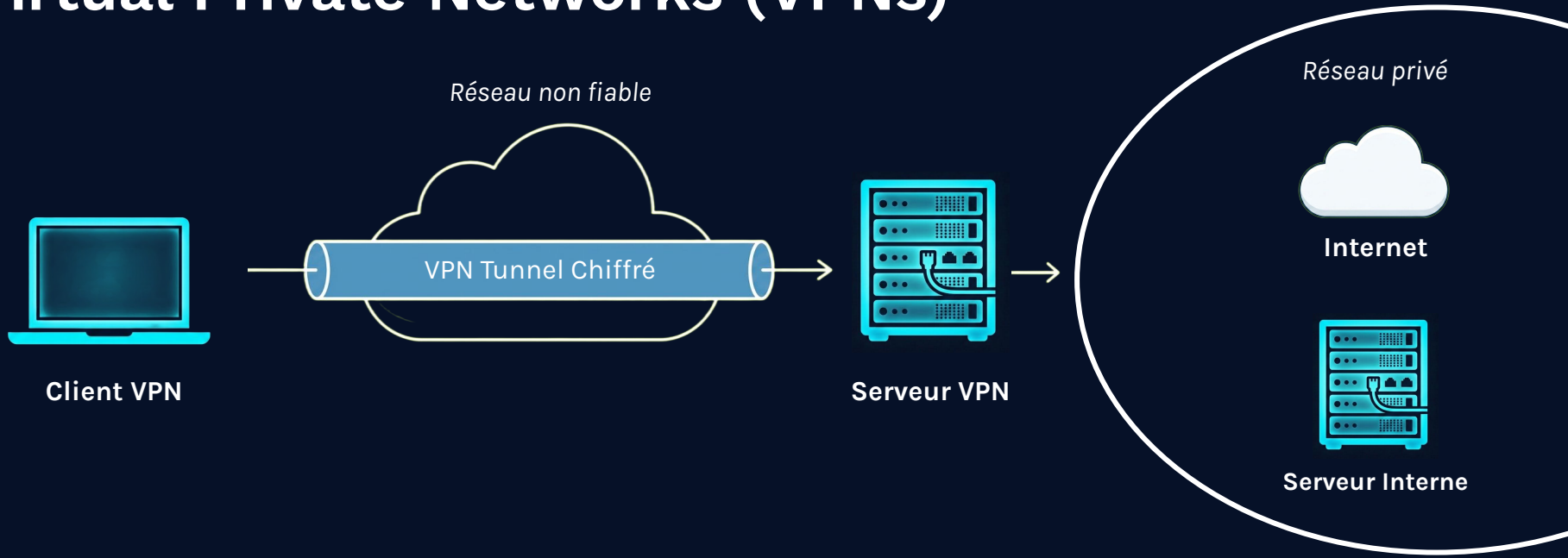
PQSHIELD

Migrating Protocols to the Post-Quantum Setting

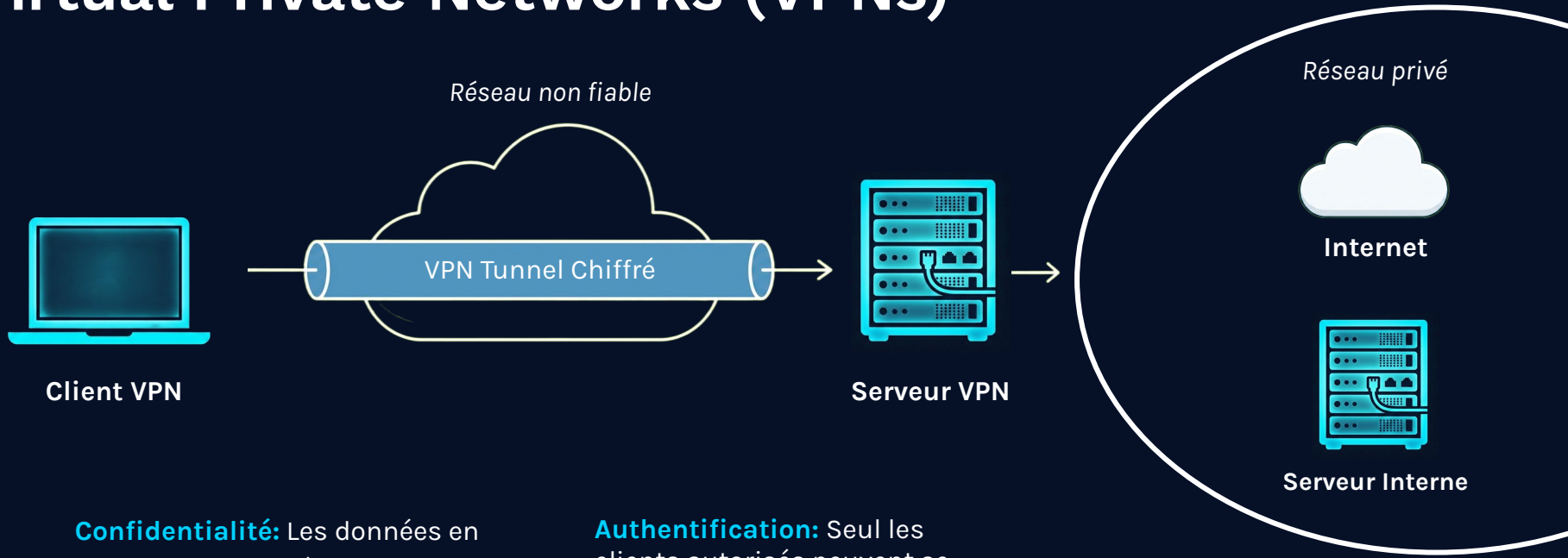
The Case of WireGuard

Keitaro Hashimoto[1], Shuichi Katsumata[1,2], **Guilhem Niot**[2], Thom Wiggers[2]
[1]AIST, [2]PQShield

Virtual Private Networks (VPNs)



Virtual Private Networks (VPNs)



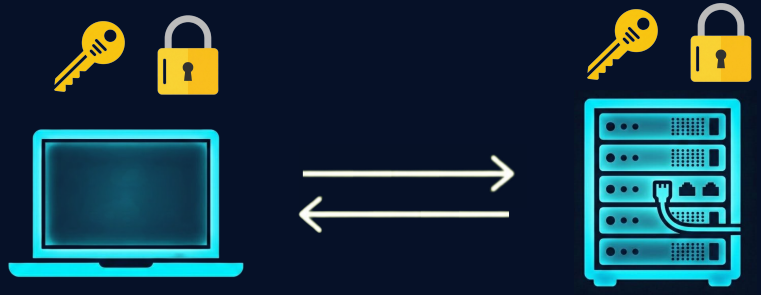
Confidentialité: Les données en transit sont chiffrées et non visibles sur le réseau.

