Adamas: One-Round Lattice-Based Threshold KEM and PKE under Adaptive Corruptions

Shengli Liu, Shuai Han, Yuxin Zhang, Yunxiao Zhou

Shanghai Jiao Tong University

October 2026 @ NIST TCPT3

Threshold Call Preview Talks #3

Presented on 2026-10-13

Our Main Results

TWO Threshold KEM

$TKEM_1$

$TKEM_2$

ONE Threshold PKE

TPKE

All our schemes enjoy

01 **One-Round Decryption (Non-Interactivity)**

One response per decryptor. No interaction among decryptors.

02 **Adaptive Security: IND-CCA Security & Active Security under Adaptive Corruptions**

Secure against chosen-ciphertext & active attacks under adaptive corruptions.

03 **Polynomial Modulus**

Provide more post-quantum confidence than super-poly modulus.

Outline

Adamas

Threshold KEM/PKE

01 **Introduction**
Syntax, CCA & active security under adaptive corruptions

02 **Our Proposal**
$TKEM_1$, $TKEM_2$, TPKE and proof ideas

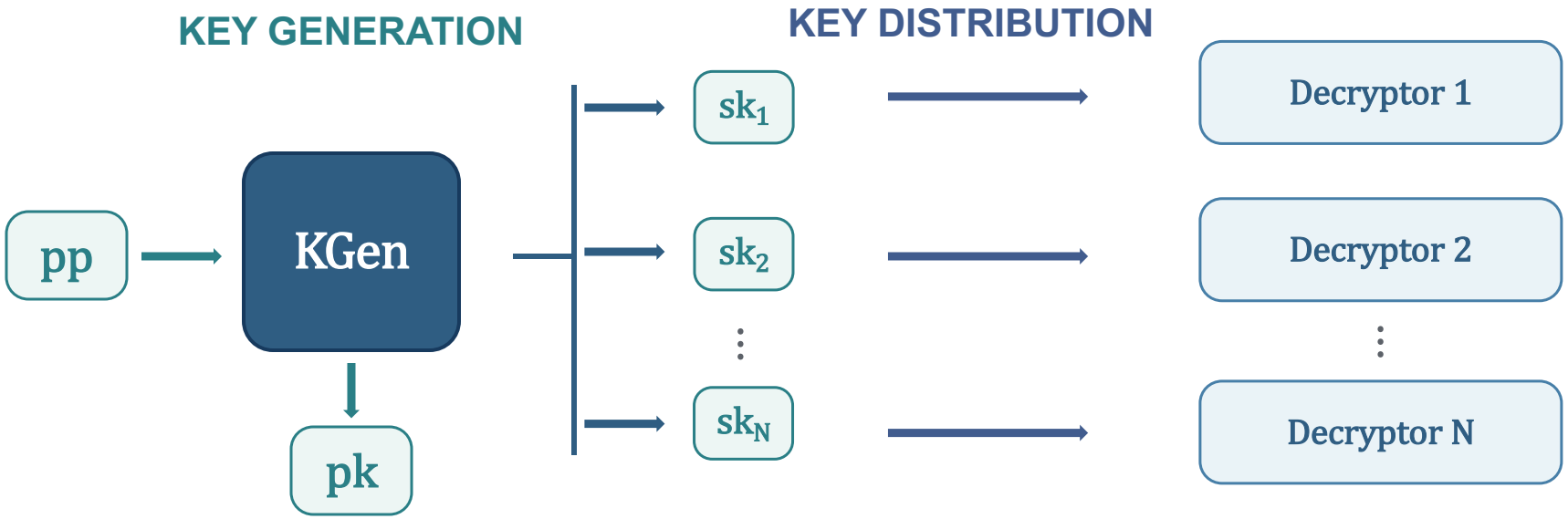
03 **Performance of Our Proposal**
Parameters and communication costs, comparison with other related works

04 **Summary**
Contributions and future work

Non-interactive (t,N)-Threshold KEM

TKEM ALGORITHM

$$\mathrm{TKEM} = (\mathrm{Setup}, \mathrm{KGen}, \mathrm{Enc}, \mathrm{PDec}, \mathrm{Comb})$$



