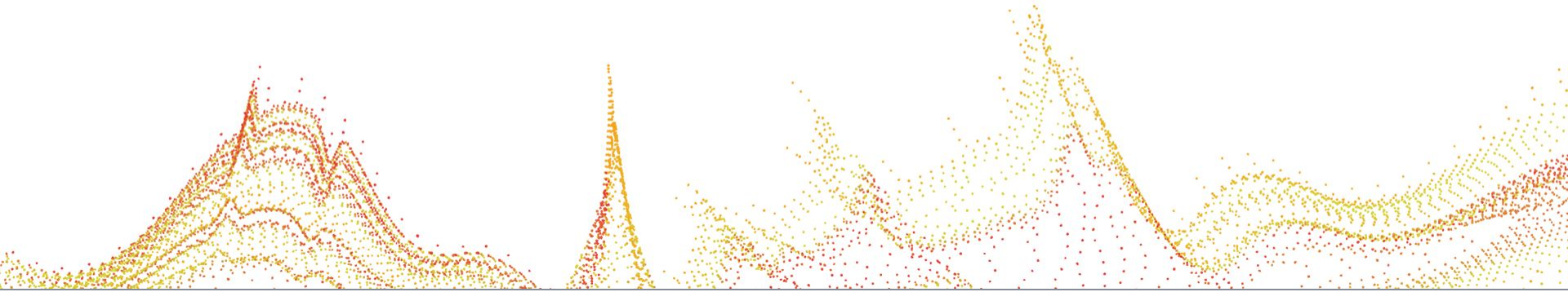
Migrating Protocols to the Post-Quantum Setting: The Case of Signal's Double Ratchet

Yevgeniy Dodis (NYU), Daniel Jost (NYU), Shuichi Katsumata (PQShield + AIST), Thomas Prest (PQShield), Rolfe Schmidt (Signal Messenger)



