Post-Quantum Multi-Recipient KEMs and their Applications

Shuichi Katsumata
AIST, JP

Kris Kwiatkowski
PQShield, UK

Federico Pintore
Univ. of Oxford, UK

Eamonn Postlethwaite
Royal Holloway, UK

Thomas Prest
PQShield, UK/FR

2021 EWHA-KMS International Workshop on Cryptography

Main question

How efficiently can we share a session key K between (N + 1) users?

- **Motivation:** Secure group messaging
- **Naive solution with El Gamal:**
  - Send $(g^{r_i}, pk_i^{r_i} \cdot K)$ for each user i
- **Variant by Kurosawa [Kur02]:**
  - Send $(g^r, pk_1^r \cdot K, \ldots, pk_N^r \cdot K)$
  - Asymptotically, saves a factor **2**

Main question

How efficiently can we share a session key K between (N + 1) users?

- **Terminology:** ciphertext compression, mKEM/mPKE, randomness reuse, etc.
- **Two flavors:**
  - Single-message (this work)
  - Multi-message (send a distinct K_i to each user)
- [BBM00, BPS00, Kur02, BBS03, Sma05, HK07, BF07, HTAS09, MH13, Yan15]
- No* post-quantum proposal

This work

- **Revisiting mPKEs & mKEMs**
  - More natural definition
  - Captures classical and post-quantum assumptions
  - QROM security
- **Instantiation from post-quantum assumptions**
  - Lattices
  - Isogenies
  - Efficiency increased by one or two orders of magnitude
- **Application to TreeKEM**
  - Interplay mKEM $\times$ TreeKEM
  - Communication cost divided by 2
- **Focus on lattice-based mKEMs**
  - Concrete security analysis in the mKEM regime
  - New lattice-based mKEMs

Revisiting mPKEs & mKEMs

Full reproducibility [BBS03]:

$$\begin{tikzcd}
\mathrm{Enc}(\mathtt{pk}_1, \mathtt{M}, r) \arrow[rr, "\mathcal{A}(\,\cdot\,, \mathtt{pk}_1, \mathtt{pk}_2, \mathtt{sk}_2, \mathtt{M}')"] & & \mathrm{Enc}(\mathtt{pk}_2, \mathtt{M}', r)
\end{tikzcd}$$

Weak reproducibility [BF05]:

$$\begin{tikzcd}
\mathrm{Enc}(\mathtt{pk}_1, \mathtt{M}, r) \arrow[rr, "\mathcal{A}(\,\cdot\,, \mathtt{pk}_1, \mathtt{pk}_2, \mathtt{sk}_2, \mathtt{M})"] \arrow[dr, "\mathcal{A}(\,\cdot\,, \mathtt{pk}_2, \mathtt{pk}_3, \mathtt{sk}_3, \mathtt{M}')"'] & & \mathrm{Enc}(\mathtt{pk}_2, \mathtt{M}, r) \arrow[dl, "\mathcal{A}(\,\cdot\,, \mathtt{pk}_1, \mathtt{pk}_3, \mathtt{sk}_3, \mathtt{M}')"] \\
& * &
\end{tikzcd}$$

Decomposability (this work):

$$\mathrm{Enc}(\mathtt{pk}_i, \mathtt{M}) = \overbrace{\mathtt{ct}_0}^{\mathrm{Enc}^{\mathtt{ind}}(r_0)} \; \overbrace{\widehat{\mathtt{ct}}_i}^{\mathrm{Enc}^{\mathtt{dep}}(\mathtt{pk}_i, \mathtt{M}, r_0, r_i)}$$

	Fully rep.	Weakly rep.	Decomposable
El Gamal	✓	✓	✓
LP [LP11]	?	?	✓
SIDH [JD11, DJP14]	?	?	✓
CSIDH	?	?	✓

Example: El Gamal. Let a ciphertext $\mathtt{ct} = (g^r, \mathsf{pk}_1^r \cdot \mathtt{M})$ with $\mathsf{pk}_1 = g^{\mathsf{sk}_1}$.

- **Full reproducibility:** $(g^r, *) \longrightarrow \left(g^r, (g^r)^{\mathsf{sk}_2} \cdot \mathtt{M}'\right)$.
- **Decomposability:** $(\mathtt{ct}_0 = g^r, \widehat{\mathtt{ct}}_1 = \mathsf{pk}_1^r \cdot \mathtt{M})$.

A ciphertext with N recipients will be $\vec{\mathtt{ct}} = (\mathtt{ct}_0, \widehat{\mathtt{ct}}_1, \ldots, \widehat{\mathtt{ct}}_N)$.
Key generation and decryption remain the same.

Generic Transforms

[BBS03]	**Fully rep.** IND-CPA PKE	➡	**Multi**-msg IND-**CPA** mPKE	SM
[BBS03]	**Fully rep.** IND-**CCA** PKE	➡	**Multi**-msg IND-CCA mPKE	SM
[BF07]	**Weakly rep.** IND-CPA PKE	➡	**Single**-msg IND-CCA mPKE	ROM
Our work	**Decomposable single**-msg IND-CPA **m**PKE	➡	**Single**-msg IND-CCA mKEM	QROM

Generic Transforms

[BBS03]	**Fully rep.** IND-CPA PKE	➜	**Multi**-msg IND-**CPA** mPKE	SM
[BBS03]	**Fully rep.** IND-**CCA** PKE	➜	**Multi**-msg IND-CCA mPKE	SM
[BF07]	**Weakly rep.** IND-CPA PKE	➜	**Single**-msg IND-CCA mPKE	ROM
Our work	**Decomposable single**-msg IND-CPA **m**PKE	➜	**Single**-msg IND-CCA mKEM	QROM

Open question: Can we (dis)prove the following statement?