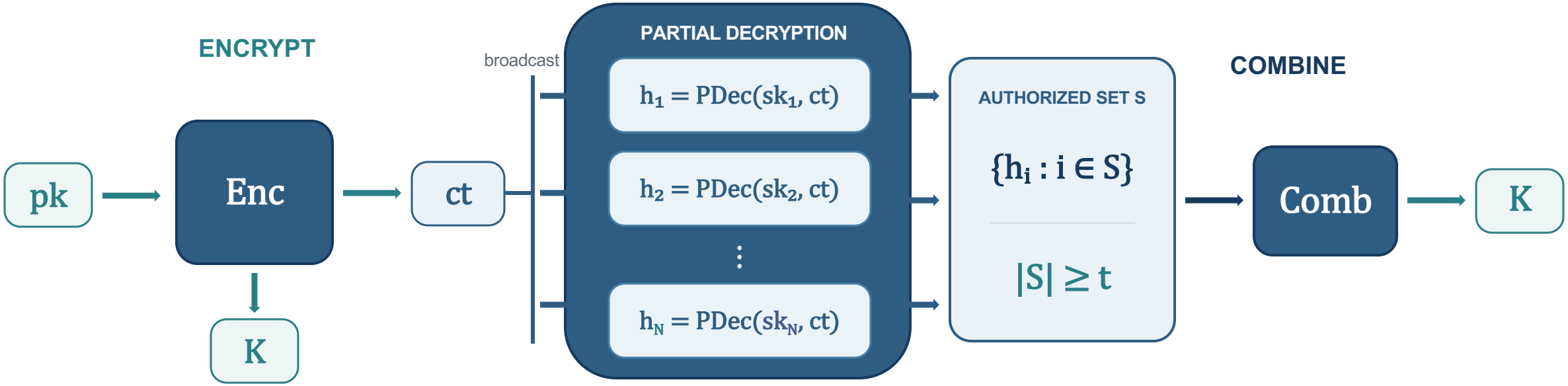
TRUSTED DEALER

We assume a trusted dealer for key generation and secret-share distribution.

Non-interactive (t,N)-Threshold KEM

TKEM ALGORITHM

$$\mathrm{TKEM} = (\mathrm{Setup}, \mathrm{KGen}, \mathrm{Enc}, \mathrm{PDec}, \mathrm{Comb})$$



ONE-ROUND DECRYPTION

Each decryptor computes its partial decryption independently.
No interaction among decryptors is required.

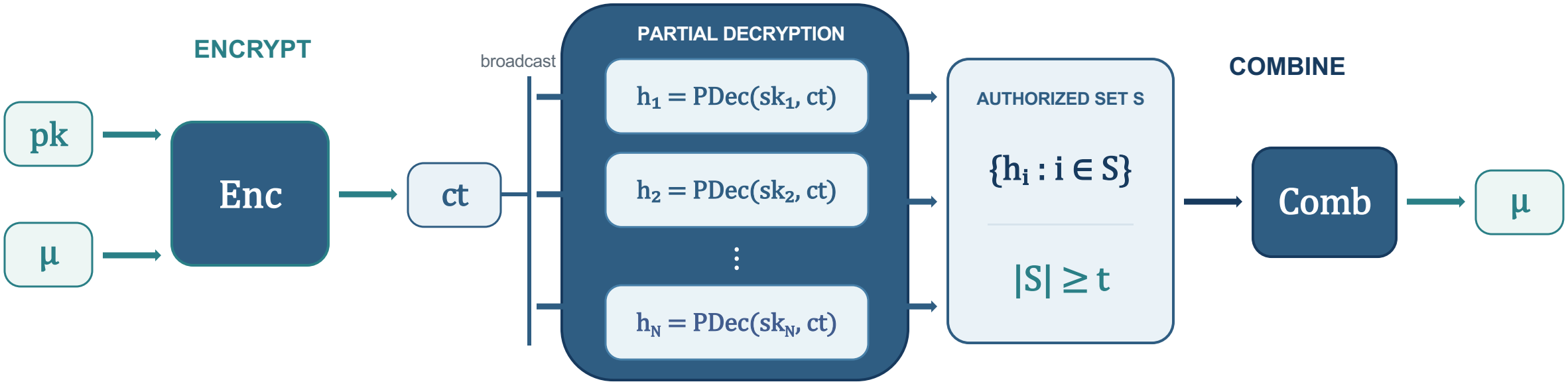
Non-interactive (t,N)-Threshold PKE

TPKE ALGORITHM

$\mathrm{TPKE} = (\mathrm{Setup}, \mathrm{KGen}, \mathrm{Enc}, \mathrm{PDec}, \mathrm{Comb})$

KEY GENERATION

KGen outputs pk and secret shares $sk_1, \ldots, sk_n$

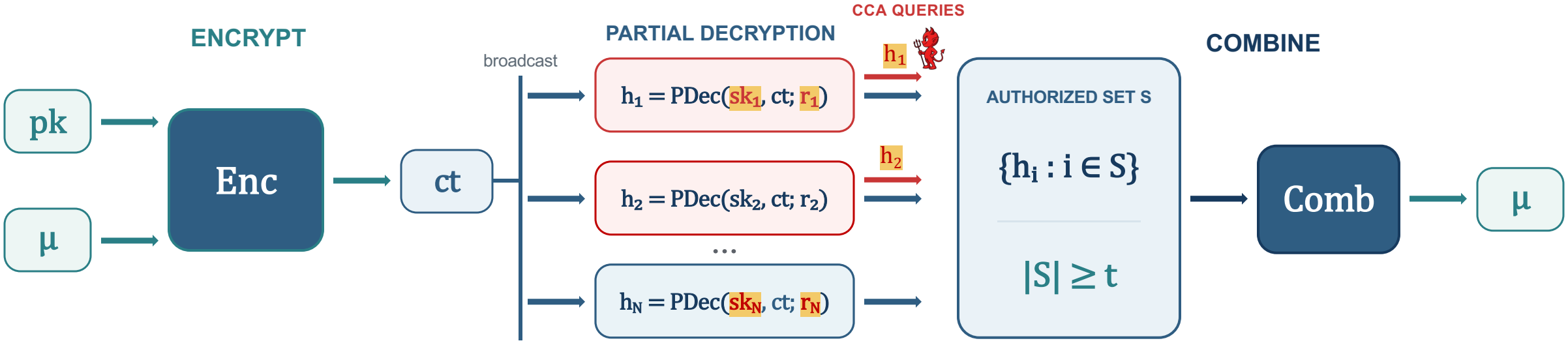


ONE-ROUND DECRYPTION

Each decryptor computes its partial decryption independently.
No interaction among decryptors is required.

