IND-CCA Lattice Threshold KEM under 30 KiB

Katharina Boudgoust

joint work with Sasha Lapiha, Rafaël del Pino and Thomas Prest

Contributions

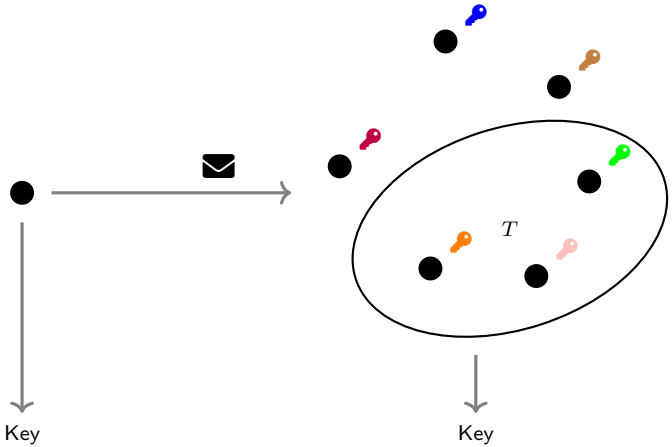
We build the first **actively secure** threshold key encapsulation mechanism based on **lattices** with ciphertext sizes **under 30 KiB**!

- CCA2-security based on Ring-LWE and NTRU
- Decapsulation consistency
- Threshold $T \leq 32$ and number of parties $N \leq 44$
- Number of decryption queries $Q = 2^{25}$

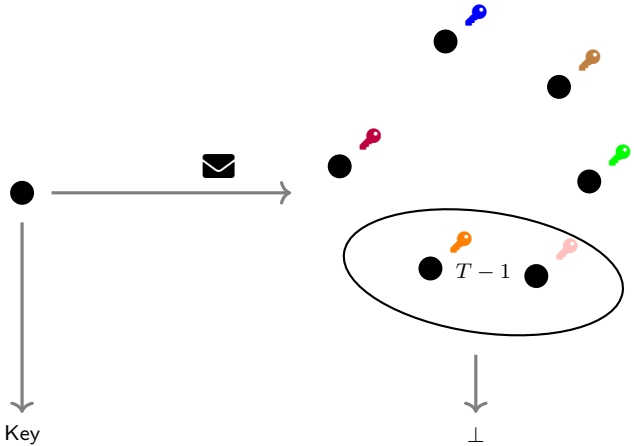
In a nutshell: significantly improve the threshold identity-based encryption by [LP25]* and plug it into their BCHK+ transform.

*Lapiha and Prest, *A Lattice-Based IND-CCA Threshold KEM from the BCHK+ Transform*, Asiacrypt'25

Threshold Key Encapsulation Mechanism (ThKEM)



Threshold Key Encapsulation Mechanism (ThKEM)



Active security:

- CCA2-security
- Decapsulation consistency

Applications:

E-Voting

Mempools

How to build a lattice ThKEM with active security?

- Take a CCA2-KEM and "thresholdize" it
  FO-transform in Kyber, requires generic MPC/ThFHE
- Take a CPA-ThKEM and add NIZK's à la Naor-Yung
  Expected to be very inefficient, but no concrete baseline
- BCHK+ transform by [LP25]*
  Seems most promising for now! But they reported 540 KiB ciphertexts . . .

*Lapiha and Prest, *A Lattice-Based IND-CCA Threshold KEM from the BCHK+ Transform*, Asiacrypt'25

Threshold Identity-Based Encryption (ThIBE)

- $\mathsf{KGen} \to (\mathsf{pk}, (\mathsf{msk}_i)_{i \in [N]})$
- $\mathsf{Enc}(m, \mathsf{pk}, \mathsf{id}) \to \mathsf{ct}$
- $\mathsf{ShareExtract}(\mathsf{msk}_i, \mathsf{id}) \to p_i$
- $\mathsf{Combine}(\mathsf{id}, \mathsf{ct}, (p_i)_{i \in S}) \to \{ m, \perp \}$
- $\mathsf{ShareVerify}(\mathsf{pk}, p_i, \mathsf{id}) \to \{ 0, 1 \}$

Correctness for $|S| \geq T$

Security for $|S| < T$

Threshold Identity-Based Encryption (ThIBE)