(Decomposable **multi**-msg IND-CPA mPKE) ➜ (**Multi**-msg IND-CCA mKEM)

Decomp. IND-CPA mPKE $\Rightarrow$ IND-CCA mKEM

Encaps$(\{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\})$

1. Generate a random M
2. $\mathsf{ct}_0 \leftarrow \mathsf{Enc}^{\mathsf{ind}}(G_1(\mathsf{M}))$
3. For $i = 1, \ldots, N$:
   - $\widehat{\mathsf{ct}}_i \leftarrow \mathsf{Enc}^{\mathsf{dep}}(\mathsf{pk}_i, \mathsf{M}, G_1(\mathsf{M}), G_2(\mathsf{pk}_i, \mathsf{M}))$
4. $\mathsf{K} := H(\mathsf{M})$
5. Return $(\mathsf{K}, \vec{\mathsf{ct}} := (\mathsf{ct}_0, (\widehat{\mathsf{ct}}_i)_{i \in [N]}))$

Decaps$(\mathsf{pk}_i, \mathsf{ct} = (\mathsf{ct}_0, \widehat{\mathsf{ct}}_i))$

1. $\mathsf{M} \leftarrow \mathsf{Dec}(\mathsf{sk}_i, \mathsf{ct})$
2. If $M = \perp$, return $\mathsf{K} := \perp$
3. $\mathsf{ct}_0 \leftarrow \mathsf{Enc}^{\mathsf{ind}}(G_1(\mathsf{M}))$
4. $\widehat{\mathsf{ct}}_i \leftarrow \mathsf{Enc}^{\mathsf{dep}}(\mathsf{pk}_i, \mathsf{M}, G_1(\mathsf{M}), G_2(\mathsf{pk}_i, \mathsf{M}))$
5. If $(\mathsf{ct}_0, \widehat{\mathsf{ct}}_i) \neq \mathsf{ct}$, return $\mathsf{K} := \perp$
6. Return $\mathsf{K} = H(\mathsf{M})$

- G_1, G_2 are PRFs, H is a hash function, all are modeled as random oracles.
- QROM proof uses compressed oracles [Zha19].
- We can achieve implicit rejection as well.

Instantiation from Post-Quantum Assumptions

The Lindner–Peikert framework [LP11]

Keygen ($\mathbf{A} \in \mathcal{R}_q^{m\times m}$)

1. Sample short matrices $\mathbf{S}, \mathbf{E}$
2. $\mathbf{B} \leftarrow \mathbf{AS} + \mathbf{E}$
3. $\mathsf{sk} := (\mathbf{S}, \mathbf{E}), \mathsf{pk} := \mathbf{B}$

Enc$(\mathsf{pk}, \mathtt{M})$

1. Sample short matrices $\mathbf{R}, \mathbf{E}', \mathbf{E}''$
2. $\mathbf{U} \leftarrow \mathbf{RA} + \mathbf{E}'$
3. $\mathbf{V} \leftarrow \mathbf{RB} + \mathbf{E}'' + \mathrm{Encode}(\mathtt{M})$
4. $\mathtt{ct} := (\mathbf{U}, \mathbf{V})$

Dec$(\mathsf{sk}, \mathtt{ct})$

1. $\mathtt{M} \leftarrow \mathrm{Decode}(\mathbf{V} - \mathbf{US})$

Encompasses these NIST Round 3 candidates:

- FrodoKEM
- Kyber
- NTRU LPRime
- Saber

The Lindner-Peikert framework is decomposable:

- Use the same $\mathbf{A}$ for all public keys.
- $\mathbf{U}$ is then independent of $\mathtt{pk}$ and $\mathtt{M}$.

$\mathsf{Enc}(\mathsf{pk} = \mathbf{B}, \mathtt{M})$

1. Sample short matrices $\mathbf{R}, \mathbf{E}', \mathbf{E}''$
2. $\mathbf{U} \leftarrow \mathbf{R}\mathbf{A} + \mathbf{E}'$
3. $\mathbf{V} \leftarrow \mathbf{R}\mathbf{B} + \mathbf{E}'' + \mathsf{Encode}(\mathtt{M})$
4. $\mathtt{ct} := (\mathbf{U}, \mathbf{V})$

The Lindner-Peikert framework is decomposable:

- → Use the same $\mathbf{A}$ for all public keys.
- → $\mathbf{U}$ is then independent of pk and $\mathtt{M}$.