CCA Security under Adaptive Corruptions

ADAPTIVE CORRUPTIONS (no secure erasures) Adaptively corrupt decryptor i to obtain its secret key sk_i and the randomness r_i used in previous partial decryptions.



CHALLENGE OF ADAPTIVE CCA SECURITY The adversary may obtain h_i first and corrupt decryptor i later. Without secure erasures, the revealed (sk_i, r_i) must explain the earlier h_i.

Existing Lattice-based Threshold PKE/KEM with CCA Security under Adaptive Corruptions

[DLN+21]
The only prior lattice-based Threshold PKE achieving both properties

One-Round Decryption

CCA Security under Adaptive Corruptions

But it requires super-polynomial modulus

Efficiency

Large-q modular computation and big-int arithmetic libraries

Security

Reduction to GapSVP problem with super-poly approximation factor

OPEN PROBLEM

"It remains an open problem to prove adaptive security under a more common LWE assumption with polynomial approximation factor."

[DLN+21] J. Devevey, B. Libert, K. Nguyen, T. Peters, and M. Yung. Non-interactive CCA2-secure threshold cryptosystems. PKC 2021.

All Our Schemes Resolve the Open Problem

$TKEM_1$

- **Synchronous** setting
- Support **large-scale** parties
- **Security-with-abort**

$TKEM_2$, TPKE

01 One-Round Decryption
Round-optimal decryption protocol

02 Adaptive Security
CCA and active security under adaptive corruptions

03 Polynomial Modulus
Polynomial approximation factor & compact ciphertexts

- **Asynchronous** setting
- Support **small-scale** parties
- TPKE achieve **Robustness**

All our schemes achieve CCA security under adaptive corruptions from MLWE with polynomial modulus, thereby resolving the open problem above.

Outline

Adamas

Threshold KEM/PKE

01 **Introduction**

Syntax, CCA & active security under adaptive corruptions

02 **Our Proposal**

$TKEM_1$, $TKEM_2$, TPKE and proof ideas

03 **Performance of Our Proposal**

Parameters and communication costs, comparison with other related works

04 **Summary**

Contributions and future work

Adamas: One-Round Threshold KEM/PKE under Adaptive Corruptions

Large-Scale Synchronous Setting

Components

- [LP11] Optimized Regev PKE
- Shamir Secret Sharing
- Simulation-extractable NIZK

+

Techniques

- Polynomial Flooding
- Zero-sum Mask

$\overline{\text{TPKE}_1}$: **CCA security under adaptive corruptions in sync. setting**

+ FO Transform →

TKEM_1

Synchronous TKEM with CCA security & security-with-abort under adaptive corruptions

Small-Scale Asynchronous Setting

Components

- Dual Regev PKE
- Simulation-sound NIZK
- Replicated Secret Sharing

+

Techniques

- Polynomial Flooding
- Adaptive Leftover Hash Lemma (LHL)

$\overline{\text{TPKE}_2}$: **CCA security under adaptive corruptions in async. setting**

+ FO Transform →

TKEM_2

Asynchronous TKEM with CCA Security & security-with-abort under adaptive corruptions

+ NIZK →

TPKE

Asynchronous TPKE with CCA security & robustness under adaptive corruptions

[LP11] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. CT-RSA 2011.

$\overline{\mathbf{TPKE_1}}$ = [LP11] Optimized Regev PKE + Shamir SS + NIZK

KGen(pp) → (pk, sk_1,…, sk_N)

$$\mathrm{pk} = \mathbf{A}\boldsymbol{k} + \boldsymbol{f}, \qquad \mathrm{sk} = \boldsymbol{k}$$

$$(\mathrm{sk}_1, \ldots, \mathrm{sk}_N) \leftarrow \mathrm{Shamir.Share}(\boldsymbol{k})$$

Enc(pk, μ) → ct=(c,d,π)

$$\mathbf{c}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$$

$$\mathrm{d} = \mathbf{s}^\top \mathrm{pk} + \mathrm{e}' + \lfloor q/p \rfloor \mu$$

$$\pi \leftarrow \mathrm{Prv}(\mathrm{crs}, (\mathrm{pk}, \mathbf{c}, \mathrm{d}) \in \mathcal{L}^{(\mathrm{ct})}, (\mathbf{s}, \mathbf{e}, \mathrm{e}', \mu))$$

