La sécurité des protocoles de messagerie instantanée de groupe

Thomas Prest

PQShield

Les résultats présentés ici sont le fruit d'une collaboration avec:

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

Pourquoi les messageries sécurisées?

Pourquoi les messageries sécurisées?

Les messageries sécurisées sont omniprésentes…

WhatsApp	2.5 milliards
Facebook Messenger	1.3 milliards
Telegram	500 millions
Snapchat	280 millions

…et constituent des vecteurs d'attaques privilégiés:

- "*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.

Pourquoi les messageries sécurisées?

Les messageries sécurisées sont omniprésentes...

WhatsApp	2.5 milliards
Facebook Messenger	1.3 milliards
Telegram	500 millions
Snapchat	280 millions

...et constituent des vecteurs d'attaques privilégiés:

- "*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.

Cet exposé: la sécurité des *protocoles* de messagerie sécurisée *de groupe*

Contraintes

Des contraintes spécifiques:

→ Asynchronie

Contraintes

Des contraintes spécifiques:

- Asynchronie
- Minimiser la confiance dans le serveur
  ($\Longrightarrow$ chiffrement de bout-en-bout)

Contraintes

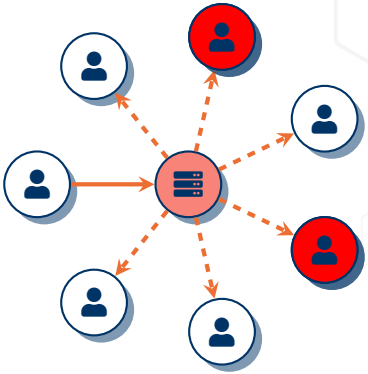
Des contraintes spécifiques:

- Asynchronie
- Minimiser la confiance dans le serveur ($\Longrightarrow$ chiffrement de bout-en-bout)
- Grand nombre d'utilisateurs ($N \gg 1$)

Contraintes

Des contraintes spécifiques:

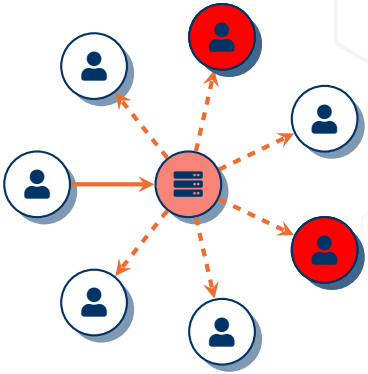
- Asynchronie
- Minimiser la confiance dans le serveur ($\Longrightarrow$ chiffrement de bout-en-bout)
- Grand nombre d'utilisateurs ($N \gg 1$)
- Sessions très longues ($t \gg 1$)



Contraintes

Des contraintes spécifiques:

- Asynchronie
- Minimiser la confiance dans le serveur ($\Longrightarrow$ chiffrement de bout-en-bout)
- Grand nombre d'utilisateurs ($N \gg 1$)
- Sessions très longues ($t \gg 1$)

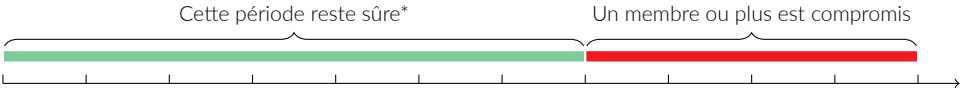


Si chaque utilisateur a une probabilité ϵ d'être compromis pendant une unité de temps, une conversation avec N membres sur une durée t se voit compromise avec probabilité $1 - (1 - \epsilon)^{Nt}$.

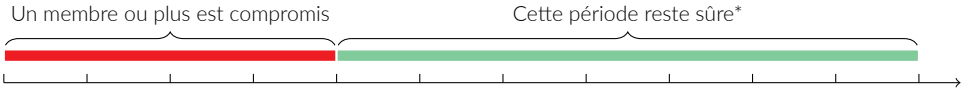
- Cette probabilité est significative dès que $Nt = \Omega(1/\epsilon)$.
- **Solution:** rafraîchir périodiquement les clefs de chiffrement (slides suivants).

