A Concrete Treatment of Efficient Continuous Group Key Agreement via Multi-Recipient PKEs

Thomas Prest

PQShield

Séminaire de Cryptographie de l'Université de Rennes 1

This is joint work with:

- Keitaro Hashimoto (Tokyo Institute of Technology et AIST, Japon)
- Shuichi Katsumata (AIST, Japon)
- Eamonn W. Postlethwaite (Royal Holloway, UK)
- Bas Westerbaan (PQShield Ltd., UK)

Full paper: `https://eprint.iacr.org/2021/1407`

Why Secure Messaging?

Why secure messaging?

Secure messaging applications are widespread...

WhatsApp	2.5 Billions
Facebook Messenger	1.3 Billions
Telegram	500 Millions
Snapchat	280 Millions

...and represent attractive targets for attackers:

- "*Al Jazeera journalists 'hacked via NSO Group spyware'*", BBC, 2020
  `https://www.bbc.com/news/technology-55396843`
- "*Grand jury subpoena for Signal user data, Central District of California*", Signal , 2020
  `https://signal.org/bigbrother/central-california-grand-jury/`
- etc.

Why secure messaging?

Secure messaging applications are widespread...

WhatsApp	2.5 Billions
Facebook Messenger	1.3 Billions
Telegram	500 Millions
Snapchat	280 Millions

...and represent attractive targets for attackers:

- "*Al Jazeera journalists 'hacked via NSO Group spyware*'", BBC, 2020
  https://www.bbc.com/news/technology-55396843
- "*Grand jury subpoena for Signal user data, Central District of California*", Signal , 2020
  https://signal.org/bigbrother/central-california-grand-jury/
- etc.

This talk: the security of secure *group* messaging *protocols*

Constraints

Specific constraints:

→ Asynchrony

Constraints

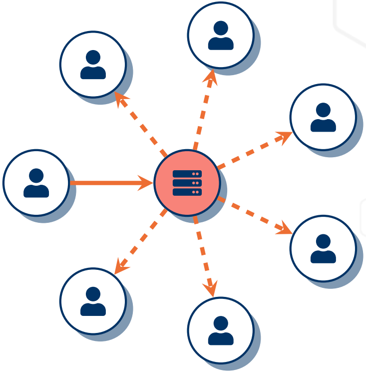
Specific constraints:

- Asynchrony
- Minimise trust in the server
  ($\Longrightarrow$ end-to-end encryption)

Constraints

Specific constraints:

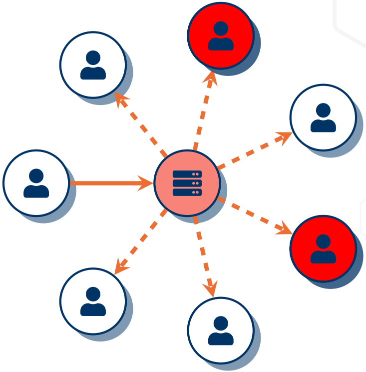
- Asynchrony
- Minimise trust in the server ($\Longrightarrow$ end-to-end encryption)
- Large number of users ($N \gg 1$)



Constraints

Specific constraints:

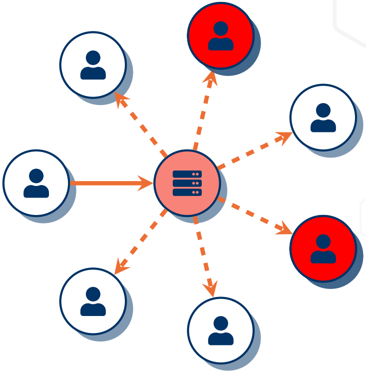
- Asynchrony
- Minimise trust in the server ($\Longrightarrow$ end-to-end encryption)
- Large number of users ($N \gg 1$)
- Very long sessions ($t \gg 1$)



Constraints

Specific constraints:

- Asynchrony
- Minimise trust in the server ($\implies$ end-to-end encryption)
- Large number of users ($N \gg 1$)
- Very long sessions ($t \gg 1$)

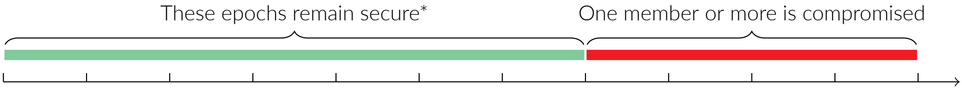


If each user has a probability ϵ of being compromised by an attacker during a unit of time, a group conversation with N members over t units of time will be compromised with probability $1 - (1 - \epsilon)^{Nt}$.