PDec(sk_i, ct=(c,d,π), S) → $h_{i,S}$ /⊥

$$\text{If } \mathrm{Vfy}(\mathrm{crs}, (\mathrm{pk}, \mathbf{c}, \mathrm{d}), \pi) = 0 : \text{return } \perp$$

$$h_{i,S} = \lambda_{i,S} \cdot \mathbf{c}^\top \mathrm{sk}_i + y_{i,S} + \Delta_{i,S}$$

polynomial flooding $y_{i,S} \leftarrow D_{\mathcal{R},\sigma_y}$

mask $\sum_{i \in S} \Delta_{i,S} = 0$

Comb(ct, S, {$h_{i,S}$}$_{i∈S}$) → μ/ ⊥

$$\text{If } |S| < t: \text{ return } \perp;$$

$$h = \sum_{i \in S} h_{i,S} = \mathbf{c}^\top \boldsymbol{k} + \sum_{i \in S} y_{i,S}$$

$$= \mathbf{s}^\top \mathrm{pk} + \mathbf{e}^\top \boldsymbol{k} - \mathbf{s}^\top \boldsymbol{f} + \sum_{i \in S} y_{i,S}$$

$$\mu = \lfloor (\mathrm{d} - h) \cdot p/q \rceil, \qquad \text{return } \mu$$

simulation-extractable NIZK

$$\mathcal{L}^{(\mathrm{ct})} = \left\{ (\mathrm{pk}, \mathbf{c}, \mathrm{d}) \middle| \exists\, (\mathbf{s}, \mathbf{e}, \mathrm{e}', \mu), \text{s.t.} \begin{array}{c} \| (\mathbf{s}, \mathbf{e}, \mathrm{e}') \| \le B_E \wedge \mathbf{c}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top \\ \mathrm{d} = \mathbf{s}^\top \mathrm{pk} + \mathrm{e}' + \lfloor q/p \rfloor \mu \end{array} \right\}$$

[LP11] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. CT-RSA 2011.

$\overline{\text{TPKE}_1}$: Sketch of CCA Security Proof under Adaptive Corruptions

1. **Uniform masks $\Delta_{i,S}$** — All but the last $h_{i,S} = \$$ → **Adaptive corruptions**: Assign unused random secret key share when parties are corrupted.
2. **Noisy linear leakage** — Last $h_{i,S} = \cdots + \mathbf{e}^\top \boldsymbol{k} - \mathbf{s}^\top \boldsymbol{f} + y$
3. **Adaptive Hint-MLWE** — $\text{pk} = \mathbf{A}\boldsymbol{k} + \boldsymbol{f} \longrightarrow \text{pk} = \$$ → **Adaptive partial dec. queries**: The coefficient (s,e) is adaptively-chosen by adversary.
4. **MLWE** — $\begin{cases} \mathbf{c}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top \\ \text{d} = \mathbf{s}^\top \text{pk} + \text{e}' + \lfloor \text{q}/\text{p} \rceil \mu \end{cases} \longrightarrow \begin{cases} \mathbf{c}^\top = \$ \\ \text{d} = \$ \end{cases}$ → **Hiding message μ**
5. **NIZK** — If $\text{Vfy}(\text{crs}, (\text{pk}, \mathbf{c}, \text{d}), \pi) = 0 : \text{return } \perp$ → **Achieving CCA security**: Reject partial decryption queries with $(\text{pk}, \mathbf{c}, \text{d}) \notin \mathcal{L}^{(\text{ct})}$

$\overline{\text{TPKE}_1}$ achieves CCA security under adaptive corruptions and polynomial modulus, but does not guarantee active security yet.

$\overline{\mathbf{TPKE_2}}$ = Dual-Regev PKE + Replicated SS + NIZK

KGen(pp) → (pk, sk_1,…, sk_N)

$$\mathrm{pk} = \mathbf{A}\boldsymbol{k},$$

Replicated secret sharing (RSS)

Enc(pk, μ) → ct=(c,d,π)

$$\mathbf{c}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$$

$$\mathrm{d} = \mathbf{s}^\top \mathrm{pk} + e' + \lfloor \mathrm{q}/\mathrm{p} \rceil \mu$$

PDec(sk_i, ct=(c,d,π)) → h_i / ⊥

Comb(pk, ct, S, $\{h_i\}_{i\in S}$) → μ / ⊥

$\overline{\mathbf{TPKE_2}}$ = Dual-Regev PKE + Replicated SS + NIZK

KGen(pp) → (pk, sk_1,…, sk_N)

$$\mathrm{pk} = \mathbf{A}\boldsymbol{k},$$

Replicated secret sharing (RSS)

Enc(pk, μ) → ct=(c,d,π)

$$\mathbf{c}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top$$

$$\mathrm{d} = \mathbf{s}^\top\mathrm{pk} + e' + \lfloor q/p \rceil \mu$$

$$\tau = \mathrm{H}(\mathbf{c}, \mathrm{d}), \qquad \pi \leftarrow \mathrm{Prv}\big(\mathrm{crs}, \mathbf{c}, \tau, (\mathbf{s}, \mathbf{e})\big)$$

One-Time Simulation Sound tag-based QA-NIZK

PDec(sk_i, ct=(c,d,π)) → h_i / ⊥

$$\tau = \mathrm{H}(\mathbf{c}, \mathrm{d}) \text{ If } \mathrm{Vfy}(\mathrm{crs}, \mathbf{c}, \tau, \pi) = 0\text{: return } \perp$$