Quelques notions de sécurité

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



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



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

Les CGKAs

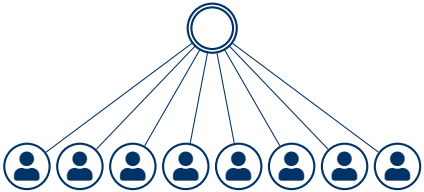
Les CGKAs (*Continuous Group Key Agreement*) concentrent les mécanismes cryptographiques des protocoles de messagerie sécurisée de groupe:

- Ajouter un utilisateur
- Enlever un utilisateur
- **Rafraîchir ses clefs de chiffrement (*Ratcheting/Commit Message*)**

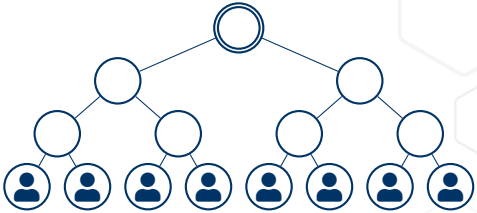
CGKAs éminents:

- *Pairwise Channels*, le CGKA de Signal
- *Sender Keys*, le CGKA de WhatsApp
- *TreeKEM* [BBR18, Wei19, BBN19, ACDT20, AJM20, ACJM20, ACC+21], le CGKA du draft IETF MLS [OBR+21, BBM+20]
- *Chained mKEM* [BBN19]

Chained mKEM



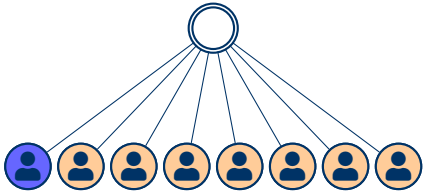
TreeKEM



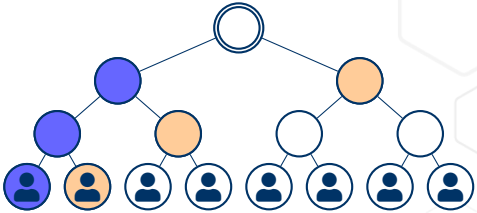
Dans Chained mKEM et TreeKEM:

- A chaque noeud (◯,◎) est associé une bi-clef de chiffrement
- Un utilisateur connaît la clef de déchiffrement d'un nœud si et seulement si ce nœud est dans son chemin (= il est un ancêtre)

Chained mKEM



TreeKEM

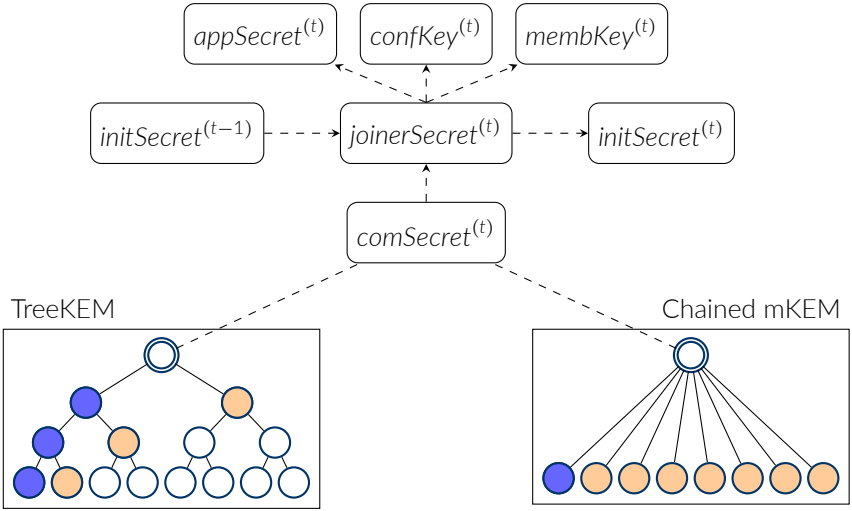


Dans Chained mKEM et TreeKEM:

- A chaque noeud (◯,◎) est associé une bi-clef de chiffrement
- Un utilisateur connaît la clef de déchiffrement d'un nœud si et seulement si ce nœud est dans son chemin (= il est un ancêtre)
- Un commit message (ici envoyé par l'utilisateur le plus à gauche) contient:
  - une clef publique pour chaque ●
  - un chiffré (asymétrique) pour chaque ◯
  - deux signatures (une pour les clefs publiques, une pour les chiffrés)

Il a donc une taille $O(N)$ pour Chained mKEM, $\Omega(\log N)$ pour TreeKEM.