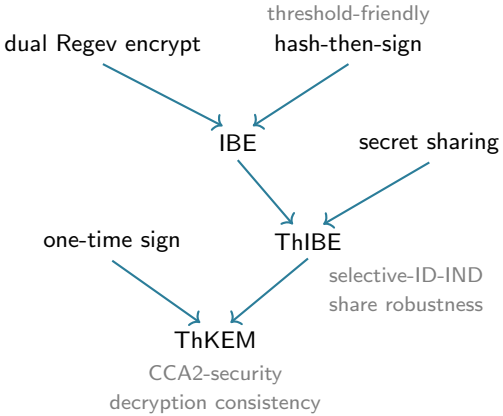
- $\mathsf{KGen} \to (\mathsf{pk}, (\mathsf{msk}_i)_{i \in [N]})$
- $\mathsf{Enc}(m, \mathsf{pk}, \mathsf{id}) \to \mathsf{ct}$
- $\mathsf{ShareExtract}(\mathsf{msk}_i, \mathsf{id}) \to p_i$
- $\mathsf{Combine}(\mathsf{id}, \mathsf{ct}, (p_i)_{i \in S}) \to \{ m, \perp \}$
- $\mathsf{ShareVerify}(\mathsf{pk}, p_i, \mathsf{id}) \to \{ 0, 1 \}$

Correctness for $|S| \geq T$

Security for $|S| < T$

1) Selective-ID-IND Security:
   - Fix target identity id^*
   - Oracle: key extraction for all $\mathsf{id} \neq \mathsf{id}^*$
   - Goal: distinguish encryption of m_0/m_1 for id^*
2) Share Robustness
   - For honest ciphertexts
   - If all p_i's pass ShareVerify, then Combine succeeds

High Level


threshold-friendly
dual Regev encrypt
hash-then-sign
IBE
secret sharing
one-time sign
ThIBE
selective-ID-IND
share robustness
ThKEM
CCA2-security
decryption consistency

Starting IBE - High Level [CHKP10]*

$R_q = \mathbb{Z}_q[X]/(X^d + 1)$ power-of-two cyclotomic ring, $H : \{0,1\}^* \to R_q$ random oracle

- KGen:
  - random target $t \in R_q$ — $\mathsf{pk} = (t, \mathbf{A})$
  - public matrix $\mathbf{A} \in R_q^{1 \times k}$ — $\mathsf{msk} = \mathbf{T}$
  - secret trapdoor $\mathbf{T} \in R_q^{k \times 1}$ — allows finding short preimages for $[\mathbf{A} \mid \star]$
- Enc(pk, m, id):
  - embed identity $\mathbf{A}_{\mathsf{id}} = [\mathbf{A} \mid H(\mathsf{id})]$
  - standard LWE-encryption to $(\mathbf{A}_{\mathsf{id}}, t)$ — $\mathsf{ct} = (\mathbf{u}, v)$
    - $\mathbf{u}^T = r\,\mathbf{A}_{\mathsf{id}} + \mathbf{f}^T$
    - $v = r\,t + f' + \lfloor q/2 \rceil\, m$
- Extract(msk, id):
  - use $\mathbf{T}$ to find short $\mathbf{z}$ such that $\mathbf{A}_{\mathsf{id}} \cdot \mathbf{z} = t$ — $\mathsf{sk}_{\mathsf{id}} = \mathbf{z}$
- Dec(ct, $\mathsf{sk}_{\mathsf{id}}$):
  - compute $v - \mathbf{u}^T\mathbf{z}$
  - then round to the highest order bit

$$v - \mathbf{u}^T\mathbf{z} = f' - \mathbf{f}^T\mathbf{z} + \lfloor q/2 \rceil\, m$$

* Cash, Hofheinz, Kiltz, Peikert, *Bonsai Trees, or How to Delegate a Lattice Basis*, Eurocrypt'10

Starting IBE - High Level [CHKP10]*

$R_q = \mathbb{Z}_q[X]/(X^d + 1)$ power-of-two cyclotomic ring, $H : \{0,1\}^* \to R_q$ random oracle