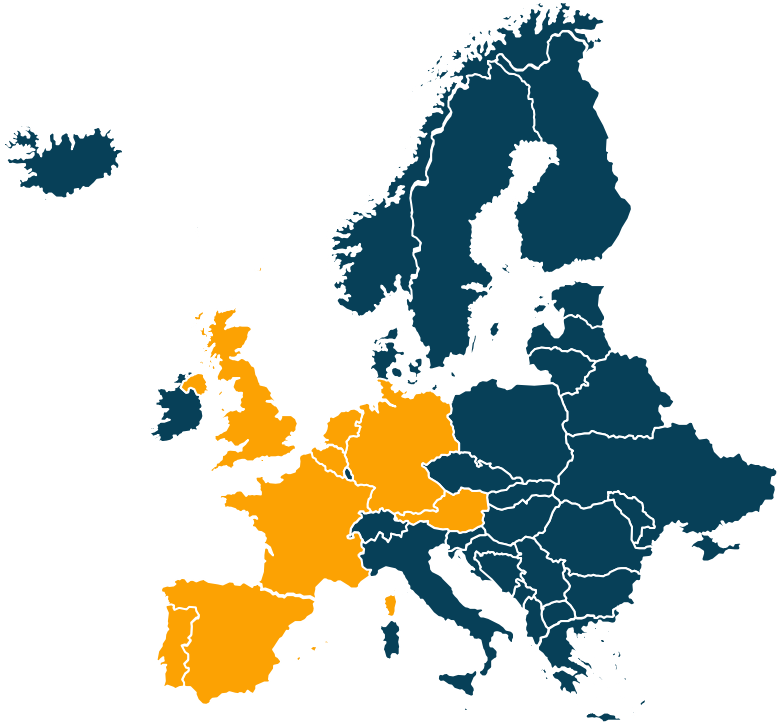
PQShield

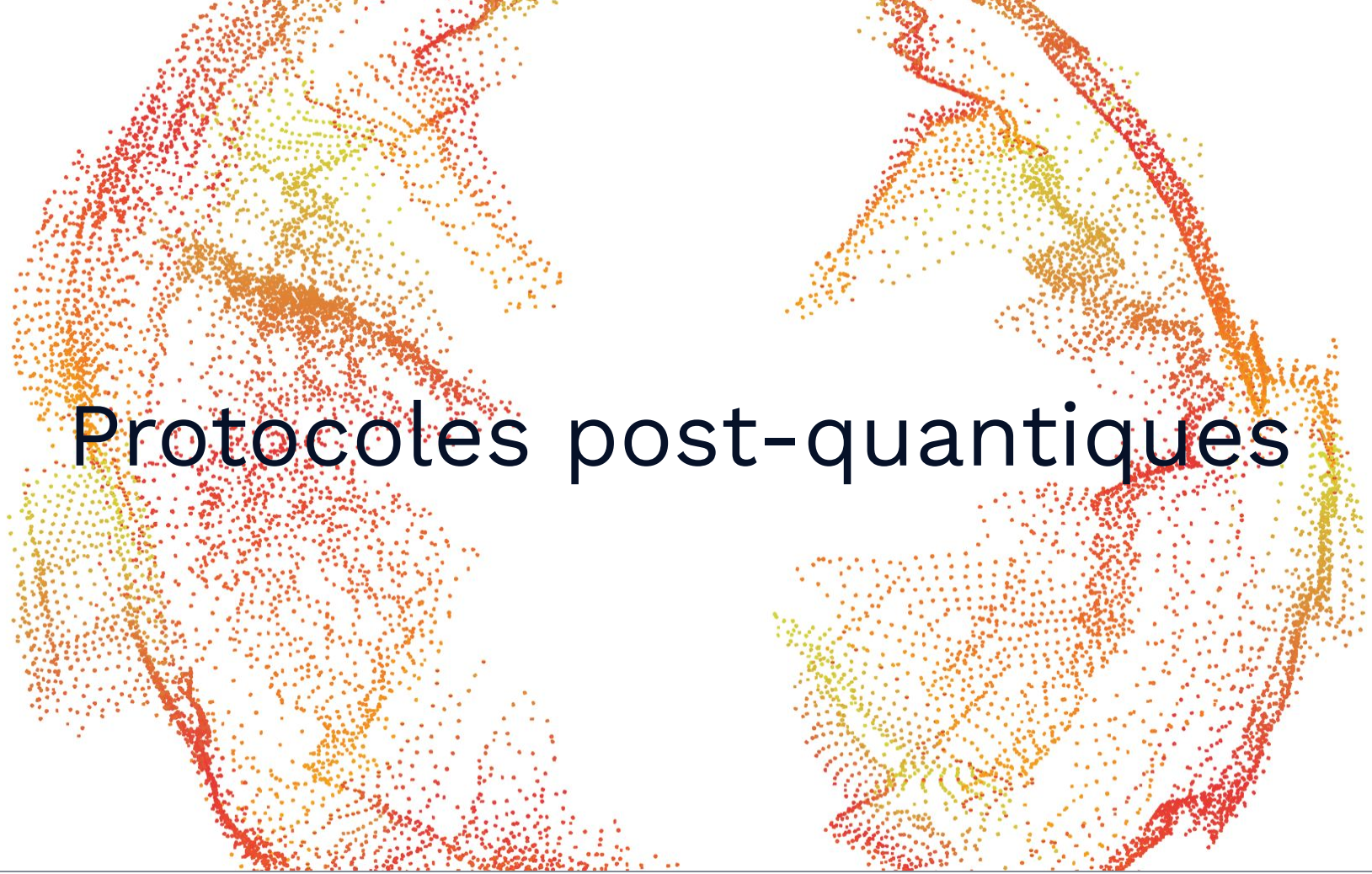
Scale-up européenne spécialisée en cryptographie post-quantique (PQC)

- Implémentations sécurisées
- Guilhem et moi ⇒ équipe Recherche
  - Nouveaux algorithmes
  - Nouveaux protocoles
  - etc.

Thomas Prest

Guilhem Niot (prochain exposé)





Protocoles post-quantiques

PQC: Ce qui est fait & ce qui reste à faire

PQSHIELD

2017
NIST PQC Standardization Process (69 candidats)

2022
4 standards annoncés

2024
FIPS 203 (ML-KEM)
FIPS 204 (ML-DSA)
FIPS 205 (SLH-DSA)

2027
FIPS 206 (FN-DSA)?
FIPS 207 (HQC-KEM)?

~2030
ANSSI, BSI, NSA: migration complète pour les systèmes critiques

Que devons-nous migrer?

Les produits

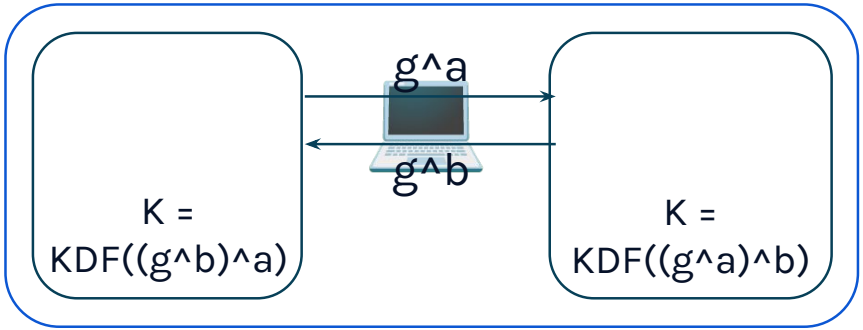
Implémentations logicielles et matérielles, contre-mesures side-channel, certification, etc.