Un changement d'époque $(t-1) \longrightarrow t$



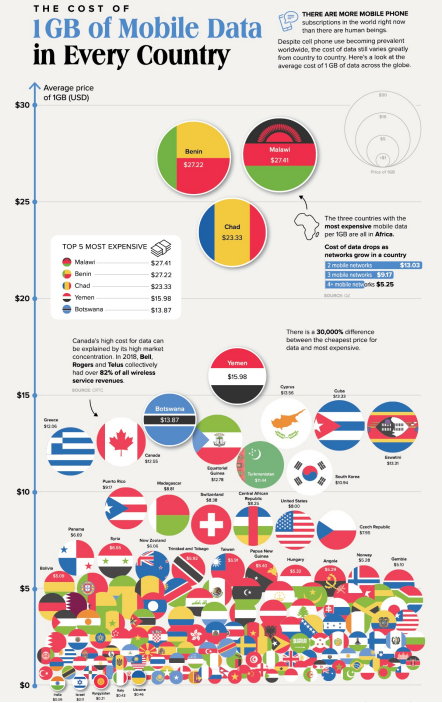
Plus la taille N d'un groupe est élevée et plus les *commit messages*:

- → sont nécessaires
- → sont coûteux

Cette tension est amplifiée par:

- → le coût et l'impact pour l'utilisateur
- → la cryptographie post-quantique ($\times 10$ ou plus par rapport à la cryptographie classique). Exemple si on prend:
  - TreeKEM
  - Classic McEliece [ABC+20]
  - 256 utilisateurs

  Si chacun envoie 2 *commit messages*, cela coûte 1 Go pour chaque utilisateur.



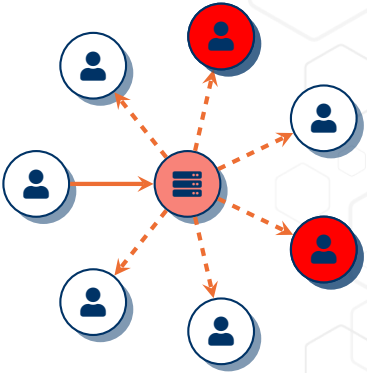
(1)

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

Notre solution

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

	Envoi	Réception	Total
Chained CmPKE	$O(N)$	$O(1)$	$O(N)$
TreeKEM	$\Omega(\log N)$	$\Omega(\log N)$	$\Omega(N \log N)$



Conceptuellement, la nouveauté de Chained CmPKE est de rendre le serveur actif (mais sans lui accorder plus de confiance).

Techniquement, Chained CmPKE est basé sur Chained mKEM, avec deux nouvelles idées:

1. **Utiliser des mPKEs (*multi-recipient PKE*) très efficaces** $\Rightarrow$ réduit les coûts d'envoi
2. **La notion de CmPKE (*committing mPKE*)** $\Rightarrow$ un coût en réception $O(1)$

Ingrédient №1: les *Committing* mPKEs

Les Committing mPKEs

Syntaxe d'un mPKE (*multi-recipient PKE*):

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

Récemment revisités dans [KKPP20], dont on s'inspire pour ce travail.

La syntaxe d'un CmPKE (*committing mPKE*) est identique:

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

De plus, on demande que T soit *engageant*, c'est-à-dire que T est lié à un unique message $\mathtt{M}$. Une notion proche: les *committing AEADs* [GLR17]

Nous fournissons une transformation (mPKE IND-CPA $\Rightarrow$ CmPKE IND-CCA), avec:

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

Nous utilisons les *key-committing AEADs* [FOR17, GLR17, ADG+20].

Impact des CmPKEs

Dans Chained mKEM, un commit message contient:

1. Une nouvelle clef de chiffrement $\mathtt{ek}$
2. Un chiffré mPKE: $(\mathtt{ct}_0, (\widehat{\mathtt{ct}}_i)_{i\in[N]})$
3. Une signature $\mathtt{sig}_1 \leftarrow \mathrm{Sign}(\mathtt{sk}, \mathtt{ek})$
4. Une signature $\mathtt{sig}_2 \leftarrow \mathrm{Sign}(\mathtt{sk}, (\mathtt{ct}_0, (\widehat{\mathtt{ct}}_i)_{i\in[N]}))$

Dans Chained CmPKE, un commit message contient:

En envoi:

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

En réception:

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

Intuitivement, toute tentative du serveur d'altérer T ou $\mathtt{ct}_i$ sera détectée à la vérification de la signature ou au déchiffrement.

Ingrédient №2: des mPKEs plus efficaces

Des mPKEs post-quantiques

[KKPP20] a mis en évidence l'existence de mPKE très efficaces basés sur LWE, LWR et SIDH. Exemple avec les schémas à la [LPR10, LP11]:

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

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

Des mPKEs post-quantiques

[KKPP20] a mis en évidence l'existence de mPKE très efficaces basés sur LWE, LWR et SIDH. Exemple avec les schémas à la [LPR10, LP11]:

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

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

Limitations [KKPP20]:

