

# 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)



# Protocoles post-quantiques

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



## Que devons-nous migrer?



### Les produits

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

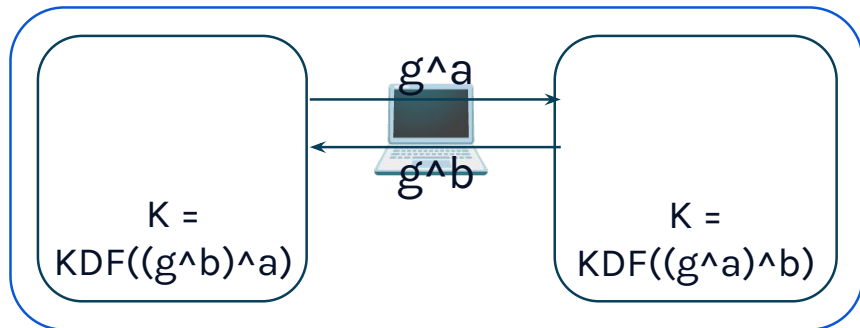


### Les protocoles

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

# ⋮ ⋮ Echange de clef: ECDH vs ML-KEM

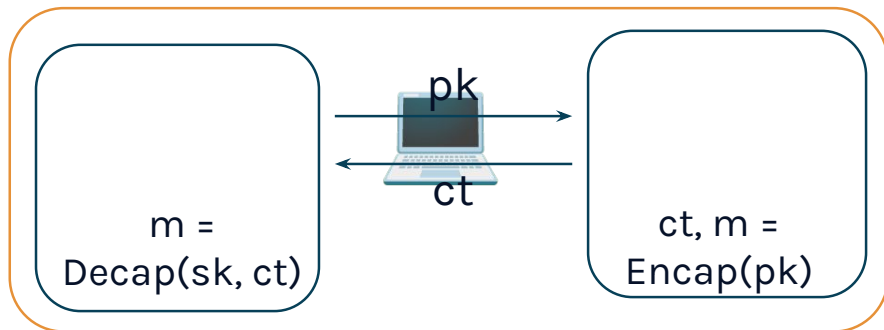
## ECDH: 32B + 32B



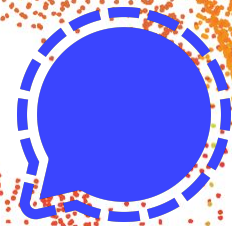
Beaucoup de protocoles pensés pour ECDH...

... 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)

