf{c}^* \in \mathcal{L}_{\mathbf{A}}\ \wedge\ d^* \approx \mathsf{prPriv}(\mathbf{k}, x^*) \approx \mathsf{prPub}(\alpha_{\mathbf{A}}(\mathbf{k}), x^*, w)}$$

Almost Tight (strong) MU^{c}–CMA security of SIG

Signing Oracle (m):

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ d := \mathsf{prPriv}(\mathbf{k}, \mathbf{c}),\ \pi := \mathsf{Prove}(\mathsf{tag} = m, (\mathbf{c}, vk, d), (\mathbf{k}, r, e'))\ )$$

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ d := \boxed{\mathsf{prPub}(\alpha_{\mathbf{A}}(\mathbf{k}), \mathbf{c}, \mathbf{s})},\ \pi := \mathsf{Sim}(\mathsf{tag} = m, (\mathbf{c}, vk, d))\ )$$

$$\sigma := (\ \boxed{\mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}_0}},\ d := \mathsf{prPub}(\alpha_{\mathbf{A}_0}(\mathbf{k}), \mathbf{c}, \mathbf{s}),\ \pi := \mathsf{Sim}(\mathsf{tag} = m, (\mathbf{c}, vk, d))\ )$$

$$\sigma := (\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}_0},\ d := \mathsf{prPub}(\boxed{\alpha_{\mathbf{A}_0}(\mathbf{k}')}, \mathbf{c}, \mathbf{s}),\ \pi := \mathsf{Sim}(\mathsf{tag} = m, (\mathbf{c}, vk, d))\ )$$

Successful forgery (m*, σ*= (c*, d*, π*)):

$$\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, (\mathbf{c}^*, vk, d^*), \pi^*) = 1$$

$$\boxed{\wedge\ \mathbf{c}^* \in \mathcal{L}_{\mathbf{A}}\ \wedge\ d^* \approx \mathsf{prPriv}(\mathbf{k}, x^*) \approx \mathsf{prPub}(\alpha_{\mathbf{A}}(\mathbf{k}), x^*, w)}$$

Hardly true due to entropy of k

Subtlety 1: QA–NIZK with Tight Security from LWE

- In our SIG and PKE constructions, we need **QA–NIZKs** proving that

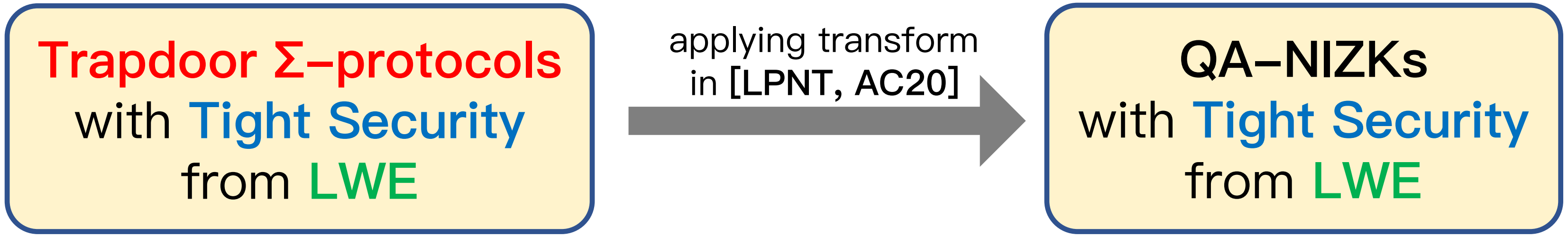
Linear Equation for PKE

$$\mathbf{c} \in \mathcal{L}_{\mathbf{A}} \quad \Leftrightarrow \quad \mathbf{c} = \mathbf{A}^{\top}\mathbf{s} + \mathbf{e}$$

Linear Equations for SIG

$$\left\{ \begin{array}{l} \mathsf{hv} \approx \Lambda_{\mathbf{k}}(\mathbf{c}) \quad \Leftrightarrow \quad \mathsf{hv} = \mathbf{c}^{\top}\mathbf{k} + \mathrm{small} \\[2em] vk = \mathsf{Com}(\mathbf{k}; \mathbf{R}) \quad \Leftrightarrow \quad \mathbf{X} \cdot \mathbf{R} + \begin{pmatrix} \mathbf{0} \\ q \cdot \mathbf{k}^{\top} \end{pmatrix} \quad \text{(Regev Encryption)} \end{array} \right.$$

- We build **QA–NIZKs** for such languages



Trapdoor Σ–protocols with Tight Security from LWE

applying transform in **[LPNT, AC20]**

QA–NIZKs with Tight Security from LWE

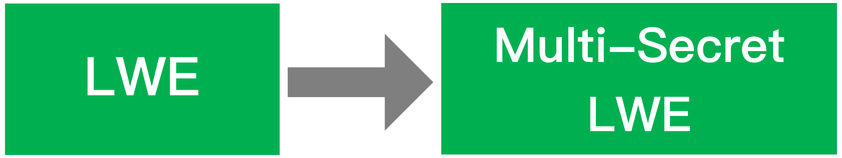
Subtlety 2: Almost Tight Reduction from LWE to Multi-Secret LWE

- In the MU^c security proof, we require the hardness of **Multi-fold Subset Membership Problem** (**SMP**) of **Probabilistic QA-HPS**

SMP:
$$(\mathbf{A}, \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top) \approx_c (\mathbf{A}, \$)$$
$$\mathbf{s} \leftarrow_\$ \mathbb{Z}_q^{\times n}, \mathbf{e} \leftarrow_\$ \chi^m$$