$\mathsf{Enc}(\mathsf{pk} = \mathbf{B}, \mathtt{M})$

1. Sample short matrices $\mathbf{R}, \mathbf{E}', \mathbf{E}''$
2. $\mathbf{U} \leftarrow \mathbf{RA} + \mathbf{E}'$
3. $\mathbf{V} \leftarrow \mathbf{RB} + \mathbf{E}'' + \mathsf{Encode}(\mathtt{M})$
4. $\mathtt{ct} := (\mathbf{U}, \mathbf{V})$

$\Longrightarrow$

$\mathsf{MultiEnc}(\{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}, \mathtt{M})$

1. Sample short matrices $\mathbf{R}, \mathbf{E}'$
2. $\mathbf{U} \leftarrow \mathbf{RA} + \mathbf{E}'$
3. For $i = 1, \ldots, N$:
   1. Sample a short matrix $\mathbf{E}''_i$
   2. $\mathbf{V}_i \leftarrow \mathbf{RB}_i + \mathbf{E}''_i + \mathsf{Encode}(\mathtt{M})$
4. $\vec{\mathtt{ct}} := (\mathbf{U}, \mathbf{V}_1, \ldots, \mathbf{V}_N)$

Each $\mathbf{V}_i$ is much smaller and faster to compute than $\mathbf{U}$:

- → Shorter dimensions
- → Bit dropping

Security reduces to LWE with many samples (see end of the talk).

SIDH [JD11, DJP14] and SIKE

- E is an elliptic curve
- $E[\ell_A^a] = \langle P_A, Q_A \rangle$
- $E[\ell_B^b] = \langle P_B, Q_B \rangle$

Keygen(E, P_A, Q_A, P_B, Q_B)

1. $\mathtt{sk} := \psi$, where $\psi : E \to E/\langle R_B \rangle$ is an isogeny of kernel R_B
2. $\mathtt{pk} := (E/\langle R_B \rangle, \psi(P_A), \psi(Q_A))$

Enc$(\mathtt{pk}, \mathtt{M})$

1. Sample an isogeny $\varphi : E \to E/\langle R_A \rangle$
2. $\mathtt{ct}_0 = (E/\langle R_A \rangle, \varphi(P_B), \varphi(Q_B))$
3. Compute $j = \text{j-Inv}(E/\langle R_A, R_B \rangle)$
4. $\widehat{\mathtt{ct}} = j \oplus \mathtt{M}$
5. $\mathtt{ct} := (\mathtt{ct}_0, \widehat{\mathtt{ct}})$

Dec$(\mathtt{sk}, \mathtt{ct})$

1. Compute $j = \text{j-Inv}(E/\langle R_A, R_B \rangle)$
2. $\mathtt{M} = j \oplus \widehat{\mathtt{ct}}$

SIDH [JD11, DJP14] and SIKE

- E is an elliptic curve
- $E[\ell_A^a] = \langle P_A, Q_A \rangle$
- $E[\ell_B^b] = \langle P_B, Q_B \rangle$

Keygen(E, P_A, Q_A, P_B, Q_B)

1. $\mathtt{sk}_i := \psi_i$, where $\psi_i : E \to E/\langle R_B \rangle$ is an isogeny of kernel $R_B^{(i)}$
2. $\mathtt{pk} := (E/\langle R_B^{(i)} \rangle, \psi_i(P_A), \psi_i(Q_A))$

Security reduces to SSDDH [DJP14].

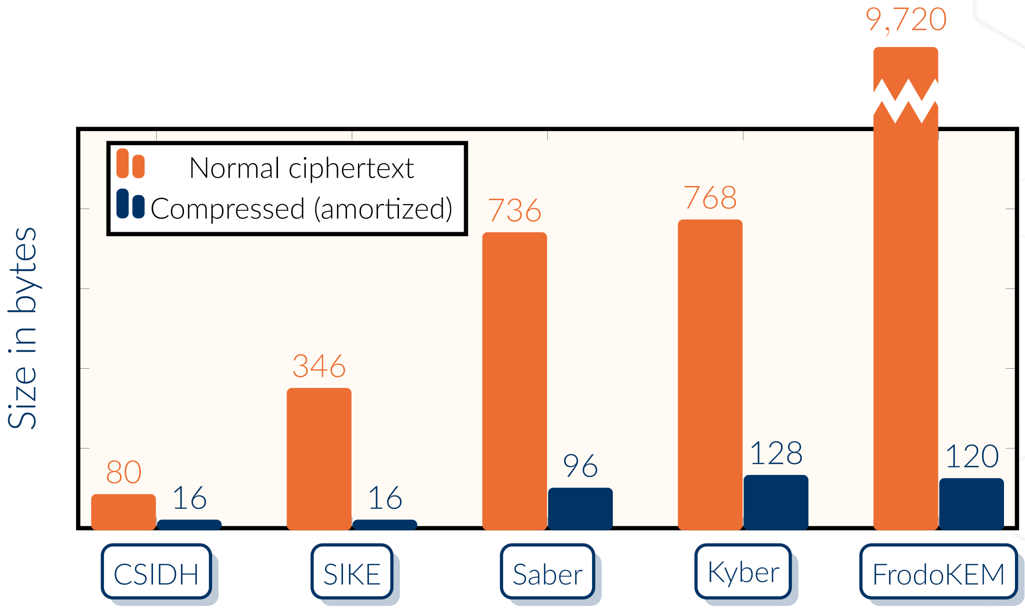
Enc$(\{\mathtt{pk}_1, \ldots, \mathtt{pk}_N\}, \mathtt{M})$

1. Sample an isogeny $\varphi : E \to E/\langle R_A \rangle$
2. $\mathtt{ct}_0 = (E/\langle R_A \rangle, \varphi(P_B), \varphi(Q_B))$
3. For $i = 1, \ldots, N$:
   1. Compute $j_i = \text{j-Inv}(E/\langle R_A, R_B^{(i)} \rangle)$
   2. $\widehat{\mathtt{ct}}_i = j_i \oplus \mathtt{M}$
4. $\vec{\mathtt{ct}} := (\mathtt{ct}_0, \widehat{\mathtt{ct}}_1, \ldots, \widehat{\mathtt{ct}}_N)$