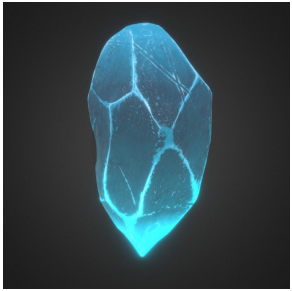
- → Paramétrisations naïves
- → Pas d'analyse de la sécurité concrète

Nouveaux mPKEs à base de réseaux

Nous proposons trois re-paramétrisations de (m)PKEs à base de réseaux:

BilboKEM640 (inspiré de FrodoKEM640)

Ilum512 (inspiré de Kyber512)



LPR757 (inspiré de NTRU LPR653)

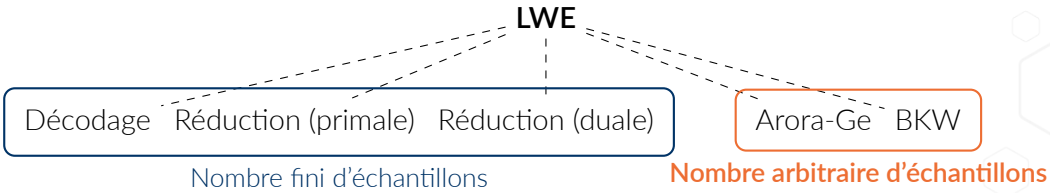
Ces mPKEs sont taillés sur mesure pour le protocole Chained CmPKE:

- → Les $(\widehat{\mathtt{ct}}_i)_{i\in[N]}$ sont extrêmement petits
- → Cela vient au prix d'une faible augmentation des tailles de $\mathtt{ek}$ et $\mathtt{ct}_0$

On rappelle que $\mathtt{M} = \mathrm{Decode}(\mathbf{V}_i - \mathtt{dk}_i \cdot \mathbf{U})$. Notre boîte à outils:

- **Bit dropping:** couper les bits de poids faible de $\mathbf{V}_i$
  - Diminue la taille de $\mathbf{V}_i$, augmente le taux d'erreur LWE
  - Augmente le taux d'échec au déchiffrement (*decryption failure rate*)
- **Coefficent dropping:** couper les coefficients superflus de $\mathbf{V}$
  - Diminue la taille de $\mathbf{V}_i$
  - Aucun!
- **Augmenter le module** q
  - Permet de compacter plus de bits de clef par coefficient de $\mathbf{V}_i$
  - Augmente la taille de $\mathbf{U}$, diminue le taux d'erreur LWE
- **Codes correcteurs d'erreur** (option non retenue)
  - Diminue le taux d'échec au déchiffrement
  - Attaques par timing, analyse de sécurité délicate [DVV19, GJY19, DTVV19]

Les principales attaques à (re-)considérer:



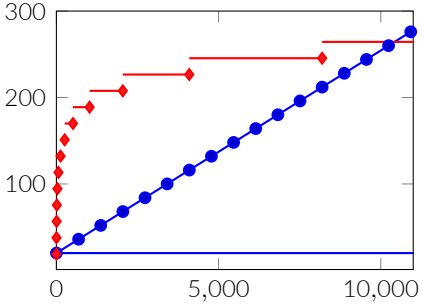
Coûts en communication de mPKEs basés sur des paramétrisations existantes (fond gris) et nouvelles (fond blanc). Sécurité: NIST I ($\geq$ AES-128).

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**

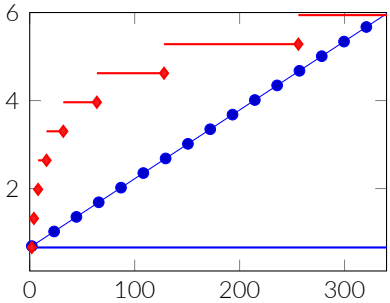
Comparaisons & Conclusion

Chained CmPKE vs TreeKEM (envoi et réception)

Taille d'un *commit message* en kilooctets en fonction du nombre d'utilisateurs.



(a) Bilbo640 vs. Frodo640

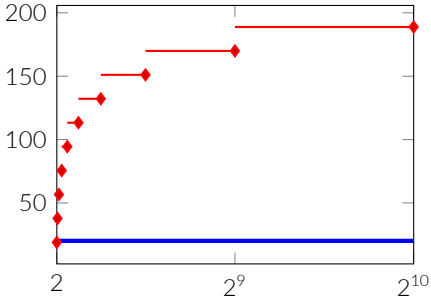


(b) SIKEp434 vs. SIKEp434

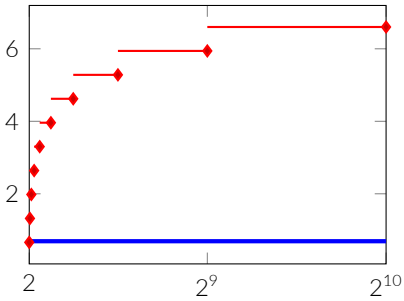
- Chained CmPKE (**envoi**) avec {Bilbo640, SIKEp434}
- Chained CmPKE (**réception**) avec {Bilbo640, SIKEp434}
- TreeKEM (**envoi et réception**) avec {Frodo640, SIKEp434}

Chained CmPKE vs TreeKEM (total normalisé par N)

Coût normalisé d'un *commit message* en kilooctets en fonction du nombre d'utilisateurs.