Multi-fold SMP:
$$(\mathbf{A}, \mathbf{S}\mathbf{A} + \mathbf{E}) \approx_c (\mathbf{A}, \$)$$
$$\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{n \times m}, \mathbf{S} \leftarrow_\$ \mathbb{Z}_q^{Q \times n}, \mathbf{E} \leftarrow_\$ \chi^{Q \times m}$$

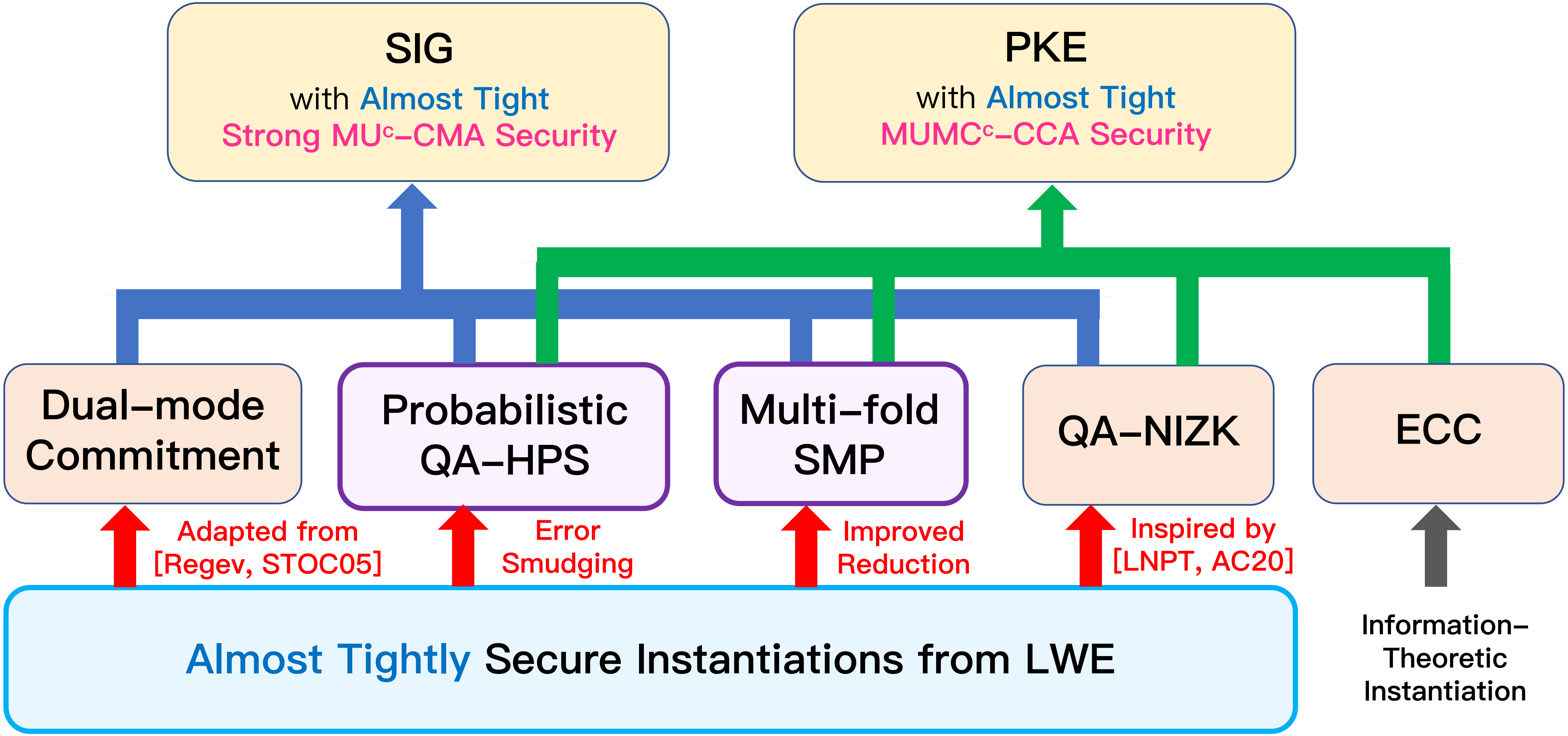
- Improved **Almost Tight** Reduction


LWE

Multi-Secret LWE

  - The reduction implicit in **[Alwen-Krenn-Pietrzak-Wichs, C13]** has $\ell = \lambda^3$
  - ✓ **Our fine-grained reduction has** $\boldsymbol{\ell = \lambda^2}$ by applying the noise lossiness approach in [Brakerski-Döttling, EC20]

Summary of Our SIG and PKE



SIG
with Almost Tight
Strong MU^c–CMA Security

PKE
with Almost Tight
MUMC^c–CCA Security

Dual–mode Commitment

Probabilistic QA–HPS

Multi–fold SMP

QA–NIZK

ECC

Adapted from [Regev, STOC05]

Error Smudging

Improved Reduction

Inspired by [LNPT, AC20]

Almost Tightly Secure Instantiations from LWE

Information–Theoretic Instantiation

Conclusion

- The first SIG and PKE schemes
  - ✓ with almost tight MU^c security from LWE in the standard model.
- Generic constructions of SIG and PKE by using
  - New technical tool: Probabilistic QA–HPS.
- Improved almost tight reductions from LWE to Multi–Secret LWE.

https://eprint.iacr.org/2023/1230

Thanks!   Questions?