Dec$(\mathtt{sk}_i, (\mathtt{ct}_0, \widehat{\mathtt{ct}}_i))$

1. Compute $j_i = \text{j-Inv}(E/\langle R_A, R_B^{(i)} \rangle)$
2. $\mathtt{M} = j_i \oplus \widehat{\mathtt{ct}}_i$

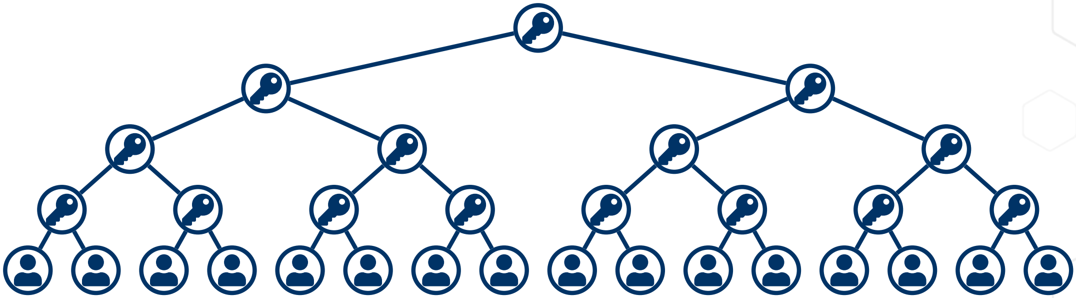
Impact on 1 PKE + 4 KEMs (NIST level I)



Application to TreeKEM

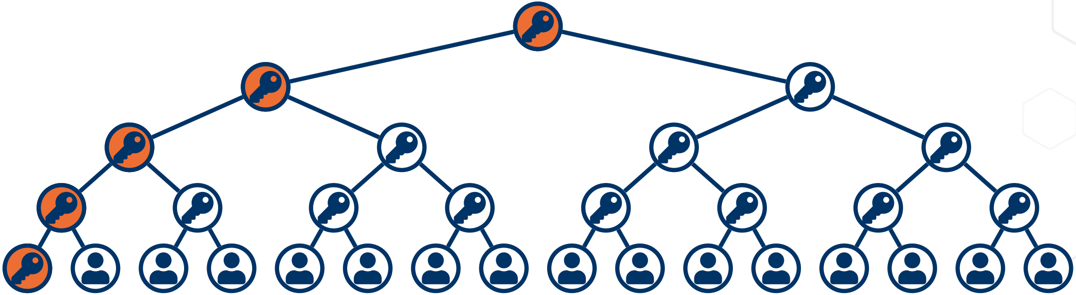
TreeKEM [BBR18, BBM+20, OBR+20, ACDT20]:

- → Key component of the MLS draft IETF proposal for group messaging
- → The N users are arranged as leaves of a (binary) tree
- → **TreeKEM invariant:** 👤 knows a private 🔑 if and only if it is in its path.



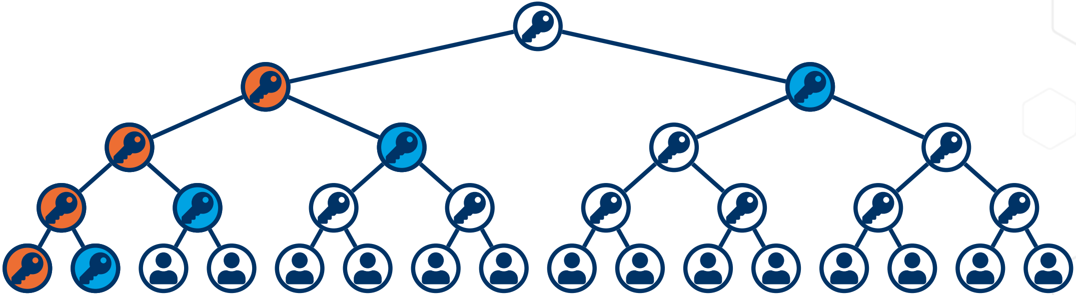
TreeKEM [BBR18, BBM+20, OBR+20, ACDT20]:

- Key component of the MLS draft IETF proposal for group messaging
- The N users are arranged as leaves of a (binary) tree
- **TreeKEM invariant:** knows a private if and only if it is in its path.



TreeKEM [BBR18, BBM+20, OBR+20, ACDT20]:

- Key component of the MLS draft IETF proposal for group messaging
- The N users are arranged as leaves of a (binary) tree
- **TreeKEM invariant:** 👤 knows a private 🔑 if and only if it is in its path.