Les protocoles

- TLS (voir https://thomwiggers.nl/)
- DNSSEC
- **Signal (cet exposé)**
- **WireGuard (exposé suivant)**
- etc.

Echange de clef: ECDH vs ML-KEM

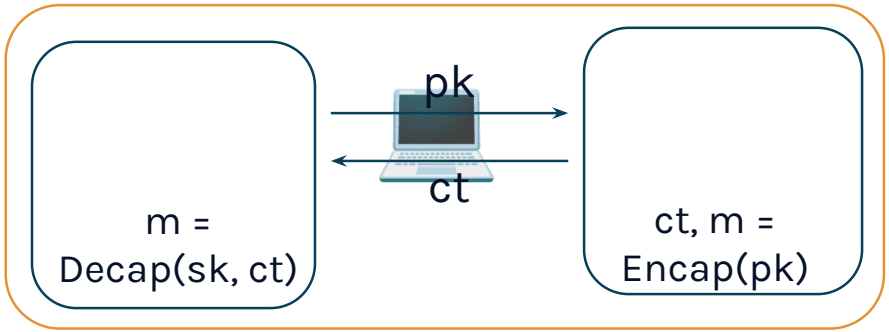
ECDH: 32B + 32B



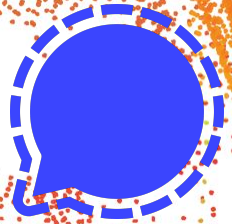
Beaucoup de protocoles pensés pour ECDH...

... ne passent pas avec ML-KEM.

ML-KEM: 1088B + 1184B



Le protocole Signal



Les protocoles de messagerie sécurisée

Messageries sécurisées

- WhatsApp, iMessage, Signal
- Tchap, Olvid
- etc.

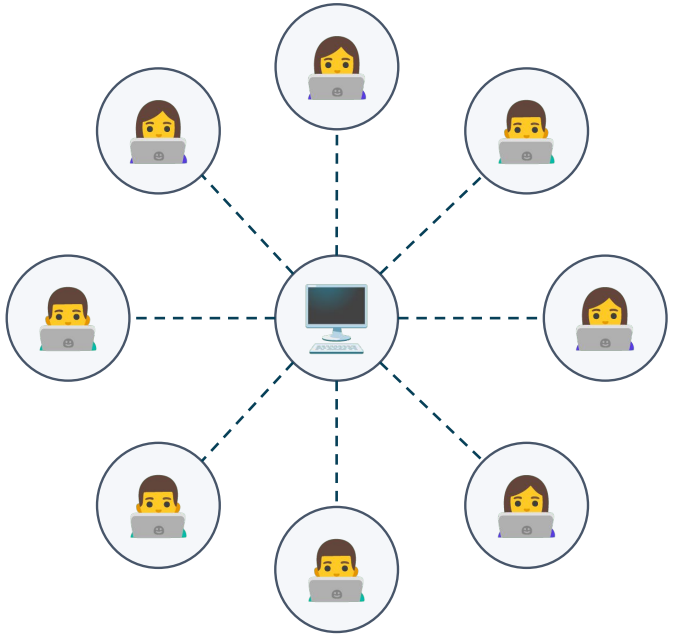
Critères

Asynchrone
(Serveur non-fiable)