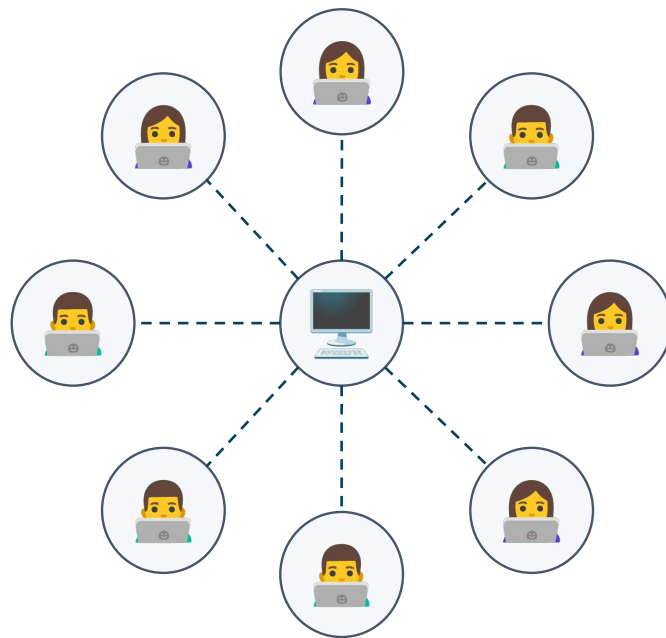
### Efficace

Faible coût en bande passante

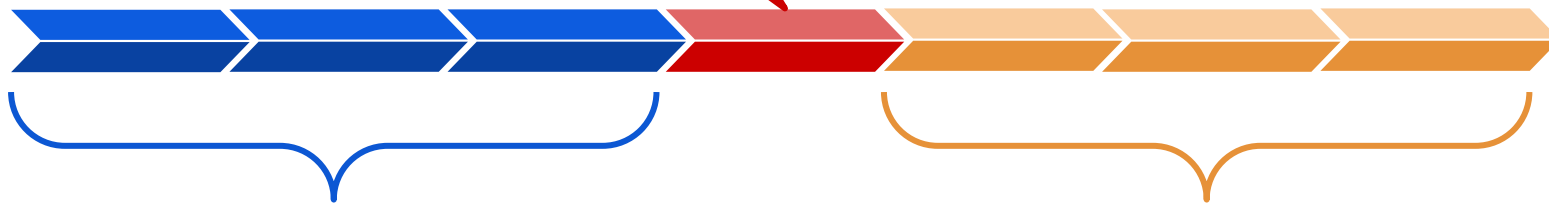
### Sessions longues

Maintien sur plusieurs années

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



Un utilisateur est  
compromis durant  
cette période!



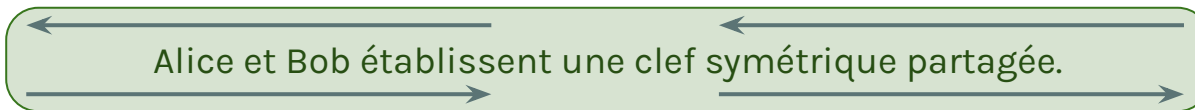
## Forward secrecy:

Les messages échangés avant la compromission restent confidentiels.

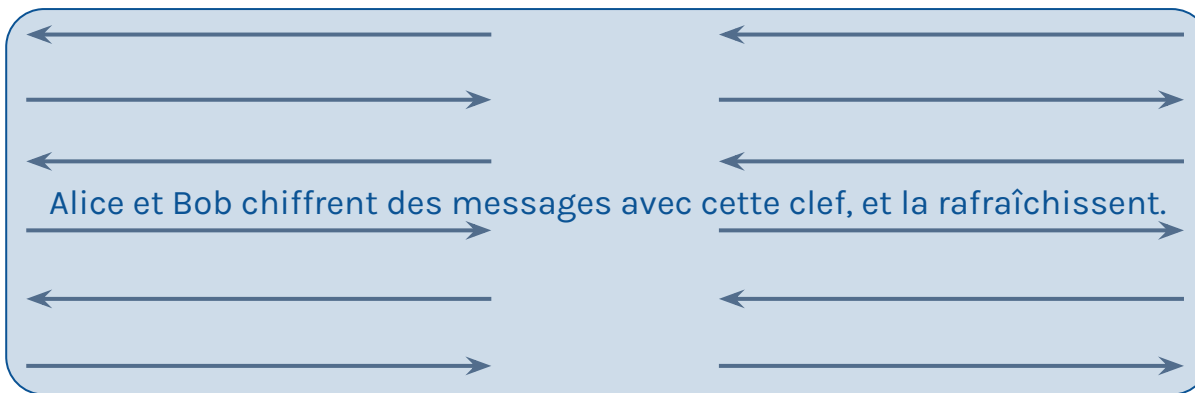
## Post-compromise security:

Les messages échangés après la compromission restent confidentiels.