Authentification: Seul les clients autorisés peuvent se connecter au serveur et accéder au réseau privé.

Virtual Private Networks (VPNs)

Phase 1: Mise en place

Le client et serveur exécutent un protocole avec leur clé cryptographique long-terme pour mettre un canal privé.



Produit un secret partagé.

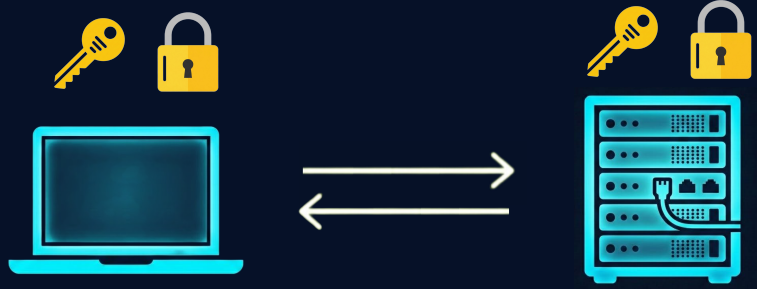
*****

Phase 2: Communication

Virtual Private Networks (VPNs)

Phase 1: Mise en place

Le client et serveur exécutent un protocole avec leur clé cryptographique long-terme pour mettre un canal privé.