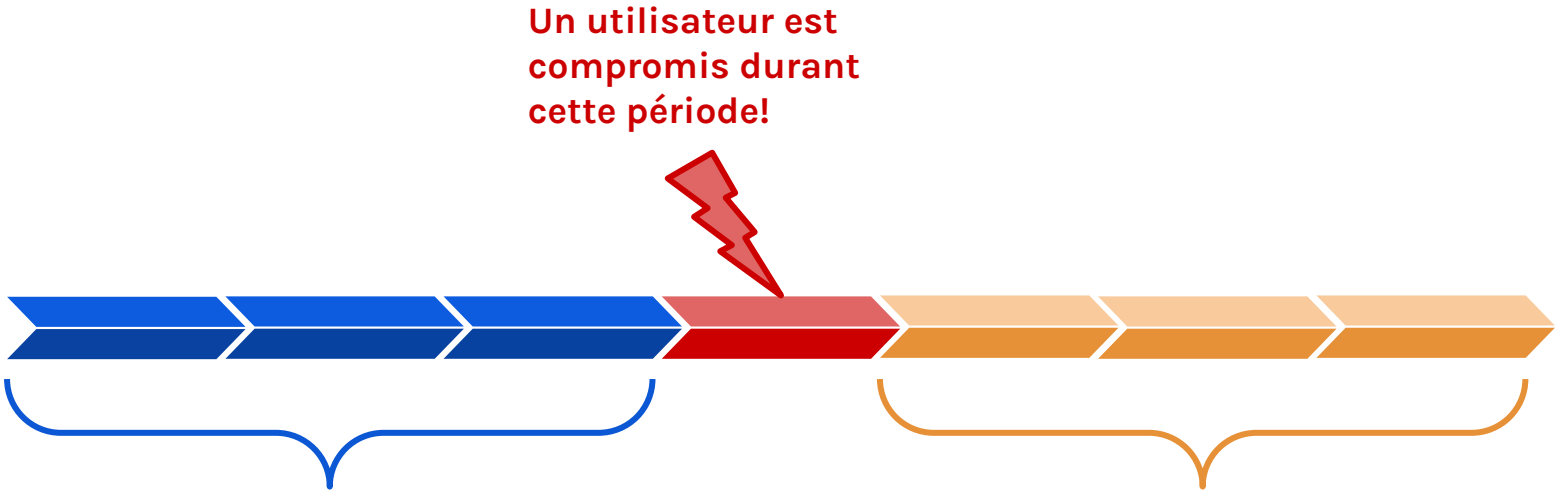
Efficace
Faible coût en bande passante

Sessions longues
Maintien sur plusieurs années

- *Forward secrecy*
- *Post-compromise security*



Forward secrecy & Post-Compromise security



Forward secrecy:
Les messages échangés avant la compromission restent confidentiels.

Post-compromise security:
Les messages échangés après la compromission restent confidentiels.

(Ancien) Signal Protocol = X3DH + Double Ratchet

Alice et Bob établissent une clef symétrique partagée.

X3DH

Alice et Bob chiffrent des messages avec cette clef, et la rafraîchissent.

Double Ratchet

...

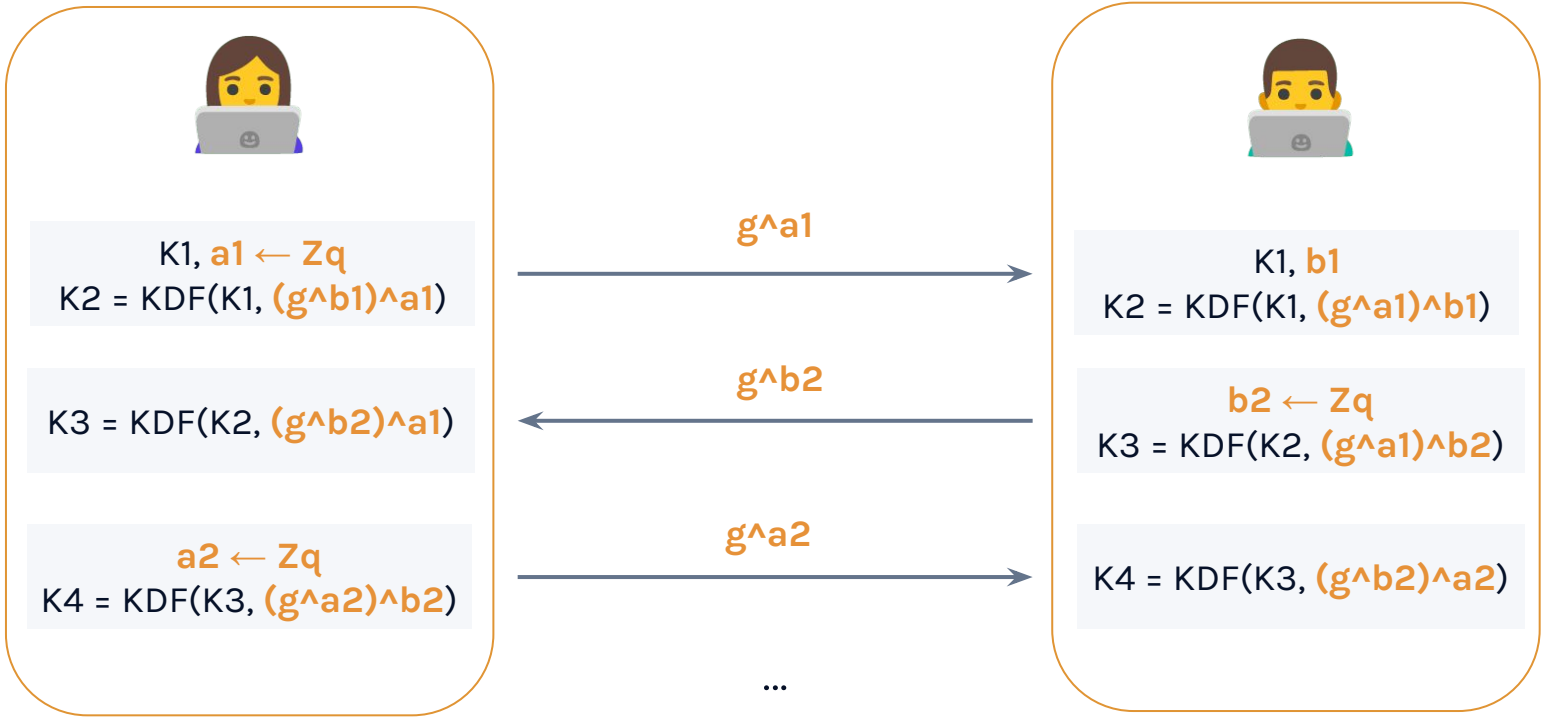
...