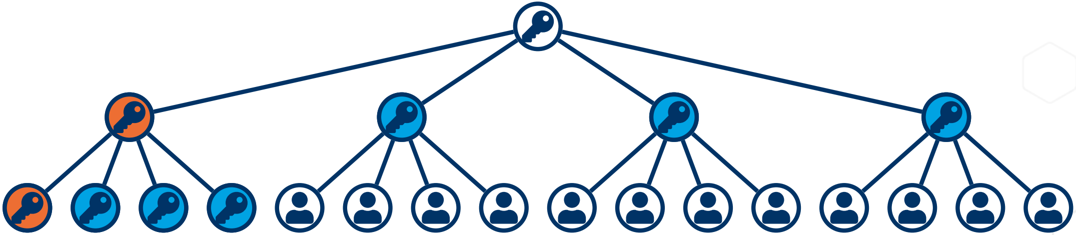


Post-compromise security: Users refresh their key material by broadcasting an update package that contains:

- One `pk` for each node in the path (except the root).
- One `ct` for each node in the co-path (siblings of nodes in the path).

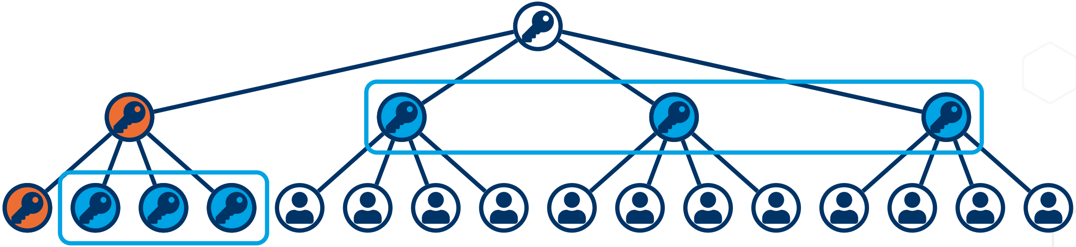
What if we use a m-ary tree instead of a binary tree?

- We send $\log_m(N)$ public keys and $(m-1) \cdot \log_m(N)$ ciphertexts
- However all ciphertexts at a same level encapsulate the same key!
- We can use a single mKEM ciphertext at each level



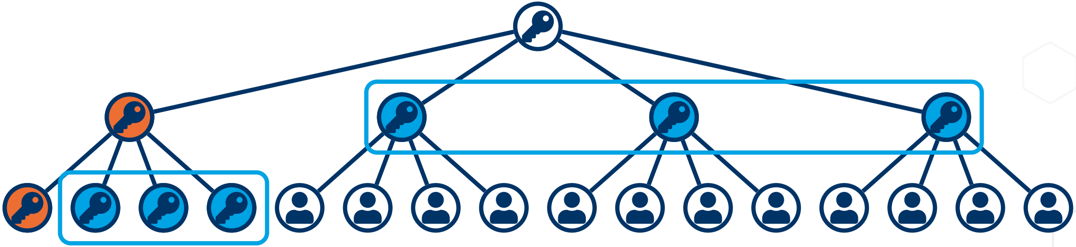
What if we use a m-ary tree instead of a binary tree?

- → We send $\log_m(N)$ public keys and $(m-1) \cdot \log_m(N)$ ciphertexts
- → However all ciphertexts at a same level encapsulate the same key!
- → We can use a single mKEM ciphertext at each level



What if we use a *m*-ary tree instead of a binary tree?

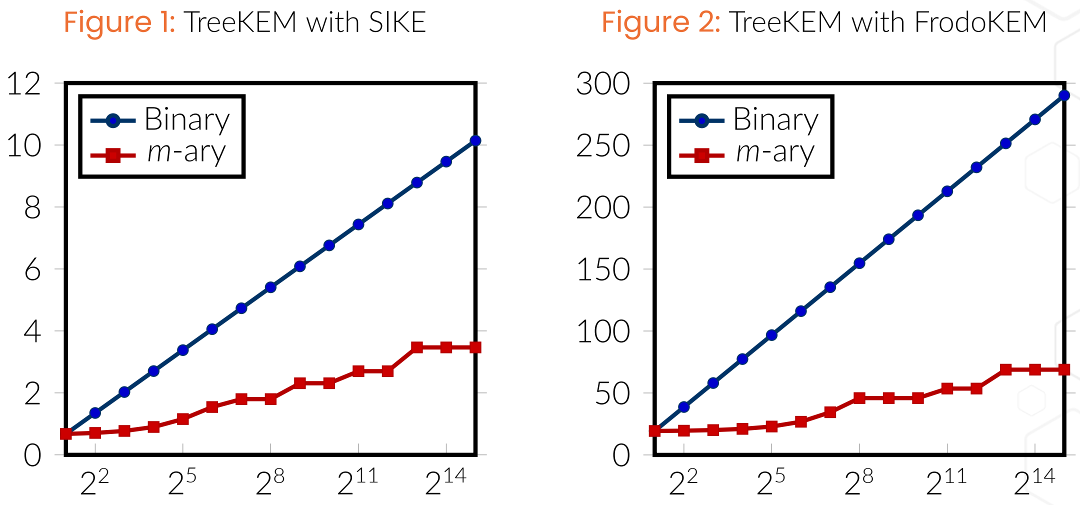
- We send $\log_m(N)$ public keys and $(m-1)\cdot\log_m(N)$ ciphertexts
- However all ciphertexts at a same level encapsulate the same key!
- We can use a single mKEM ciphertext at each level



Size of an update package:

- **Standard TreeKEM:** $\log_2(N)\cdot(|\mathsf{pk}|+|\mathsf{ct}_0|+|\widehat{\mathsf{ct}}_i|)$
- **m-ary trees + mKEM:** $\log_m(N)\cdot(|\mathsf{pk}|+|\mathsf{ct}_0|+|\widehat{\mathsf{ct}}_i|\cdot m)$

Size of an update package in kilobytes as a function of number of users (NIST level I)



Note for later: for all examples, the package size is $\propto \log N$.