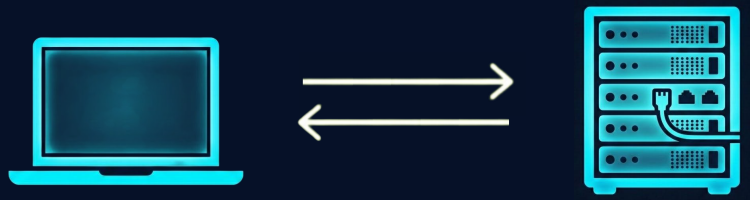
Produit un secret partagé.

*****

Phase 2: Communication

Utilise de la cryptographie symétrique peu coûteuse pour chiffrer le trafic avec le secret partagé.

*****

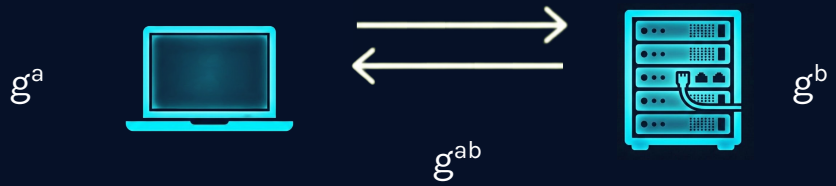


WireGuard

- VPN proposé par Donenfeld [NDSS:Don17], basé sur le Diffie-Hellman (DH) Key Exchange.
- Choix cryptographiques tranchés: pas de mécanisme de négociation, seul un petit ensemble de constructions cryptographiques modernes.
  - Permet une mise en place (phase 1) en un seul tour
  - Fonctionnalités de sécurité uniques.
- Intégré au noyau Linux.

WireGuard Post-Quantique ?

- **Phase 2:** ok, cryptographie symétrique.
- **Phase 1:**



Phase 1: Mise en place

Indistinguabilité des clés : La clé de sortie est pseudo-aléatoire même en cas de fuite de clé statique (ou éphémère).

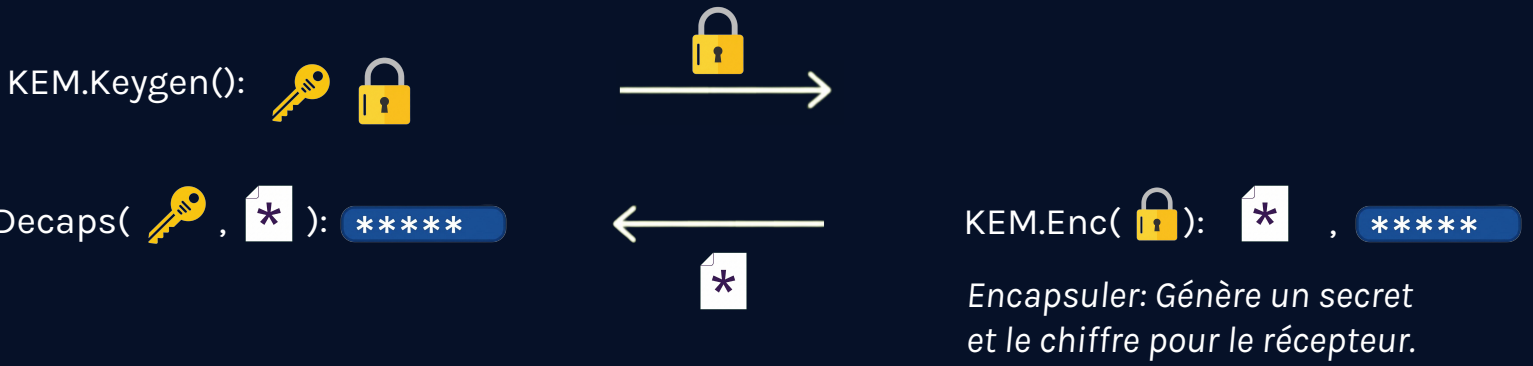
Authentification d'entité : Aucune partie ne peut usurper l'identité d'une autre entité.