- KGen:
  - random target $t \in R_q$
  - public matrix $\mathbf{A} \in R_q^{1 \times k}$
  - secret trapdoor $\mathbf{T} \in R_q^{k \times 1}$
- Enc(pk, m, id):
  - embed identity $\mathbf{A}_{\mathsf{id}} = [\mathbf{A} \mid H(\mathsf{id})]$
  - standard LWE-encryption to $(\mathbf{A}_{\mathsf{id}}, t)$
    - $\mathbf{u}^T = r\,\mathbf{A}_{\mathsf{id}} + \mathbf{f}^T$
    - $v = r\,t + f' + \lfloor q/2 \rceil\, m$
- Extract(msk, id):
  - use $\mathbf{T}$ to find short $\mathbf{z}$ such that $\mathbf{A}_{\mathsf{id}} \cdot \mathbf{z} = t$
- Dec(ct, $\mathsf{sk}_{\mathsf{id}}$):
  - compute $v - \mathbf{u}^T\mathbf{z}$
  - then round to the highest order bit

Main improvement over [LP25]:

1) simpler design of $\mathbf{A}_{\mathsf{id}}$
2) embedd NTRU trapdoor in $H(\mathsf{id})$

* Cash, Hofheinz, Kiltz, Peikert, *Bonsai Trees, or How to Delegate a Lattice Basis*, Eurocrypt'10

Thresholdize Extract - High Level

Challenge: Build trapdoor $\mathbf{T}$ in a threshold-friendly manner $\rightarrow$ à la Plover [EEN+24]*

- KGen:
  - random target $t \in R_q$
  - $\mathbf{A} \approx$ LWE instance
  - $\mathbf{T} \approx$ LWE secrets
  - $\mathsf{msk}_i \approx$ secret-shares of $\mathbf{T}$
- ShareExtract:
  - $\mathbf{A}_{\mathsf{id}} = [\mathbf{A} \mid H(\mathsf{id})]$
  - use Schnorr-type 3-round protocol to obtain $\mathbf{z}$ such that $\mathbf{A}_{\mathsf{id}}\,\mathbf{z} = t$

Secret sharing:

- [LP25]: Shamir secret sharing with zero-masks à la ThRaccoon
  scales well for large N, but no share robustness
- **Ours**: Vandermonde secret sharing with short shares and short reconstruction coefficients
  share robustness, but limited to $T \leq N \leq 64$

*Esgin, Espitau, Niot, Prest, Sakzad, Steinfeld, *Plover: Masking-Friendly Hash-and-Sign Lattice Signatures*, Eurocrypt'24

Concrete Numbers of our ThIBE for $T \leq 32$

Bits of security	Robust?	N	# Queries	|pk|	|ct| (KiB)
128	No	64	2^{46}	6.7	28.5
128	Yes	44	2^{25}	8.2	30.4
256	No	64	2^{44}	13.9	57.1
256	Yes	44	2^{22}	16.5	61.7

🙏 Small bug in our parameter script ⇒ numbers slightly changed with respect to the published version

➔ E-print version updated ia.cr/2026/021

Wrapping Up

We build the first **actively secure** threshold key encapsulation mechanism based on **lattices** with ciphertext sizes **under 30 KiB**!

- CCA2-security based on Ring-LWE and NTRU
- Decapsulation consistency
- Threshold T and number of parties N such that $T \leq N \leq 64$
- Number of decryption queries $Q = 2^{25}$

Follow-up: Planing to submit an even more optimized version, called **Amber** to NIST's Multi-Party Threshold Cryptography call.

Thanks!

We build the first **actively secure** threshold key encapsulation mechanism based on **lattices** with ciphertext sizes **under 30 KiB**!

- CCA2-security based on Ring-LWE and NTRU
- Decapsulation consistency
- Threshold T and number of parties N such that $T \leq N \leq 64$
- Number of decryption queries $Q = 2^{25}$

Follow-up: Planing to submit an even more optimized version, called **Amber** to NIST's Multi-Party Threshold Cryptography call.

David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert.
Bonsai trees, or how to delegate a lattice basis.
In *EUROCRYPT*, Lecture Notes in Computer Science, pages 523–552. Springer, 2010.

Muhammed F. Esgin, Thomas Espitau, Guilhem Niot, Thomas Prest, Amin Sakzad, and Ron Steinfeld.
Plover: Masking-friendly hash-and-sign lattice signatures.
In *EUROCRYPT (6)*, Lecture Notes in Computer Science, pages 316–345. Springer, 2024.

Oleksandra Lapiha and Thomas Prest.
A lattice-based IND-CCA threshold KEM from the BCHK+ transform.
pages 461–494, 2025.