Scalable Ciphertext Compression Techniques for Post-Quantum KEMs and their Applications

Shuichi Katsumata
AIST, JP

Kris Kwiatkowski
PQShield, UK

Federico Pintore
University of Oxford, UK

Thomas Prest
PQShield, UK

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**
- **Terminology:** ciphertext compression, mKEM/mPKE, randomness reuse, etc.
- [BBM00, BPS00, Kur02, BBS03, Sma05, HK07, BF07, HTAS09, MH13, Yan15]
- No* post-quantum proposal

This work

- **Revisiting mKPEs & 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

Revisiting mPKEs & mKEMs

Full reproducibility [BBS03]:

$$\mathsf{Enc}(\mathsf{pk}_1, \mathsf{M}) \xrightarrow{\mathcal{A}(\,\cdot\,, \mathsf{pk}_1, \mathsf{pk}_2, \mathsf{sk}_2, \mathsf{M}')} \mathsf{Enc}(\mathsf{pk}_2, \mathsf{M}')$$

Decomposability (this work):

$$\mathsf{Enc}(\mathsf{pk}_i, \mathsf{M}) = \underbrace{\mathsf{ct}_0}_{\mathsf{Enc}^{\mathsf{ind}}(r_0)} \; \underbrace{\widehat{\mathsf{ct}}_i}_{\mathsf{Enc}^{\mathsf{dep}}(\mathsf{pk}_i, \mathsf{M}, r_0, r_i)}$$

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

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

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

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

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

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

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

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

- 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. $\mathbf{V} \leftarrow \mathbf{RB} + \mathbf{E}'' + \mathsf{Encode}(\mathtt{M})$
4. $\mathtt{ct} := (\mathbf{U}, \mathbf{V})$

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

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

Encompasses many 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 $\mathtt{M}$.

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

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 $\mathtt{pk}$ and $\mathtt{M}$.

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

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. $\mathtt{ct} := (\mathbf{U}, \mathbf{V})$

$\Longrightarrow$

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

1. Sample short matrices $\mathbf{R}, \mathbf{E}'$
2. $\mathbf{U} \leftarrow \mathbf{RA} + \mathbf{E}'$
3. For $i = 1, \ldots, k$:
   1. $\mathbf{E}''_i \leftarrow \chi_5$
   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.

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].

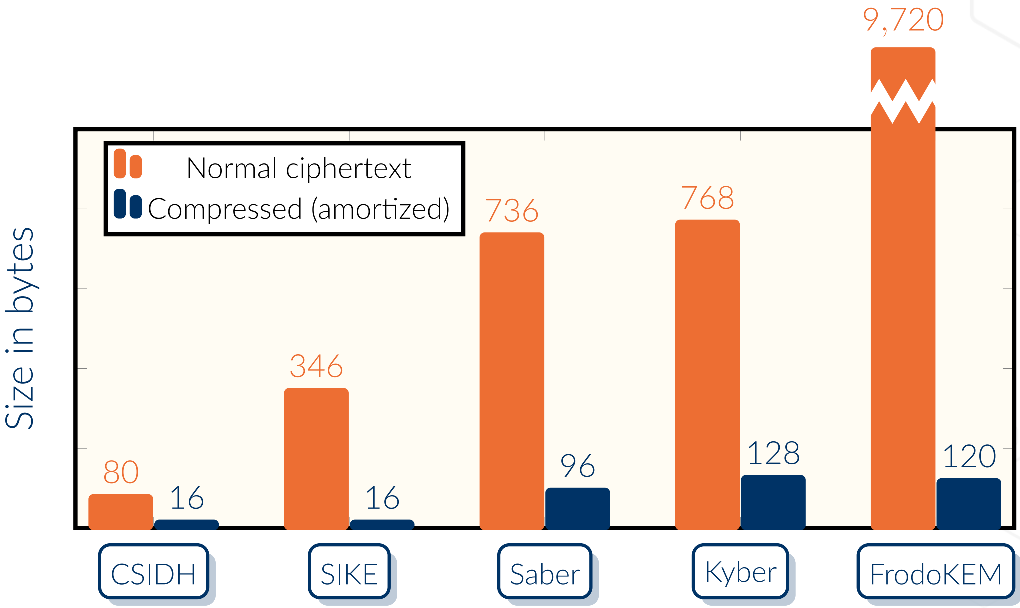
Enc$(\{\mathtt{pk}_1, \ldots, \mathtt{pk}_N\}, \mathtt{M})$

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. $\mathtt{M} = j_i \oplus \widehat{\mathtt{ct}}_i$