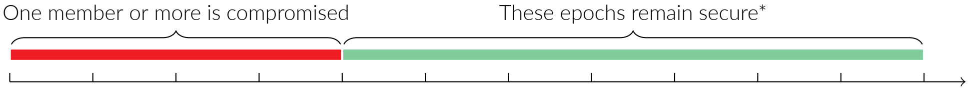
- This probability is significant as soon as $Nt = \Omega(1/\epsilon)$.
- **Solution:** periodically refresh encryption keys (next slides).

Some security notions

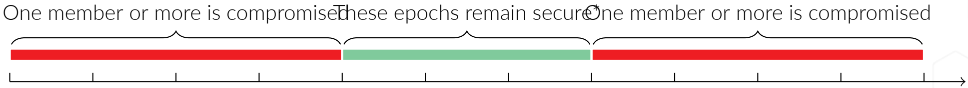
Forward secrecy (FS) [CCG16, CGCD^{+}17, ACD19]:



Post-Compromise Security (PCS) [CCG16, CGCD^{+}17, ACD19]:



Post-Compromise Forward Security (PCFS) [ACDT20, ACJM20, AJM20]:



CGKAs

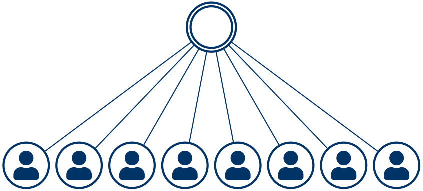
CGKAs (*Continuous Group Key Agreement*) concentrate the cryptographic mechanisms of secure gorup messaging protocols:

- → Add a user
- → Remove a user
- → **Remove one's encryption keypair (*Ratcheting/Commit Message*)**

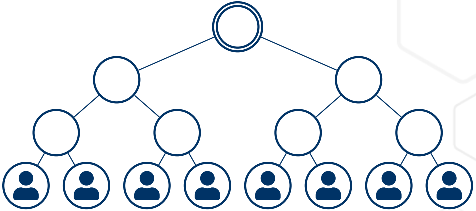
Prominent CGKAs:

- → *Pairwise Channels* (Signal)
- → *Sender Keys* (WhatsApp)
- → *TreeKEM* [BBR18, Wei19, BBN19, ACDT20, AJM20, ACJM20, ACC$^+$21] (IETF MLS draft standard [OBR$^+$21, BBM$^+$20])
- → *Chained mKEM* [BBN19]

Chained mKEM



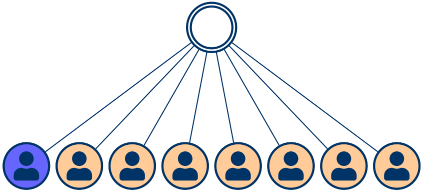
TreeKEM



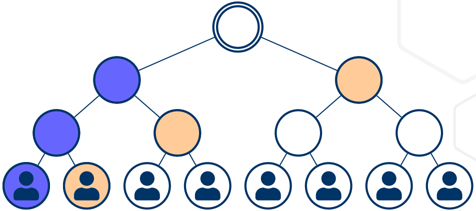
In Chained mKEM and TreeKEM:

- → To each node (○,◎) is associated an encryption keypair
- → A user knows the decryption key of a node if and only if this node is in their path (i.e. the node is an ancestor of the user's node)