Atténuation de DoS : Le premier message du client vers le serveur est authentifié.

Dissimulation d'identité : La transcription seule ne révèle pas l'identité des parties communicantes.

WireGuard Post-Quantique ?

- **Remplaçons DH par des KEMs!**



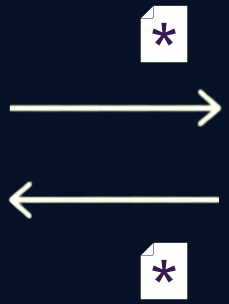
Authentification côté récepteur : La décapsulation est liée à la clé statique du récepteur.

Sécurité côté expéditeur : Aucune entité autre que l'expéditeur et le récepteur ne peut dériver le secret. **Mais l'expéditeur n'est pas authentifié.**

WireGuard Post-Quantique ?

- **Remplaçons DH par des KEMs! Hülsing et al. (https://eprint.iacr.org/2020/379)**

Encapsuler vers la clé statique du Serveur.



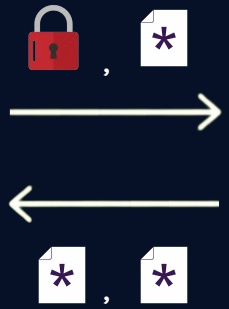
Encapsuler vers la clé statique du Client.

WireGuard Post-Quantique ?

- **Remplaçons DH par des KEMs! Hülsing et al. (https://eprint.iacr.org/2020/379)**

Encapsuler vers la clé statique du Serveur.

+ Générer une clé KEM éphémère



Encapsuler vers la clé statique du Client.

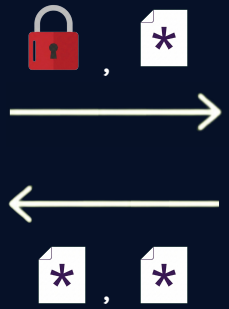
+ Encapsuler vers la clé éphémère du Client.

WireGuard Post-Quantique ?

- **Remplaçons DH par des KEMs! Hülsing et al. (https://eprint.iacr.org/2020/379)**

Encapsuler vers la clé statique du Serveur.

+ Générer une clé KEM éphémère



Encapsuler vers la clé statique du Client.

+ Encapsuler vers la clé éphémère du Client.

Note : Le client n'est pas authentifié dans le premier message avec le KEM seul (utile pour la protection DoS) ⟶ Rétablir la protection avec des MAC symétriques et un secret partagé statique.

WireGuard Post-Quantique : La taille des primitives PQ

Pour offrir ses propriétés intéressantes (simplicité, efficacité, protection DoS), WireGuard repose sur UDP. Pas de fragmentation. **Le standard PQ KEM ne rentre pas !**

Paquet UDP : ~1200 octets

ML-KEM-512 :


= 800 octets


= 768 octets

WireGuard Post-Quantique : La taille des primitives PQ

Pour offrir ses propriétés intéressantes (simplicité, efficacité, protection DoS), WireGuard repose sur UDP. Pas de fragmentation. **Le standard PQ KEM ne rentre pas !**

Paquet UDP : ~1200 octets

Hülsing et al. suggère d'utiliser McEliece (KEM PQ alternatif) pour le KEM à long terme.

= 96 octets.

+ Saber pour le KEM éphémère (~900 octets clés et textes chiffrés)

… mais le prix est payé ailleurs

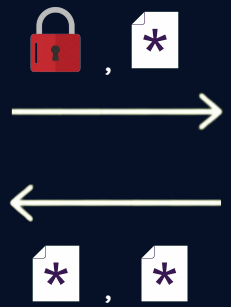
- McEliece possède d'énormes clés publiques  de 260 ko . ⟶ Avec de nombreux clients, la mémoire du serveur serait rapidement épuisée.

Aggravé par l'implémentation de WireGuard dans le noyau Linux (allocation mémoire contrainte).