# ⋮ ⋮ (Ancien) Signal Protocol = X3DH + Double Ratchet



X3DH



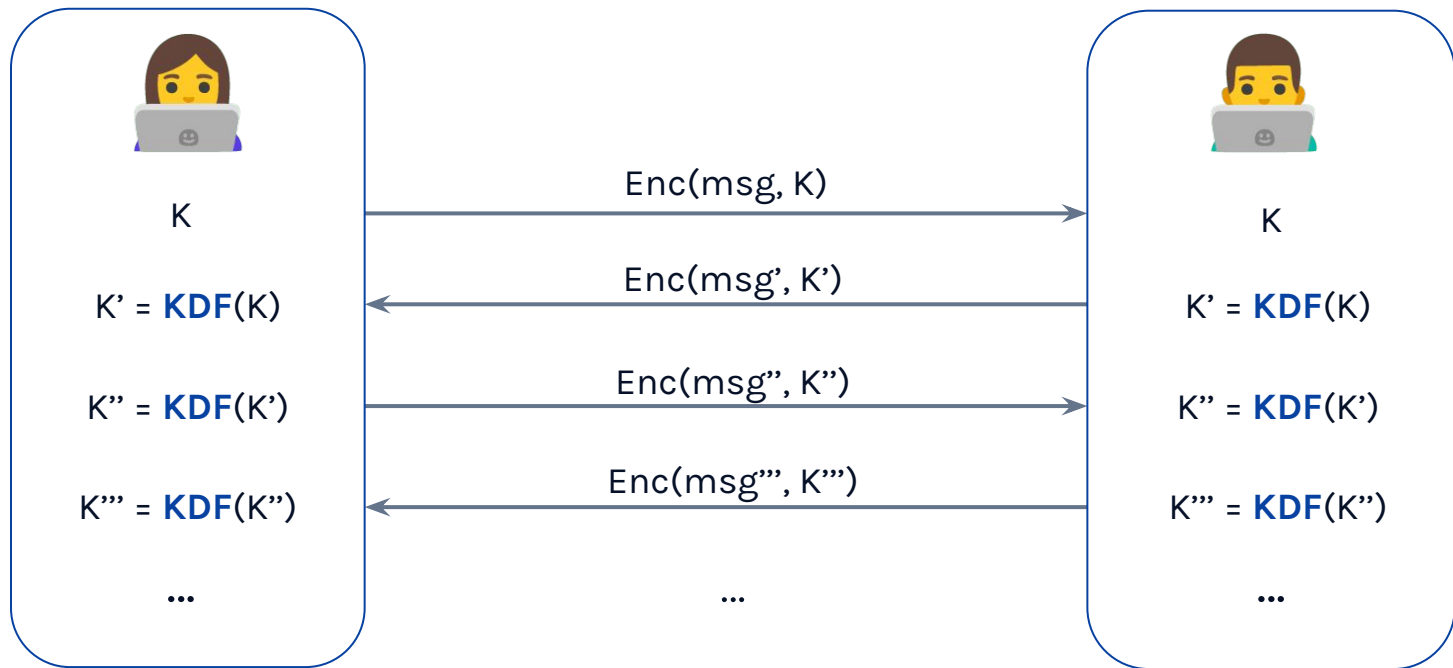
Double Ratchet



# Double et Triple Ratchets



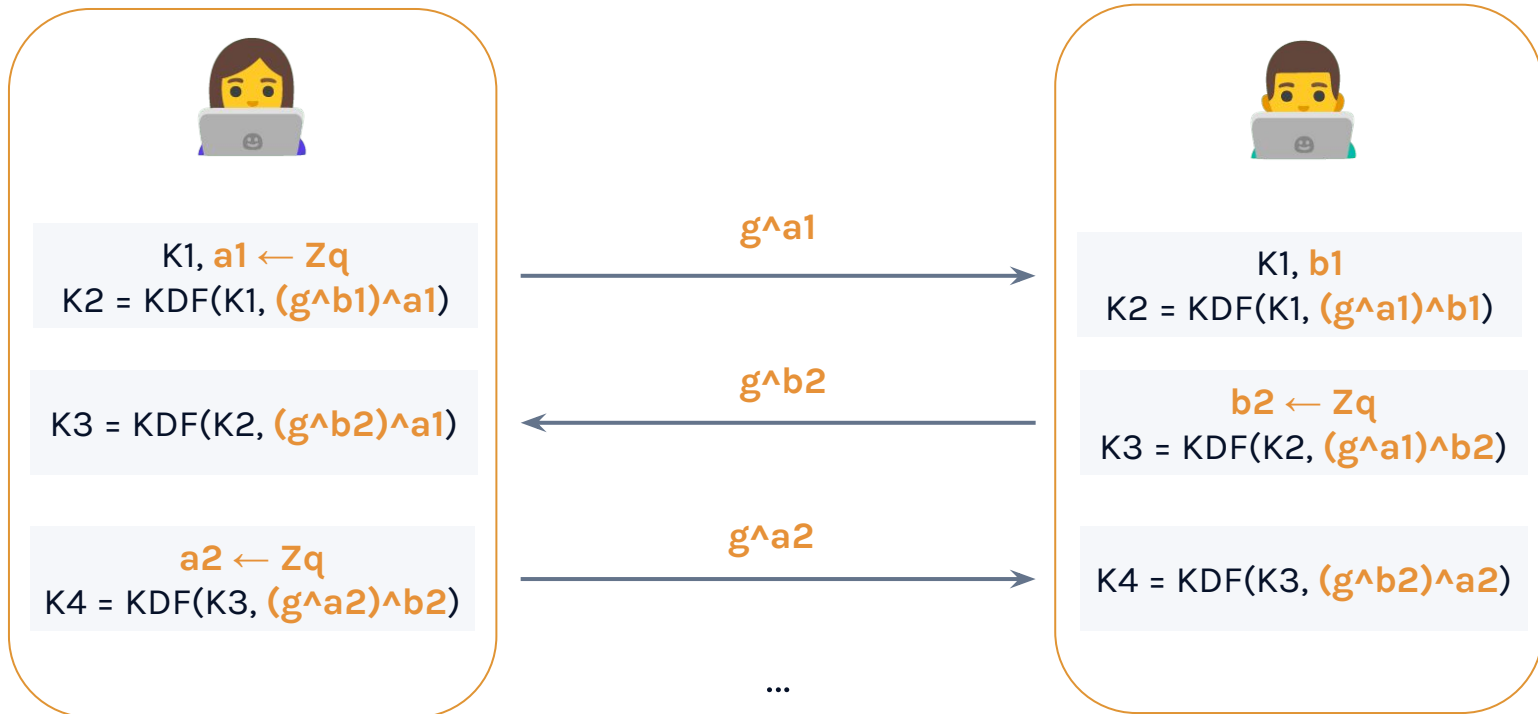
# Clignet (= Ratchet) symétrique



**Fournit la *forward secrecy*!**



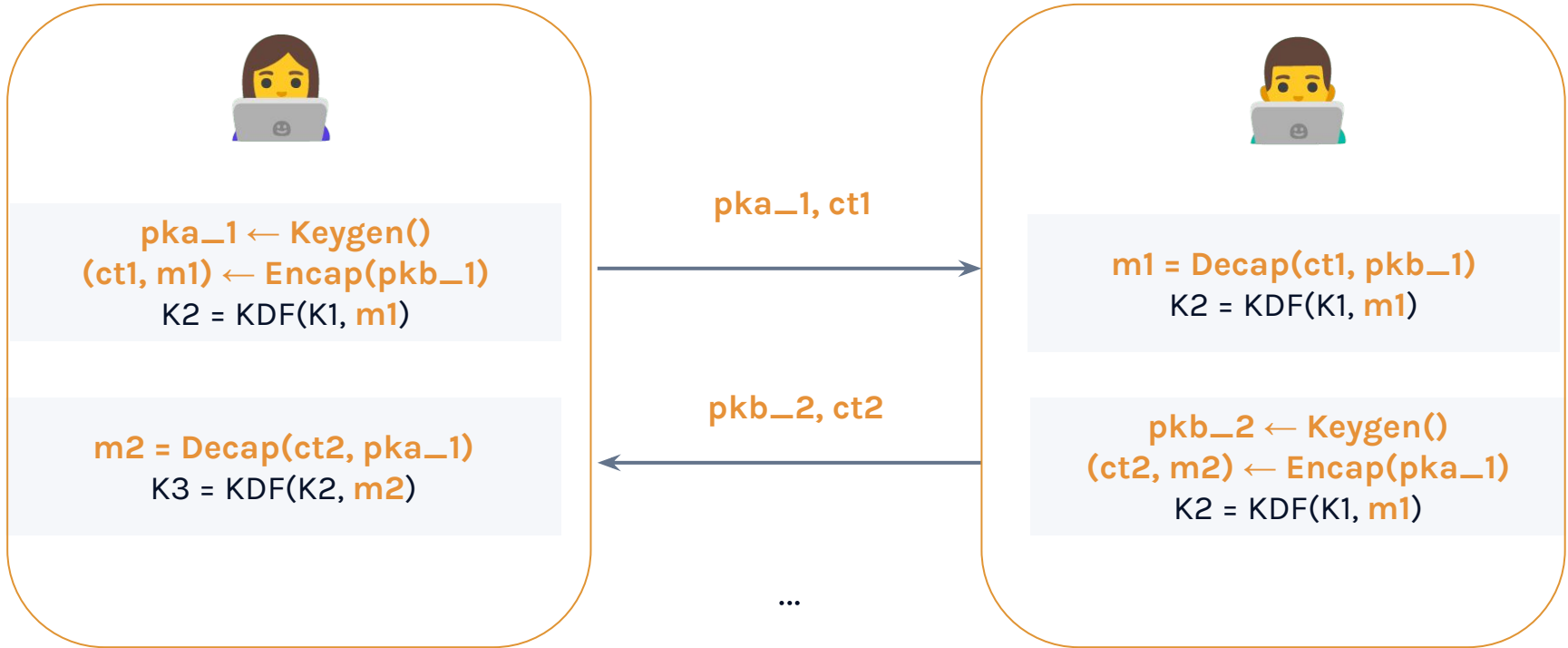