Comb(pk, ct, S, $\{h_i\}_{i\in S}$) → μ / ⊥

$\overline{\mathbf{TPKE}_2}$ = Dual-Regev PKE + Replicated SS + NIZK

KGen(pp) → (pk, sk_1,…, sk_N)

$$\left(\boldsymbol{k}, \{\mathrm{sk}_i\}_{\mathrm{i}\in[\mathrm{N}]}\right) \leftarrow \mathrm{RSS.Share}\left(\mathrm{N}, \mathrm{t}, D_{\mathcal{R}^{\mathrm{m}},\sigma_k}\right)$$

$$\mathcal{I} = \{I \subseteq [\mathrm{N}] : |I| = \mathrm{N} - \mathrm{t} + 1\}$$

$$\mathrm{pk} = \mathbf{A}\boldsymbol{k}, \qquad \mathrm{sk}_i = \{\boldsymbol{k}_I : \mathrm{I} \in \mathcal{I}, \mathrm{i} \in \mathrm{I}\}$$

Replicated secret sharing (RSS)

Enc(pk, μ) → ct=(c,d,π)

$$\mathbf{c}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top$$

$$\mathrm{d} = \mathbf{s}^\top\mathrm{pk} + e' + \lfloor \mathrm{q}/\mathrm{p} \rfloor \mu$$

$$\tau = \mathrm{H}(\mathbf{c}, \mathrm{d}), \qquad \pi \leftarrow \mathrm{Prv}\left(\mathrm{crs}, \mathbf{c}, \tau, (\mathbf{s}, \mathbf{e})\right)$$

One-Time Simulation Sound tag-based QA-NIZK

PDec(sk_i, ct=(c,d,π)) → h_i / ⊥

$$\tau = \mathrm{H}(\mathbf{c}, \mathrm{d}) \text{ If } \mathrm{Vfy}(\mathrm{crs}, \mathbf{c}, \tau, \pi) = 0\text{: return } \perp$$

For each $\boldsymbol{k}_I \in \mathrm{sk}_i$:

$$h_{i,I} = \mathbf{c}^\top \boldsymbol{k}_I + y_{i,I}$$

return $h_i = \{h_{i,I}\}$

polynomial flooding
$y_{i,I} \leftarrow D_{\mathcal{R},\sigma_y}$

Comb(pk, ct, S, $\{h_i\}_{i\in S}$) → μ / ⊥

If $|\mathrm{S}| < \mathrm{t}$: return ⊥;

$$(w_i)_{\mathrm{i}\in\mathrm{S}} \leftarrow \mathrm{Select}(\mathrm{S})$$

$$h := \sum_{\mathrm{i}\in\mathrm{S}} \sum_{\mathrm{I}\in w_i} h_{i,I}$$

$$\mu := \lfloor (\mathrm{d} - h) \cdot \mathrm{p}/\mathrm{q} \rceil, \qquad \text{return } \mu$$

Correctness $\mathrm{B}_1 + \sqrt{\mathrm{C}}\left(\mathrm{B}'\mathrm{B}_\mathrm{k}' + \mathrm{B}_\mathrm{y}\right) \leq \lfloor \mathrm{q}/\mathrm{p} \rfloor / 2$

Aggregate flooding width: $\sqrt{\mathrm{C}} \cdot \sigma_y$

where $\mathrm{C} = \binom{\mathrm{N}}{\mathrm{N}-\mathrm{t}+1}$

$\overline{\mathbf{TPKE_2}}$: Sketch of Adaptively CCA Security Proof

1 **Zero-Knowledge**

Replace π^* in challenge ciphertext by Sim(crs, c^*).

$\overline{\mathbf{TPKE_2}}$: Sketch of Adaptively CCA Security Proof

1 Zero-Knowledge

Replace π^* in challenge ciphertext by Sim(crs, c^*).

2 Hint-MLWE

Challenge ciphertext $ct^* = (c^*, d^*, \pi^*) \approx (\$, d^*, \pi^*)$ due to Hint-MLWE.

$\overline{\mathbf{TPKE_2}}$: Sketch of Adaptively CCA Security Proof

1 Zero-Knowledge

Replace π^* in challenge ciphertext by Sim(crs, c^*).

2 Hint-MLWE

Challenge ciphertext $ct^* = (c^*, d^*, \pi^*) \approx (\$, d^*, \pi^*)$ due to Hint-MLWE.

3 OTSS NIZK

Partial Decryption Oracle: only accept ct=(c,d,π)

with **MLWE instance** $\mathbf{c}^\top = \boldsymbol{s}^\top \mathbf{A} + \boldsymbol{e}^\top$.

$\overline{\mathbf{TPKE_2}}$: Sketch of Adaptively CCA Security Proof

1 Zero-Knowledge

Replace π^* in challenge ciphertext by Sim(crs, c^*).

2 Hint-MLWE

Challenge ciphertext $ct^* = (c^*, d^*, \pi^*) \approx (\$, d^*, \pi^*)$ due to Hint-MLWE.