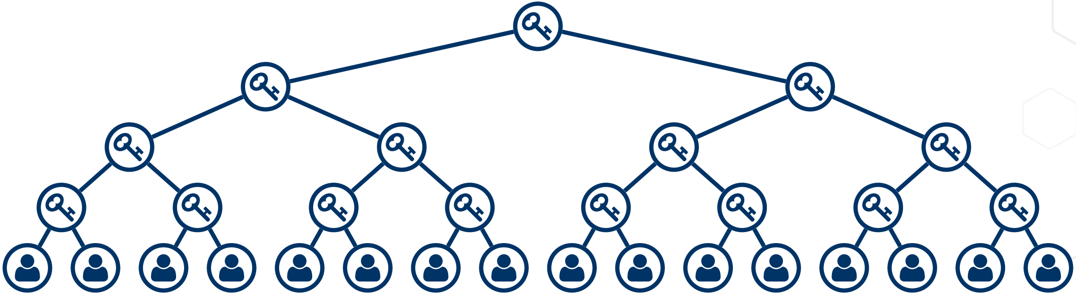
Impact on 1 PKE + 4 KEMs (NIST level I)



Application to TreeKEM

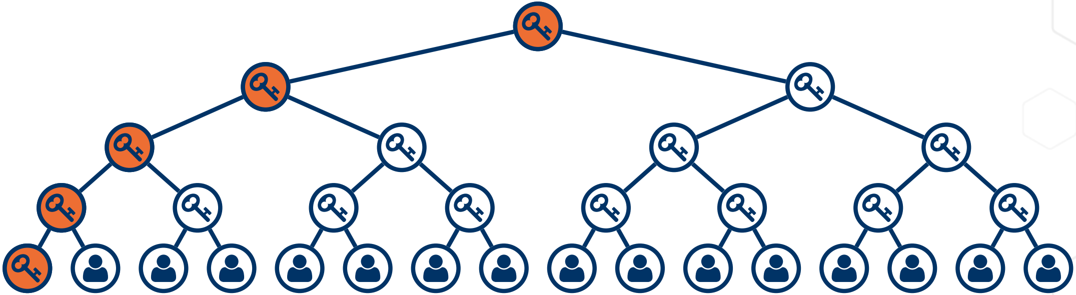
TreeKEM [BBR18, BBM+20, OBR+20, ACDT20]:

- 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.



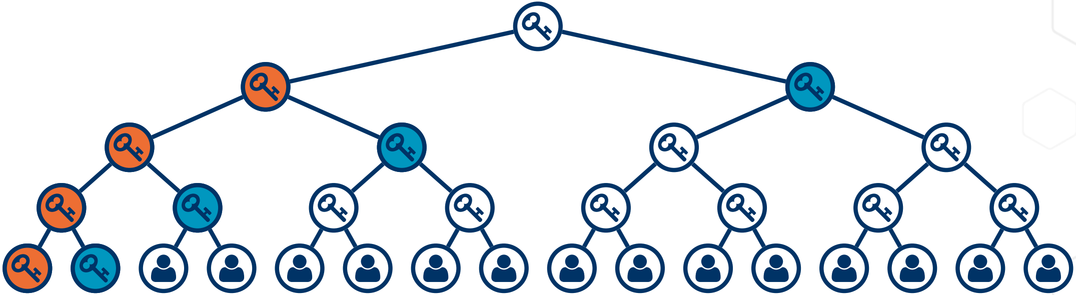
TreeKEM [BBR18, BBM+20, OBR+20, ACDT20]:

- 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.

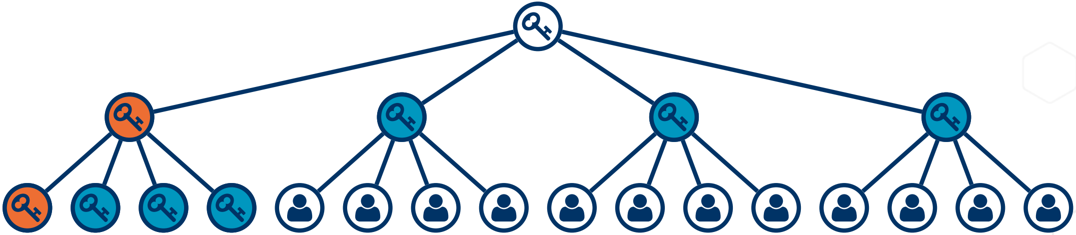


Users that are compromised can refresh their key material by broadcasting an update package that contains:

- 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).

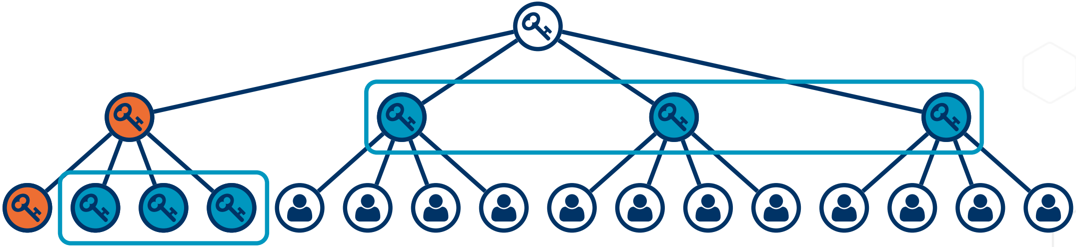
What if we use a m-ary tree instead of a binary tree?