Focus on Lattice-Based mKEMs

We study the concrete security of the mKEMs based on:

- Kyber
- Saber
- FrodoKEM

We also propose two (more) efficient lattice-based mKEMs:

- **BilboKEM** (based on FrodoKEM)
- **Ilum** (based on Kyber)

All schemes in this slide instantiate the Lindner-Peikert framework.

- Security for key recovery: LWE/LWR with finite samples
- Security for encryption: LWE/LWR with **unbounded** samples

⚠ Work in progress

How is it related to LWE?

MultiEnc$(\{\mathsf{pk}_1, \ldots, \mathsf{pk}_N\}, \mathtt{M})$

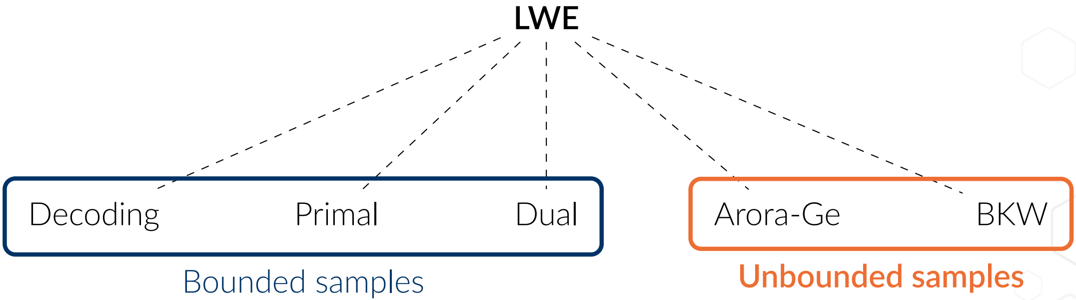
1. Sample short matrices $\mathbf{R}, \mathbf{E}', \{\mathbf{E}''_1, \ldots, \mathbf{E}''_N\}$
2. $\mathbf{U} \leftarrow \mathbf{R}\mathbf{A} + \mathbf{E}'$
3. For $i = 1, \ldots, N$:
   1. $\mathbf{V}_i \leftarrow \mathbf{R}\mathbf{B}_i + \mathbf{E}''_i + \mathrm{Encode}(\mathtt{M})$
4. $\vec{\mathtt{ct}} := (\mathbf{U}, \mathbf{V}_1, \ldots, \mathbf{V}_N)$

Attacker's goal: Given $(\mathbf{U}, \mathbf{V}_1, \ldots, \mathbf{V}_N)$, recover $\mathtt{M}$.

Note that:

$$(\mathbf{U} \| \mathbf{V}_1 \| \ldots \| \mathbf{V}_N) = \mathbf{R} \times (\mathbf{A} \| \mathbf{B}_1 \| \ldots \| \mathbf{B}_N) + (\mathbf{E}' \| \mathbf{E}''_1 \| \ldots \| \mathbf{E}''_N) + (\mathbf{0} \| \mathtt{M} \| \ldots \| \mathtt{M})$$

Attacks Against LWE (Very Contained)



Methodology

Primal attack:

- Leaky-LWE framework: `https://github.com/lducas/leaky-LWE-Estimator/tree/NIST-round3`
- Determining the block Size: we use the PIM (as opposed to GSA)
- Fitting function for "dimensions for free": same as Kyber/FrodoKEM

Arora-Ge:

- LWE estimator: `https://lwe-estimator.readthedocs.io/en/latest/_apidoc/estimator/estimator.arora_gb.html`
- Take rounding into account

BKW:

- Based on Coded-BKW
- Additional rounding **not** taken into account
- Next step: [BGJ+20] and quantum

Attacks with Unbounded Samples

Arora-Ge:

- Algebraic attack (linearization)
- Requires $n^{O(d)}$ samples, where $d = \max(\|\mathbf{E}'\|_\infty, \|\mathbf{E}_i''\|_\infty)$

BKW:

- Combinatorial attack (lattice reduction + guessing)
- Complexity: $2^{\tilde{O}(n)}$

Scheme	$\|\mathbf{E}'\|_\infty$	$\|\mathbf{E}_i''\|_\infty$
Kyber	≥ 5	≥ 100
Saber	$= 8$	≥ 16
FrodoKEM	≥ 13	≥ 13
NTRU LPRime	$= 3$	≥ 288

In effect, the $\|\mathbf{E}_i''\|_\infty$ are large thanks to bit dropping in the $\mathbf{V}_i$.

- Rules out Arora-Ge and BKW in practice!

Security Metrics at NIST Level 1 ($\approx$ AES-128)

S = Samples, C = Classical, Q = Quantum, G = Gates, O = Operations.

	Primal			Arora-Ge			BKW		
Scheme	S.	CG	QG	S	CO	QO	S	CO	QO
Saber	512	152	142	∞	2646	TBD	∞	158	TBD
Kyber	512	151	143	∞	2408	TBD	∞	147	TBD
FrodoKEM	656	175	164	∞	2798	TBD	∞	226	TBD

Same trends at high security levels.