3 OTSS NIZK

Partial Decryption Oracle: only accept ct=(c,d,π)

with **MLWE instance** $\mathbf{c}^\top = \boldsymbol{s}^\top \mathbf{A} + \boldsymbol{e}^\top$.

4 PDec Simulation

Simulate partial decryptions $\{h_{i,I} = \boldsymbol{c}^\top \boldsymbol{k_I} + y_{i,I}\}$ with

$\{h_{i,I} = \boldsymbol{s}^\top \mathbf{A} \boldsymbol{k}_I + \boldsymbol{e}^\top \boldsymbol{k}_I + y_{i,I}\}$

$\overline{\mathbf{TPKE_2}}$: Sketch of Adaptively CCA Security Proof

1 Zero-Knowledge

Replace π^* in challenge ciphertext by Sim(crs, c^*).

2 Hint-MLWE

Challenge ciphertext $ct^* = (c^*, d^*, \pi^*) \approx (\$, d^*, \pi^*)$ due to Hint-MLWE.

3 OTSS NIZK

Partial Decryption Oracle: only accept ct=(c,d,π)

with **MLWE instance** $\mathbf{c}^\top = \boldsymbol{s}^\top \mathbf{A} + \boldsymbol{e}^\top$.

4 PDec Simulation

Simulate partial decryptions $\{h_{i,I} = \boldsymbol{c}^\top \boldsymbol{k_I} + y_{i,I}\}$ with

$\{h_{i,I} = \boldsymbol{s}^\top \mathbf{A} \boldsymbol{k}_I + \boldsymbol{e}^\top \boldsymbol{k}_I + y_{i,I}\}$

Uniform c* acts as extractor

Adaptively chosen noisy linear hints

5 Adaptive LHL

Change $ct^* = (\$, d^*, \pi^*)$ to $ct^* = (\$, \$, \pi^*)$ where $d^* = \boldsymbol{c}^{*\top} \boldsymbol{k}_{\tilde{I}} + \cdots$ is uniform due to Adaptive LHL.

$\overline{\mathbf{TPKE}_2}$: Sketch of Adaptively CCA Security Proof

1 Zero-Knowledge

Replace π^* in challenge ciphertext by Sim(crs, c^*).

2 Hint-MLWE

Challenge ciphertext $ct^* = (c^*, d^*, \pi^*) \approx (\$, d^*, \pi^*)$ due to Hint-MLWE.

Adaptive-CCA Security

The reduction samples the key and shares, allowing consistent answers to later corruptions.

3 OTSS NIZK

Partial Decryption Oracle: only accept ct=(c,d,π)

with **MLWE instance** $\mathbf{c}^\top = \boldsymbol{s}^\top \mathbf{A} + \boldsymbol{e}^\top$.

4 PDec Simulation

Simulate partial decryptions $\{h_{i,I} = \boldsymbol{c}^\top \boldsymbol{k_I} + y_{i,I}\}$ with

$\{h_{i,I} = \boldsymbol{s}^\top \mathbf{A} \boldsymbol{k}_I + \boldsymbol{e}^\top \boldsymbol{k}_I + y_{i,I}\}$