Chained mKEM



TreeKEM

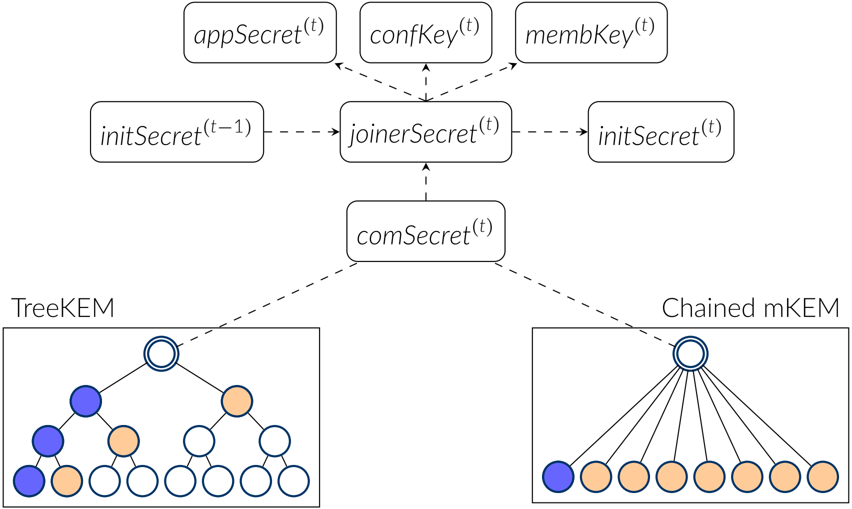


In Chained mKEM and TreeKEM:

- → To each node (◯,◎) is associated an encryption keypair
- → A user knows the decryption key of a node if and only if this node is in their path (i.e. the node is an ancestor of the user's node)
- → A commit message (here, sent by the leftmost user) contains:
  - › an encryption key for each
  - › an asymmetric ciphertext for each
  - › two signatures (one that authenticates encryption keys, one that authenticates ciphertext)

Therefore it has a size $O(N)$ for Chained mKEM, and $\Omega(\log N)$ for TreeKEM.

A change of epoch $(t-1) \longrightarrow t$



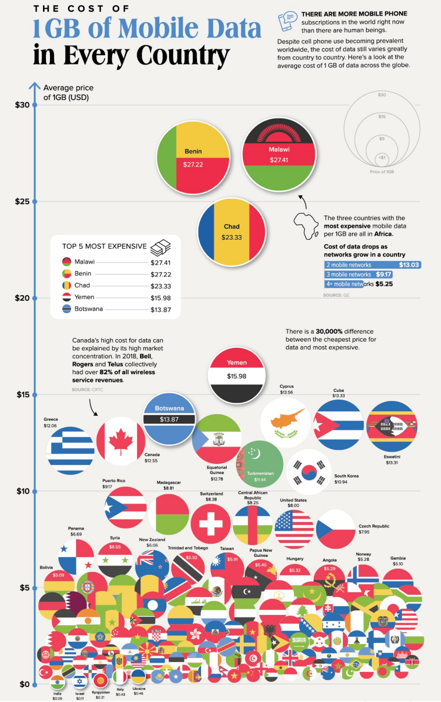
The larger a group size N is, the more *commit messages*:

- are necessary
- are costly

This tension is amplified by:

- the cost and impact for end users
- post-quantum cryptography ($\times 10$ or more compared to classical cryptography). Example with:
  - TreeKEM
  - Classic McEliece [ABC+20]
  - 256 users

  If each user sends one *commit message*, the bandwidth cost if 512 MiB per user.



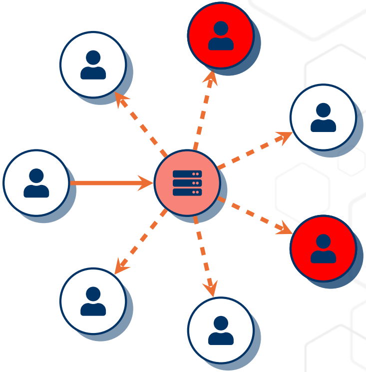
([1])

[1] https://www.visualcapitalist.com/cost-of-mobile-data-worldwide/

Our solution

Chained CmPKE, un CGKA avec un coût asymétrique en envoi et réception (de *commit message*):

	Upload	Download	Total
Chained CmPKE	$O(N)$	$O(1)$	$O(N)$
TreeKEM	$\Omega(\log N)$	$\Omega(\log N)$	$\Omega(N \log N)$



Conceptually, the main change of Chained CmPKE is to make the server more active (but without entrusting it more).

Technically, Chained CmPKE is based on Chained mKEM, with two new ideas:

1. **Use very efficient mPKEs (*multi-recipient PKE*)** $\Rightarrow$ reduce upload costs
2. **The use of CmPKE (*committing mPKE*)** $\Rightarrow$ a cost $O(1)$

Ingredient №1: *Committing* mPKEs

Committing mPKEs

Syntax of a mPKE (*multi-recipient PKE*):

- $\mathsf{mEnc}(\mathtt{M}, (\mathtt{ek}_i)_{i \in [N]}) \to (\mathtt{ct}_0, (\widehat{\mathtt{ct}}_i)_{i \in [N]})$
- $\mathsf{mDec}(\mathtt{dk}_i, (\mathtt{ct}_0, \widehat{\mathtt{ct}}_i)) \to \{\mathtt{M} \text{ or } \perp\}$

Recently revisited in [KKPP20], which inspired this work.