… mais le prix est payé ailleurs

- McEliece possède d'énormes clés publiques de 260 ko . → Avec de nombreux clients, la mémoire du serveur serait rapidement épuisée.

Aggravé par l'implémentation de WireGuard dans le noyau Linux (allocation mémoire contrainte).



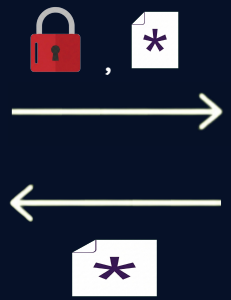
Notre proposition : KEM Renforcé

- Utiliser un KEM de style ML-KEM.
- Encapsuler à la fois les clés statiques et éphémères du client pour réduire la taille du deuxième message.
- Nécessite les deux clés pour retrouver le secret partagé.

… mais le prix est payé ailleurs

- McEliece possède d'énormes clés publiques de 260 ko . → Avec de nombreux clients, la mémoire du serveur serait rapidement épuisée.

Aggravé par l'implémentation de WireGuard dans le noyau Linux (allocation mémoire contrainte).



Notre proposition : KEM Renforcé

- Utiliser un KEM de style ML-KEM.
- Encapsuler à la fois les clés statiques et éphémères du client pour réduire la taille du deuxième message.
- Nécessite les deux clés pour retrouver le secret partagé.

KEM Renforcé

ML-KEM:

KEM.Enc( ):  ,

KEM.Enc( ):  ,

KEM Renforcé:

RKEM.Enc(,): 

KEM Renforcé

ML-KEM:

KEM.Enc( ):  ,

KEM.Enc( ):  ,

KEM Renforcé:

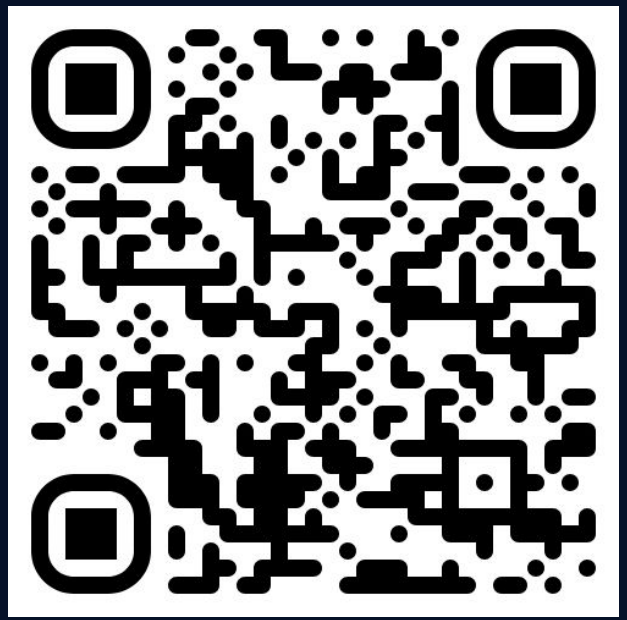
RKEM.Enc(,):  , ,

Le texte chiffré RKEM est 40 % plus petit que deux textes chiffrés ML-KEM-512. Les clés statiques et éphémères RKEM font environ 1 ko : assez petites pour tenir dans l'UDP, et bien plus petites que McEliece qui avait des clés publiques de 260 ko !

Quelques points à retenir

- Nous ne disposons pas d'un analogue PQ direct de Diffie-Hellman (DH) : les protocoles doivent être modifiés en profondeur, et il peut être difficile d'obtenir les mêmes propriétés de sécurité.
- Les primitives PQ sont nettement plus volumineuses, dépassant les limites habituelles du réseau. Ici, nous avons conçu une primitive spécialisée pour compresser deux éléments PQ ensemble et réussir à respecter ces limites.

Questions ?



Revisiting PQ WireGuard: A Comprehensive Security Analysis With a New Design Using Reinforced KEMs

Keitaro Hashimoto[1], Shuichi Katsumata[1,2], **Guilhem Niot**[2], Thom Wiggers[2]

[1]AIST, [2]PQShield