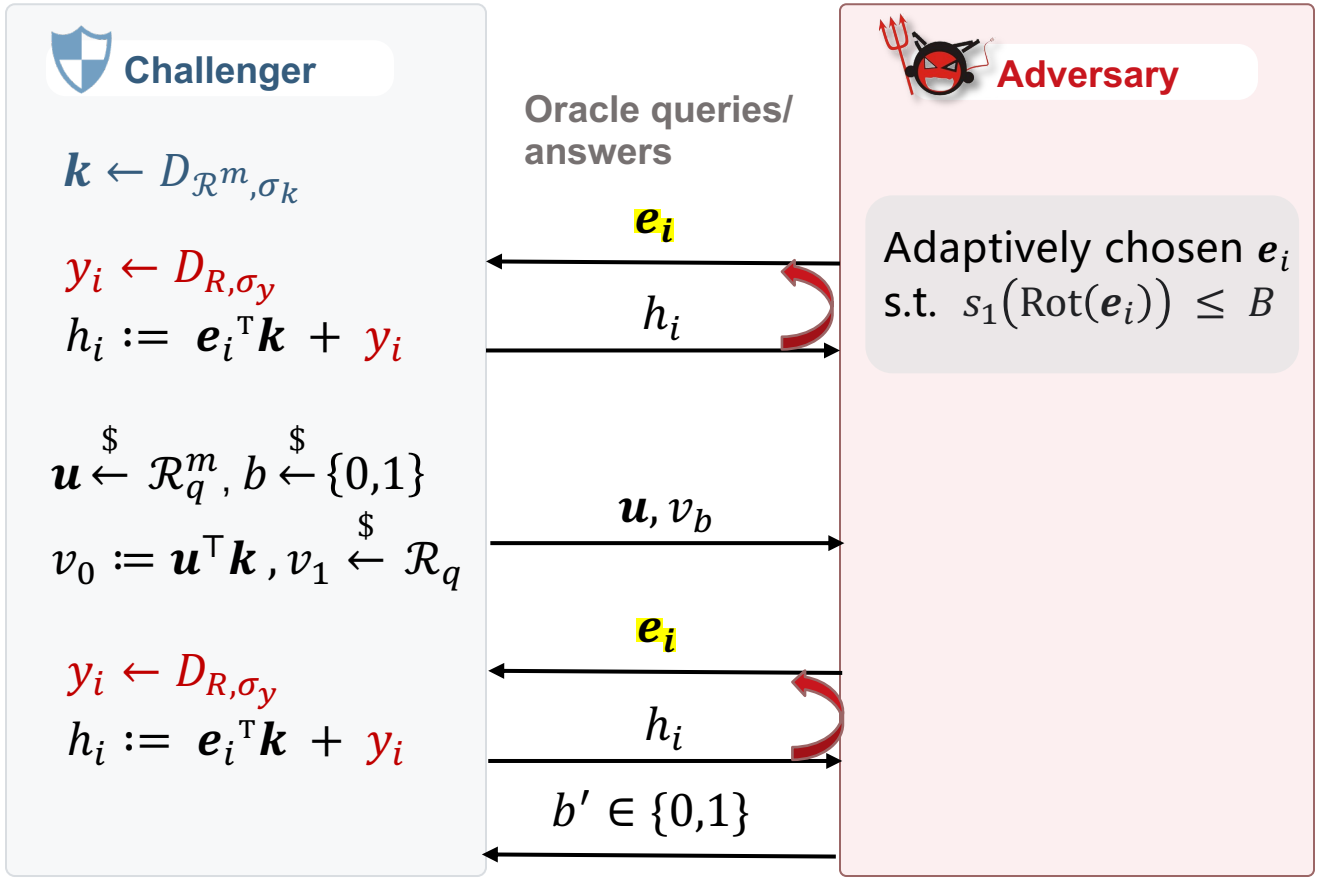
Uniform c* acts as extractor

Adaptively chosen noisy linear hints

5 Adaptive LHL

Change $ct^* = (\$, d^*, \pi^*)$ to $ct^* = (\$, \$, \pi^*)$ where $d^* = \boldsymbol{c}^{*\top} \boldsymbol{k}_{\tilde{I}} + \cdots$ is uniform due to Adaptive LHL.

New Technique: Adaptive LHL under noisy linear leakage



Theorem (brief)

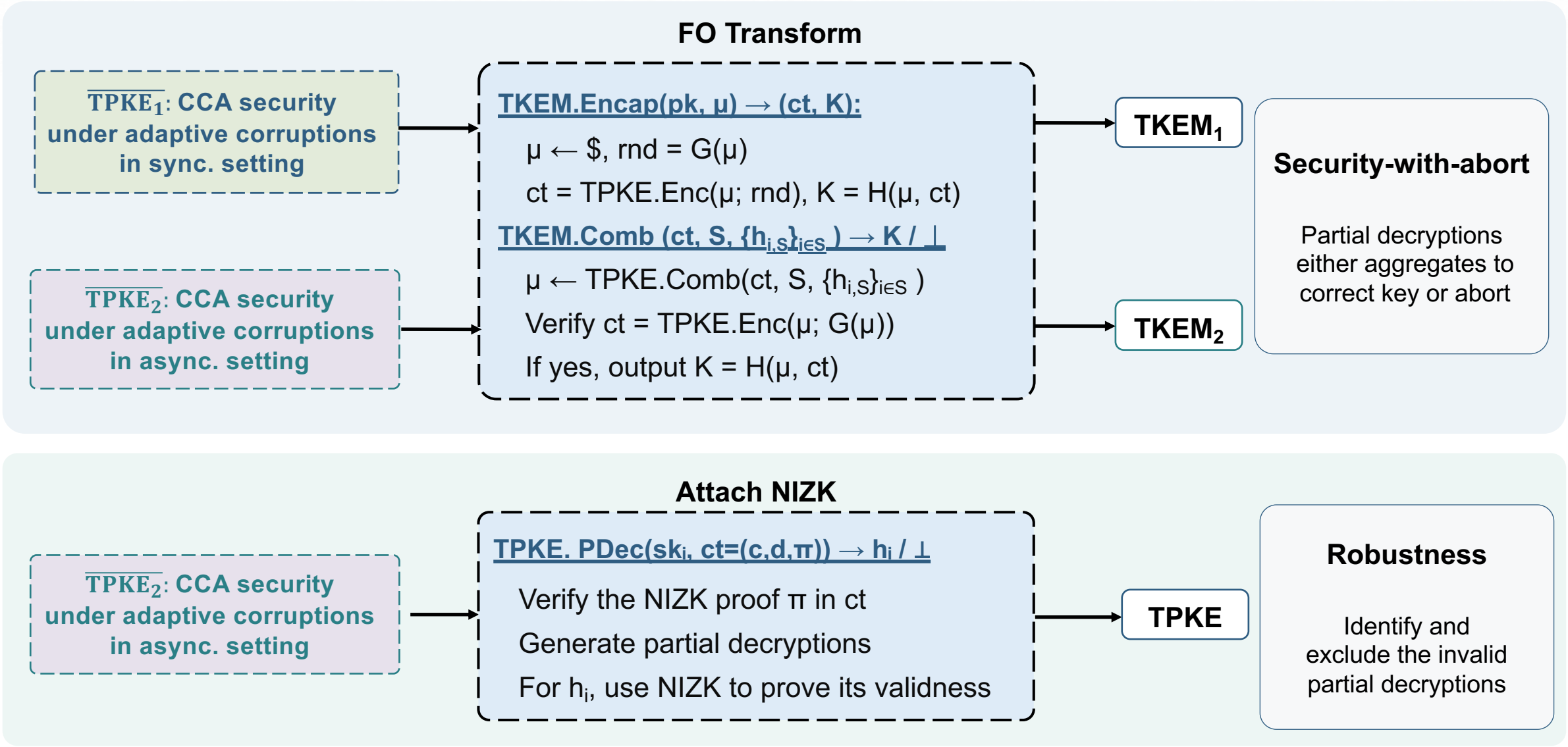
For any all-powerful $\mathcal{A}$ that makes ℓ queries, if