The syntax of a CmPKE (*committing mPKE*) is identical:

- $\mathsf{CmEnc}(\mathtt{M}, (\mathtt{ek}_i)_{i \in [N]}) \to (\mathsf{T}, (\mathtt{ct}_i)_{i \in [N]})$
- $\mathsf{CmDec}(\mathtt{dk}_i, (\mathsf{T}, \mathtt{ct}_i)) \to \{\mathtt{M} \text{ or } \perp\}$

In addition, we require that T is *committing*, i.e. T is bound to a unique message $\mathtt{M}$. A related notion: *committing AEADs* [GLR17]

We provide an (mPKE IND-CPA $\Rightarrow$ CmPKE IND-CCA) transform, with:

- $\mathtt{ct}_i = \widehat{\mathtt{ct}}_i$.
- $\mathsf{T} = (\mathtt{ct}_0, c)$ and $|c| = 32$ bytes.

Our transform uses *key-committing AEADs* [FOR17, GLR17, ADG+20].

Impact of CmPKEs

In Chained mKEM, a commit message contains:

1. A new encryption key $\mathtt{ek}$
2. An mPKE ciphertext: $(\mathtt{ct}_0, (\widehat{\mathtt{ct}}_i)_{i \in [N]})$
3. A signature $\mathtt{sig}_1 \leftarrow \mathsf{Sign}(\mathtt{sk}, \mathtt{ek})$
4. A signature $\mathtt{sig}_2 \leftarrow \mathsf{Sign}(\mathtt{sk}, (\mathtt{ct}_0, (\widehat{\mathtt{ct}}_i)_{i \in [N]}))$

In Chained CmPKE, a commit message contains:

Upload:

1. $\mathtt{ek}$
2. A CmPKE ciphertext: $(\mathsf{T}, (\mathtt{ct}_i)_{i \in [N]})$
3. $\mathtt{sig}_1 \leftarrow \mathsf{Sign}(\mathtt{sk}, \mathtt{ek})$
4. $\mathtt{sig}_2 \leftarrow \mathsf{Sign}(\mathtt{sk}, \mathsf{T})$

Download:

1. $\mathtt{ek}$
2. $(\mathsf{T}, \mathtt{ct}_i)$
3. $\mathtt{sig}_1$
4. $\mathtt{sig}_2$

Intuitively, any attempt from the server to tamper with T or $\mathtt{ct}_i$ is detected upon signature verification or during decryption.

Ingredient №2: more efficient mPKEs

Post-quantum mPKEs

[KKPP20] highlighted the existence of very efficient mPKEs based on LWE, LWR and SIDH. Exemple with LPR-style schemes [LPR10, LP11]:

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

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

Post-quantum mPKEs

[KKPP20] highlighted the existence of very efficient mPKEs based on LWE, LWR and SIDH. Exemple with LPR-style schemes [LPR10, LP11]:

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

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

$\Longrightarrow$

$\mathsf{mEnc}(\{\mathbf{ek}_1, \ldots, \mathbf{ek}_N\}, \mathtt{M})$

1. Sample short matrices $\mathbf{R}, \mathbf{E}'$
2. $\mathbf{U} \leftarrow \mathbf{R}\mathbf{A} + \mathbf{E}'$
3. For $i = 1, \ldots, N$:
   1. Sample a short matrix $\mathbf{E}''_i$
   2. $\mathbf{V}_i \leftarrow \mathbf{R}\mathbf{B}_i + \mathbf{E}''_i + \mathsf{Encode}(\mathtt{M})$
4. $\left(\mathtt{ct}_0, (\widehat{\mathtt{ct}}_i)_{i \in [N]}\right) := (\mathbf{U}, (\mathbf{V}_i)_{i \in [N]})$