(a) Bilbo640 vs. Frodo640

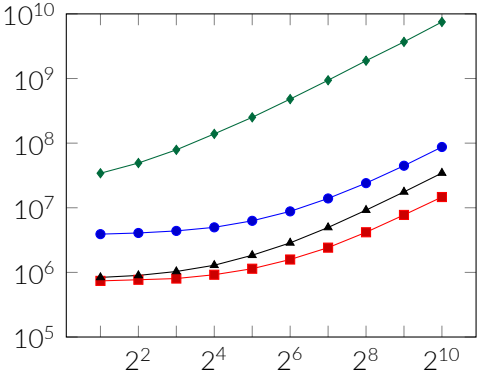


(b) SIKEp434 vs. SIKEp434

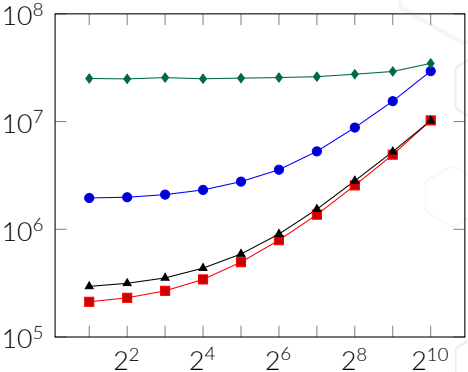
Chained CmPKE avec {Bilbo640, SIKEp434}

TreeKEM avec {Frodo640, SIKEp434}

Coût calculatoire d'un *commit message*



(a) Commit (envoi)



(b) Process (réception)

Temps d'exécution en nanosecondes de quelques procédures en fonction de la taille du groupe, pour Ilum512 (—■—), LPRime757 (—▲—), Bilbo640 (—●—), SIKEp434 (—◆—). Echelles logarithmiques. Temps obtenus sur un Apple M1 @3.2 GHz.

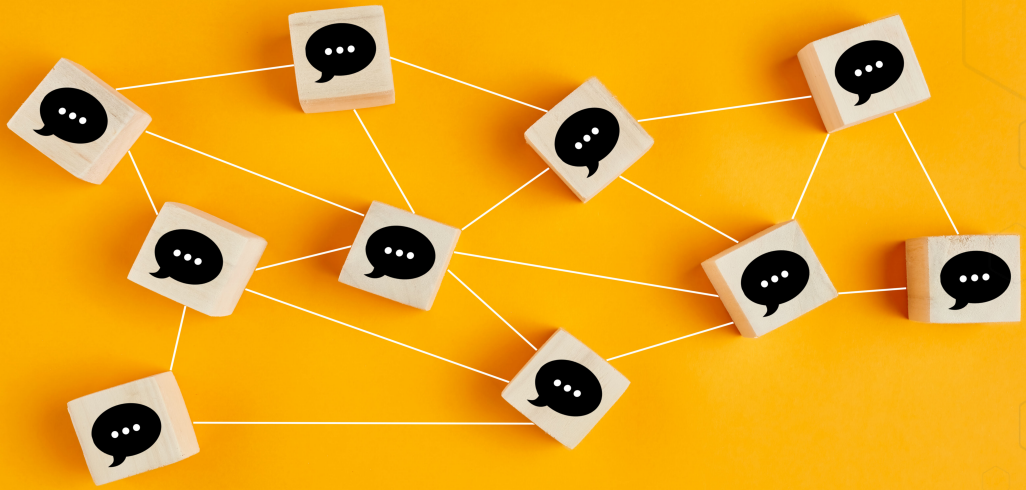
Conclusion

Nous avons proposé *Chained CmPKE*, un CGKA:

- → Très rapide
- → Plus compact que TreeKEM: $O(N))$ au lieu de $\Omega(N \log N)$
- → Plus simple que TreeKEM
- → Vérifiant les mêmes notions de sécurité (voir papier)

Ainsi que des techniques qui peuvent revêtir un intérêt propre:

1. Les *committing mPKEs*
2. Des mPKEs à base de réseaux plus performants



Des 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 2016Computer 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.