- $\frac{1}{\sigma_k^2} + \frac{\ell B^2}{\sigma_y^2} \leq \frac{1}{\sigma^2}$ and
- $\sigma \geq \eta_\epsilon\left(\Lambda_q^{\perp}(\boldsymbol{u})\right)$,

then it holds that

$$\left|\Pr[b = b'] - \frac{1}{2}\right| \leq \mathrm{negl}(\lambda)$$

Achieving Security-with-Abort and Robustness



Outline

Adamas

Threshold KEM/PKE

01 **Introduction**

Syntax, CCA & active security under adaptive corruptions

02 **Our Proposal**

$TKEM_1$, $TKEM_2$, TPKE and proof ideas

03 **Performance of Our Proposal**

Parameters and communication costs, comparison with other related works

04 **Summary**

Contributions and future work

$TKEM_1$, $TKEM_2$, TPKE: Settings and Tradeoffs

Property	$TKEM_1$	$TKEM_2$, TPKE
Setting	**Synchronous**	**Asynchronous**
Decryptors	**Any polynomial N**	**Small N**
Secret sharing	Shamir secret sharing	Replicated secret sharing (RSS)
Leakage technique	Adaptive Hint-MLWE	Adaptive LHL with noisy leakage
Modulus	$O(\sqrt{\ell}^* \cdot \mathbf{poly}(\lambda))$	$O(N^{t/2} \cdot \sqrt{\ell} \cdot \mathbf{poly}(\lambda))$
Adaptive CCA & Active Security	✓	✓
Robustness	**Not provided**	**Provided for TPKE**

* ℓ denotes the number of partial-decryption queries.

Efficiency Estimates for $TKEM_1$ and $TKEM_2$, TPKE

Parameter settings at 128-bit security

Representative setting	$TKEM_1$	$TKEM_2$, TPKE
Parties and threshold	$N = 1024,\ t \leq 1024$	$N = 16,\ t = 2$
Decryption-query bound	$\ell = 2^{60}$	$\ell = 2^{60}$
Ciphertext (including NIZK proof)	**101 KB**	**158 KB**
Partial decryption per party	**1.87 KB**	**38 KB***

* The 38 KB estimate applies to $TKEM_2$. For TPKE, it does not include the additional NIZK proof required for robustness.

Comparison with Related Work

Comparison with existing lattice-based threshold KEM/PKE schemes from lattices

Scheme	Non-interactivity	CCA	Adaptive Corruptions	Modulus	Active Security	Setting
LP25 (Amber)	×	✓	×	polynomial	✓	sync.
BPLP26 (Amber)	×	✓	×	polynomial	✓	sync.
HKMN26	✓	✓	×	polynomial	×	async.
DLN+21	✓	✓	✓	**exponential**	✓	async.
Our $TKEM_1$	✓	✓	✓	**polynomial**	✓	**sync.**
Our $TKEM_2$, TPKE	✓	✓	✓	**polynomial**	✓	**async.**

THIS WORK

The first lattice-based Threshold KEM/PKE achieving IND-CCA security under adaptive corruptions and polynomial modulus, simultaneously.

Outline

Adamas

Threshold KEM/PKE

01 **Introduction**

Syntax, CCA & active security under adaptive corruptions

02 **Our Proposal**

$TKEM_1$, $TKEM_2$, TPKE and proof ideas

03 **Performance of Our Proposal**

Parameters and communication costs, comparison with other related works

04 **Summary**

Contributions and future work

Summary and Outlook

MAIN RESULT

One-round Lattice-based TKEM/TPKE under adaptive corruptions with polynomial modulus

TWO SCHEMES

$TKEM_1$

- Synchronous setting
- Polynomially many decryptors

$TKEM_2$, TPKE

- Asynchronous setting for small N
- TPKE achieves robustness

NEXT STEPS

- Optimized open-source implementation
- More efficient short-coefficient, short-share secret sharing for $TKEM_2$,TPKE

NEW TECHNIQUE

Adaptive LHL under noisy linear leakage

Thank you!

ADAMAS

ePrint: ia.cr/2026/1627