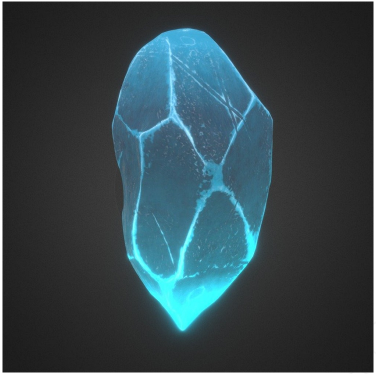
BilboKEM and Ilum

We propose lattice-based mKEMS that are tailored to mKEM constraints:

- **"Standard":** Minimize $|\widehat{\mathtt{ct}}_i|$
- **TreeKEM:** Minimize $\tau = \min_m \frac{|\mathtt{pk}|+|\mathtt{ct}_0|+|\widehat{\mathtt{ct}}_i|\cdot m}{\log_2(m)}$

BilboKEM (variant of FrodoKEM)

Ilum (variant of Kyber)



How do we minimize the bitsize of the V_i?

Recall that $\mathtt{M} = \mathrm{Decode}(\mathbf{V}_i - \mathtt{sk}_i \cdot \mathbf{U})$.

We leverage the following tools:

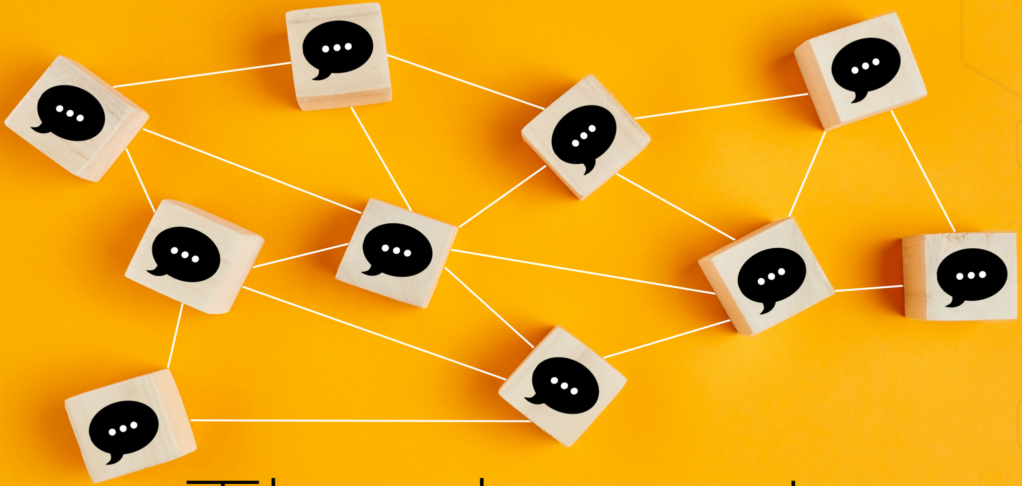
- **Bit dropping:** drop the least significant bits of $\mathbf{V}_i$
  - Reduces the size of $\mathbf{V}_i$, increase the LWE error rate
  - Increases the decryption failure rate
- **Coefficent dropping:** drop superflous coefficients of $\mathbf{V}$
  - Reduces the size of $\mathbf{V}_i$
  - None!
- **Increase the modulus q**
  - Allows to pack more bits per coefficient of $\mathbf{V}_i$
  - Increases the size of $\mathbf{U}$, decreases the LWE error rate
- **Error correcting codes** (we discarded this option)
  - Decreases the decryption failures rate
  - Timing attacks, delicate security analysis [DVV19, GJY19, DTVV19]

BilboKEM

	FrodoKEM-640	BilboKEM-640
Dimension n	640	640
Dimensions $\bar{m} \times \bar{n}$	8×8	8×8
Modulus q	2^{15}	$\mathbf{2^{16}}$
Std. dev. σ	2.8	2.8
Bits dropped in $\mathbf{V}_i$	0 out of 15	**13** out of 16
(Key bits) / coef.	2 out of 15	2 out of 3
Lattice (S/CG/QG)	656/152/142	∞/162/152
Arora-Ge (S/CO/QO)	∞/2798/TBD	∞/4601/TBD
BKW (S/CO/QO)	∞/226/TBD	∞/224/TBD
Decryption failure rate	2^{-138}	2^{-129}
$\lvert$pk$\rvert$ (bytes)	9600	10240
$\lvert\mathbf{U}\rvert$ (bytes)	9600	10240
$\lvert\mathbf{V}_i\rvert$ (bytes)	120	**24**

Ilum

	Kyber-512	Ilum-512-C
Ring degree n	256	256
Module rank ℓ	2	2
Modulus q	3329	3329
Error param. (η_1, η_2)	(3, 2)	(3, 2)
Bits / coef. in $(\mathbf{U}, \mathbf{V}_i)$	(10, 4)	**(11, 3)**
Coef. dropped in $\mathbf{V}_i$	0 out of 256	**128** out of 256
(Key bits) / coef.	1 out of 4	1 out of 3
Lattice (S/CG/QG)	512/151/143	∞/150/142
Arora-Ge (S/CO/QO)	∞/2408/TBD	∞/2227/TBD
BKW (S/CO/QO)	∞/147/TBD	∞/157/TBD
Decryption failure rate	2^{-139}	2^{-125}
$\lvert$pk$\rvert$ (bytes)	768	**768**
$\lvert\mathbf{U}\rvert$ (bytes)	640	**704**
$\lvert\mathbf{V}_i\rvert$ (bytes)	128	**48**



Thank you!

https://eprint.iacr.org/2020/1107

Joël Alwen, Sandro Coretti, Yevgeniy Dodis, and Yiannis Tselekounis.
Security analysis and improvements for the IETF MLS standard for group messaging.
In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part I*, volume 12170 of *LNCS*, pages 248–277. Springer, Heidelberg, August 2020.