Double et Triple Ratchets

Cliquet (= Ratchet) symétrique

Fournit la *forward secrecy*!

Cliquet Diffie-Hellman (Interactif)



Fournit la *post-compromise security*!

Idée sous-jacente: on injecte de l'aléa dans la clef partagée Ki

Cliquet Post-Quantique

PQSHIELD

pka_1 ← Keygen()
(ct1, m1) ← Encap(pkb_1)
K2 = KDF(K1, m1)

pka_1, ct1 →

m1 = Decap(ct1, pkb_1)
K2 = KDF(K1, m1)

m2 = Decap(ct2, pka_1)
K3 = KDF(K2, m2)

← pkb_2, ct2

pkb_2 ← Keygen()
(ct2, m2) ← Encap(pka_1)
K2 = KDF(K1, m1)

...

Fournit la *post-compromise security* post-quantique!

Même idée qu'avant, via un outil différent (ML-KEM au lieu d'ECDH).

Optimisation: codes à effacement

Le problème de la bande passante

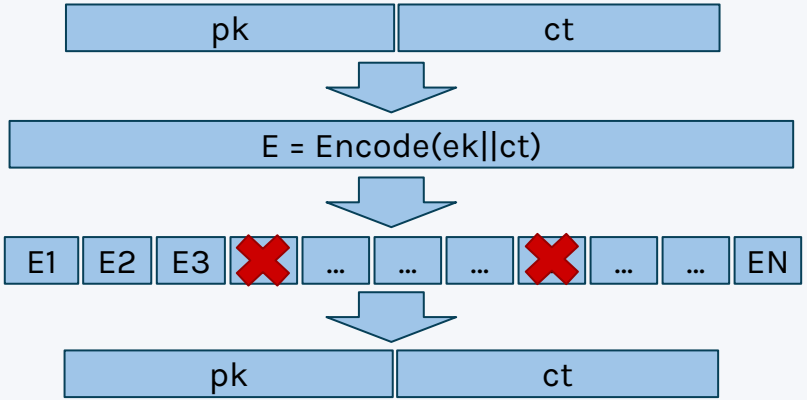
- **Diffie-Hellman:** |pk| = 32 octets ⇒ on envoie pk à chaque message (pour déchiffrement instantané).
- **ML-KEM-768:** |pk| + |ct| = 2 kilo-octets ⇒ impact notable
  - **Apple PQ3:** envoie (pk, ct) environ tous les 50 messages.
    (pk, ct) est renvoyé jusqu'à accusé de réception ⇒ répétitions si l'autre personne est hors-ligne.

La solution de Triple Ratchet

On encode (pk, ct) avec des **codes à effacement (Reed-Solomon),** puis on découpe en plusieurs morceaux.

Advantages:

- Flexible.
- Surcoût par message minimal.
- Robuste against contre les messages perdus.



Protocoles à cliquets – vue d'ensemble

	Cliquet symétrique	Cliquet Diffie-Hellman	Cliquet post-quantique
Signal Double Ratchet	Chaque message	Chaque message	--
Apple PQ3	Chaque message	Chaque message	Tous les ~50 messages
Signal Triple Ratchet	Chaque message	Chaque message	En continu (codes à effacement)

Plusieurs optimisations omises par manque de temps.
Nos idées sont déployées par Signal Messenger! https://signal.org/blog/spqr/