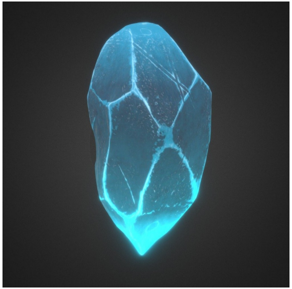
Limitations of [KKPP20]:

- Naive parametrisations
- No concrete security analysis

We propose three re-parametrisations of lattice-based (m)PKEs:

BilboKEM640 (inspired form FrodoKEM640)

Ilum512 (inspired from Kyber512)



LPR757 (inspired from NTRU LPR653)

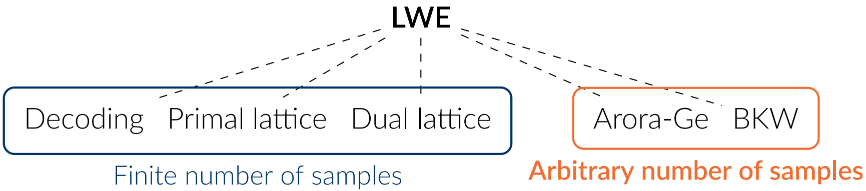
These mPKEs are tailored for the Chained CmPKE protocol:

- → The $(\widehat{\mathtt{ct}}_i)_{i \in [N]}$ are extremely small
- → This entails a small increase of the sizes of $\mathtt{ek}$ and $\mathtt{ct}_0$

Recall that $\mathtt{M} = \mathrm{Decode}(\mathbf{V}_i - \mathsf{dk}_i \cdot \mathbf{U})$. Our toolkit:

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

The main attacks to (re-)consider:



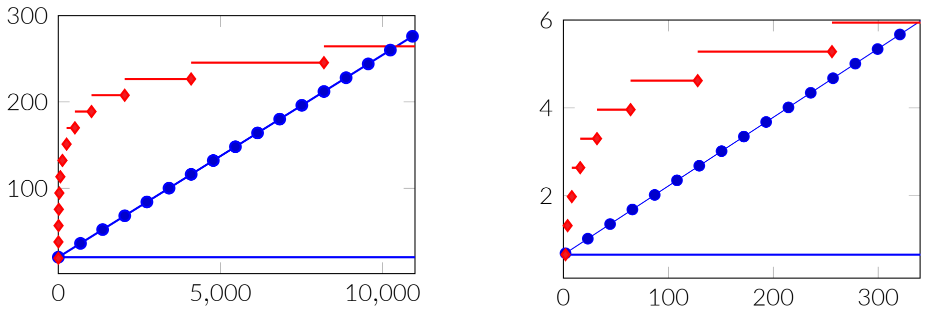
Communication costs of mPKEs based on existing (gray background) and new (fond blanc) parametrisations. Sécurité: NIST I ($\geq$ AES-128).

Schéma	$\lvert\mathtt{ek}\rvert$	$\lvert\mathtt{ct}_0\rvert$	$\lvert\widehat{\mathbf{ct}}_i\rvert$
Kyber512 [SAB+20]	768 (+32)	640	**128**
Ilum512	768	704	**48**
LPRime653 [BBC+20]	865 (+32)	865 (+32)	**128**
LPRime757	1076	1076	**32**
Frodo640 [NAB+20]	9600 (+16)	9600	**120**
Bilbo640	10240	10240	**24**
SIKEp434 [JAC+20]	330	330	**16**

Comparisons & Conclusion

Chained CmPKE vs TreeKEM (upload and download)

Size of a commit message in KiB as a function of the group size.



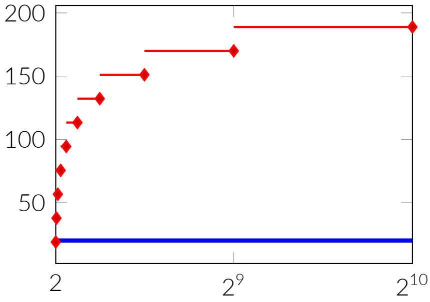
(a) Bilbo640 vs. Frodo640

(b) SIKEp434 vs. SIKEp434

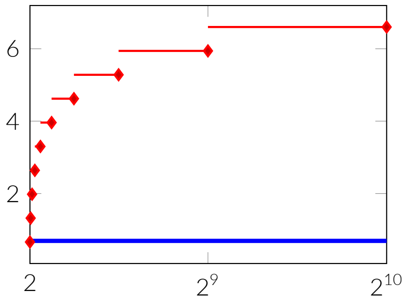
- Chained CmPKE (**upload**) with {Bilbo640, SIKEp434}
- Chained CmPKE (**download**) with {Bilbo640, SIKEp434}
- TreeKEM (**upload and download**) with {Frodo640, SIKEp434}

Chained CmPKE vs TreeKEM (total normalised by N)

Normalised cost of a commit message in KiB as a function of the group size.