- 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



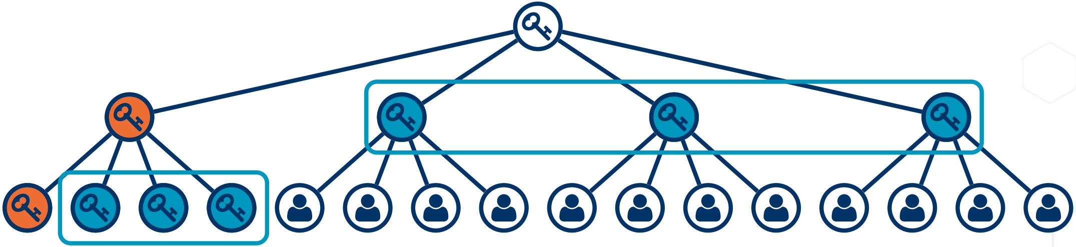
What if we use a m-ary tree instead of a binary tree?

- 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)

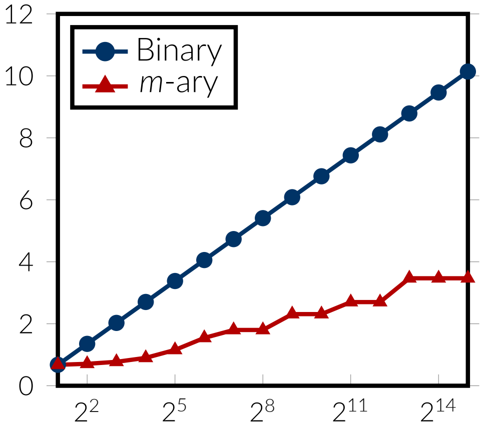


Figure 1: TreeKEM with SIKE

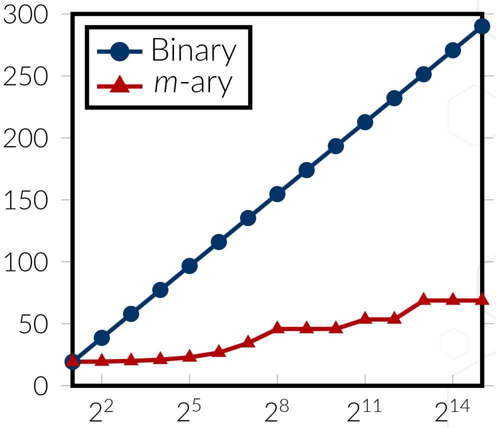
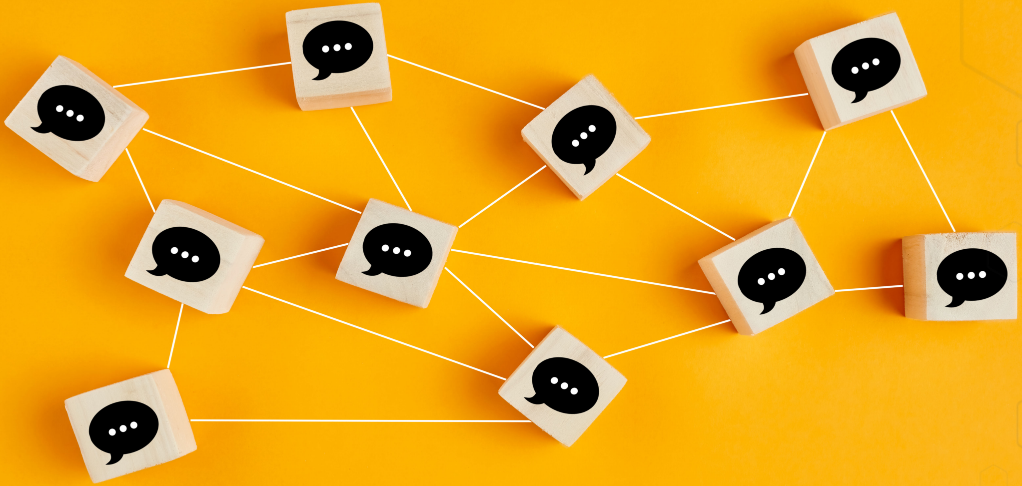


Figure 2: TreeKEM with FrodoKEM

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

Slides: `https://tprest.github.io/pdf/slides/mkem-ac-2020.pdf`



Thank you!

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.
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.

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.

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.