Mihir Bellare, Alexandra Boldyreva, and Silvio Micali.
Public-key encryption in a multi-user setting: Security proofs and improvements.
In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

Richard Barnes, Benjamin Beurdouche, Jon Millican, Emad Omara, Katriel Cohn-Gordon, and Raphael Robert.
The Messaging Layer Security (MLS) Protocol.
Internet-Draft draft-ietf-mls-protocol-09, Internet Engineering Task Force, March 2020.
Work in Progress.

Karthikeyan Bhargavan, Richard Barnes, and Eric Rescorla.
TreeKEM: Asynchronous Decentralized Key Management for Large Dynamic Groups A protocol proposal for Messaging Layer Security (MLS).
Research report, Inria Paris, May 2018.

Mihir Bellare, Alexandra Boldyreva, and Jessica Staddon.
Randomness re-use in multi-recipient encryption schemeas.
In Yvo Desmedt, editor, *PKC 2003*, volume 2567 of *LNCS*, pages 85–99. Springer, Heidelberg, January 2003.

Manuel Barbosa and Pooya Farshim.
Efficient identity-based key encapsulation to multiple parties.
In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 428–441. Springer, Heidelberg, December 2005.

Manuel Barbosa and Pooya Farshim.
Randomness reuse: Extensions and improvements.

In Steven D. Galbraith, editor, *11th IMA International Conference on Cryptography and Coding*, volume 4887 of *LNCS*, pages 257–276. Springer, Heidelberg, December 2007.

Alessandro Budroni, Qian Guo, Thomas Johansson, Erik Mårtensson, and Paul Stankovski Wagner.
Making the BKW algorithm practical for LWE.
In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *INDOCRYPT 2020*, volume 12578 of *LNCS*, pages 417–439. Springer, Heidelberg, December 2020.

Olivier Baudron, David Pointcheval, and Jacques Stern.
Extended notions of security for multicast public key cryptosystems.
In Ugo Montanari, José D. P. Rolim, and Emo Welzl, editors, *ICALP 2000*, volume 1853 of *LNCS*, pages 499–511. Springer, Heidelberg, July 2000.

Luca De Feo, David Jao, and Jerome Plût.
Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies.
In *Journal of Mathematical Cryptology*, volume 8 (3), pages 209–247, 2014.

Jan-Pieter D'Anvers, Marcel Tiepelt, Frederik Vercauteren, and Ingrid Verbauwhede.
Timing attacks on error correcting codes in post-quantum schemes.
In *TIS@CCS*, pages 2–9. ACM, 2019.

Jan-Pieter D'Anvers, Frederik Vercauteren, and Ingrid Verbauwhede.
The impact of error dependencies on ring/mod-LWE/LWR based schemes.
In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 103–115. Springer, Heidelberg, 2019.

Qian Guo, Thomas Johansson, and Jing Yang.
A novel CCA attack using decryption errors against LAC.
In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 82–111. Springer, Heidelberg, December 2019.

Dennis Hofheinz and Eike Kiltz.
Secure hybrid encryption from weakened key encapsulation.

In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 553–571. Springer, Heidelberg, August 2007.

Harunaga Hiwatari, Keisuke Tanaka, Tomoyuki Asano, and Koichi Sakumoto.
Multi-recipient public-key encryption from simulators in security proofs.
In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP 09*, volume 5594 of *LNCS*, pages 293–308. Springer, Heidelberg, July 2009.

David Jao and Luca De Feo.
Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies.
In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 19–34. Springer, Heidelberg, November / December 2011.

Kaoru Kurosawa.
Multi-recipient public-key encryption with shortened ciphertext.
In David Naccache and Pascal Paillier, editors, *PKC 2002*, volume 2274 of *LNCS*, pages 48–63. Springer, Heidelberg, February 2002.

Richard Lindner and Chris Peikert.
Better key sizes (and attacks) for LWE-based encryption.
In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, Heidelberg, February 2011.

Takahiro Matsuda and Goichiro Hanaoka.
Key encapsulation mechanisms from extractable hash proof systems, revisited.
In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 332–351. Springer, Heidelberg, February / March 2013.

Emad Omara, Benjamin Beurdouche, Eric Rescorla, Srinivas Inguva, Albert Kwon, and Alan Duric.
The Messaging Layer Security (MLS) Architecture.
Internet-Draft draft-ietf-mls-architecture-04, Internet Engineering Task Force, January 2020.
Work in Progress.

Nigel P. Smart.

Efficient key encapsulation to multiple parties.
In Carlo Blundo and Stelvio Cimato, editors, *SCN 04*, volume 3352 of *LNCS*, pages 208–219. Springer, Heidelberg, September 2005.

Zheng Yang.
On constructing practical multi-recipient key-encapsulation with short ciphertext and public key.
SCN, 8(18):4191–4202, 2015.

Mark Zhandry.
How to record quantum queries, and applications to quantum indifferentiability.
In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 239–268. Springer, Heidelberg, August 2019.