# Cliquet Diffie-Hellman (Interactif)



**Fournit la post-compromise security!**

**Idée sous-jacente:** on injecte de l'aléa dans la clef partagée  $K_i$

# :: Cliquet Post-Quantique



**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

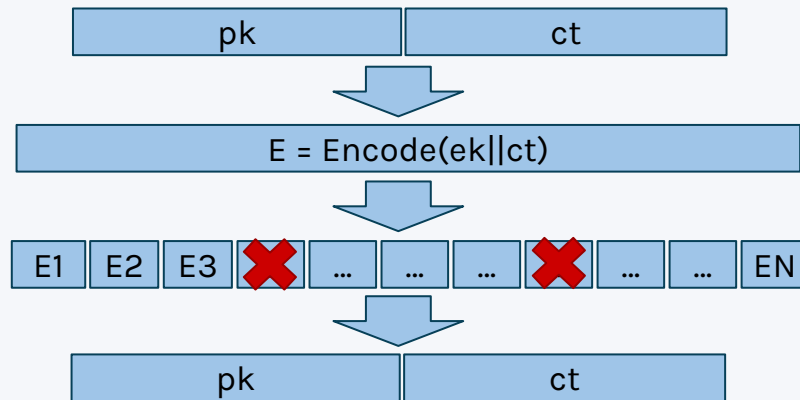
- **Diffie-Hellman:**  $|pk| = 32$  octets  $\Rightarrow$  on envoie  $pk$  à chaque message (pour déchiffrement instantané).
- **ML-KEM-768:**  $|pk| + |ct| = 2$  kilo-octets  $\Rightarrow$  impact notable
  - **Apple PQ3:** envoie  $(pk, ct)$  environ tous les 50 messages.  
( $pk, ct$ ) est renvoyé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

### Avantages:

- 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/>