(a) Bilbo640 vs. Frodo640

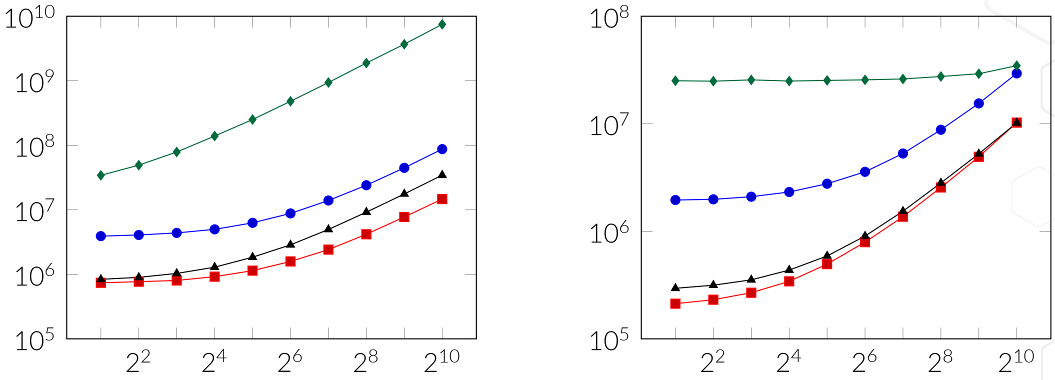


(b) SIKEp434 vs. SIKEp434

Chained CmPKE with {Bilbo640, SIKEp434}

TreeKEM with {Frodo640, SIKEp434}

Computational cost of a commit message



(a) Commit (sender side)   (b) Process (receiver side)

Running time of some procedures as a function of the group size, for Ilum512 (■), LPRime757 (▲), Bilbo640 (●), SIKEp434 (◆). Logarithmic scales. Timings obtained on an Apple M1 @3.2 GHz.

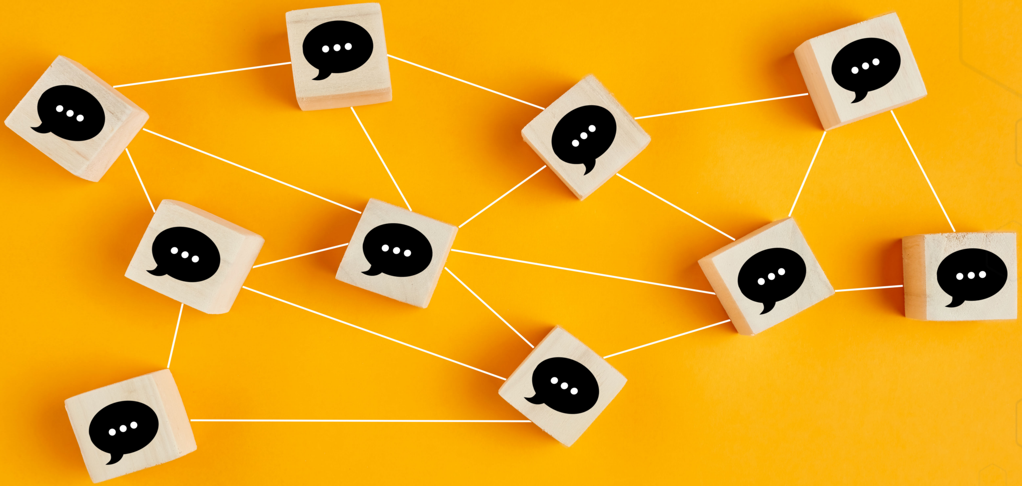
Conclusion

We proposed *Chained CmPKE*, a CGKA that is:

- Very fast
- More compact than TreeKEM: $O(N)$ instead of $\Omega(N \log N)$
- Simpler than TreeKEM
- Satisfying the same security notions (see paper)

As well as techniques that might be of independent interest:

1. *Committing mPKEs*
2. More efficient lattice-based mPKEs



Questions?

Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, and Wen Wang.
Classic McEliece.
Technical report, National Institute of Standards and Technology, 2020.
available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

Joël Alwen, Margarita Capretto, Miguel Cueto, Chethan Kamath, Karen Klein, Ilia Markov, Guillermo Pascual-Perez, Krzysztof Pietrzak, Michael Walter, and Michelle Yeo.
Keep the dirt: Tainted treekem, adaptively and actively secure continuous group key agreement.
In *2021 IEEE Symposium on Security and Privacy (S&P)*, pages 596–612, Los Alamitos, CA, USA, may 2021. IEEE Computer Society.

Joël Alwen, Sandro Coretti, and Yevgeniy Dodis.
The double ratchet: Security notions, proofs, and modularization for the Signal protocol.
In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 129–158. Springer, Heidelberg, May 2019.

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.

Joël Alwen, Sandro Coretti, Daniel Jost, and Marta Mularczyk.
Continuous group key agreement with active security.
In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part II*, volume 12551 of *LNCS*, pages 261–290. Springer, Heidelberg, November 2020.

Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg.
How to abuse and fix authenticated encryption without key commitment.
Cryptology ePrint Archive, Report 2020/1456, 2020.
`https://eprint.iacr.org/2020/1456`.

Joël Alwen, Daniel Jost, and Marta Mularczyk.
On the insider security of MLS.
Cryptology ePrint Archive, Report 2020/1327, 2020.
`https://eprint.iacr.org/2020/1327`.

Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang.
NTRU Prime.
Technical report, National Institute of Standards and Technology, 2020.
available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

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-11, Internet Engineering Task Force, December 2020.
Work in Progress.

Karthikeyan Bhargavan, Benjamin Beurdouche, and Prasad Naldurg.
Formal Models and Verified Protocols for Group Messaging: Attacks and Proofs for IETF MLS.
Research report, Inria Paris, December 2019.

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.

Katriel Cohn-Gordon, Cas J. F. Cremers, and Luke Garratt.
On post-compromise security.
In Michael Hicks and Boris Köpf, editors, *CSF 2016 Computer Security Foundations Symposium*, pages 164–178. IEEE Computer Society Press, 2016.

Katriel Cohn-Gordon, Cas Cremers, Benjamin Dowling, Luke Garratt, and Douglas Stebila.
A formal security analysis of the signal messaging protocol.
In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 451–466, 2017.

Jan-Pieter D'Anvers, Marcel Tiepelt, Frederik Vercauteren, and Ingrid Verbauwhede.
Timing attacks on error correcting codes in post-quantum schemes.
In Begül Bilgin, Svetla Petkova-Nikova, and Vincent Rijmen, editors, *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.

Pooya Farshim, Claudio Orlandi, and Razvan Rosie.
Security of symmetric primitives under incorrect usage of keys.
IACR Transactions on Symmetric Cryptology, pages 449–473, 2017.

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.

Paul Grubbs, Jiahui Lu, and Thomas Ristenpart.
Message franking via committing authenticated encryption.
In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 66–97. Springer, Heidelberg, August 2017.

David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Amir Jalali, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Joost Renes, Vladimir Soukharev, David Urbanik, Geovandro Pereira, Koray Karabina, and Aaron Hutchinson.
SIKE.
Technical report, National Institute of Standards and Technology, 2020.

available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

Shuichi Katsumata, Kris Kwiatkowski, Federico Pintore, and Thomas Prest.
Scalable ciphertext compression techniques for post-quantum KEMs and their applications.
In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020*, *Part I*, volume 12491 of *LNCS*, pages 289–320. Springer, Heidelberg, December 2020.

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.

Vadim Lyubashevsky, Chris Peikert, and Oded Regev.
On ideal lattices and learning with errors over rings.
In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010.

Michael Naehrig, Erdem Alkim, Joppe Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila.
FrodoKEM.
Technical report, National Institute of Standards and Technology, 2020.
available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

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-06, Internet Engineering Task Force, March 2021.
Work in Progress.

Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé.
CRYSTALS-KYBER.
Technical report, National Institute of Standards and Technology, 2020.
available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

Matthew Weidner.
Group messaging for secure asynchronous collaboration.
Mphil dissertation, University of Cambridge, Cambridge, UK, 2019.