Vérification formelle de la proposition de l'ETSI sur un protocole de QKD

Thomas Prévost, Bruno Martin, Olivier Alibart

Vérification formelle de la proposition de l'ETSI sur un protocole de QKD

Thomas Prévost
Université Côte d'Azur
I3S - CNRS
Sophia-Antipolis, France
thomas.prevost@univ-cotedazur.fr

Bruno Martin
Université Côte d'Azur
I3S - CNRS
Sophia-Antipolis, France
bruno.martin@univ-cotedazur.fr

Olivier Alibart
Université Côte d'Azur
InPhyNi - CNRS
Nice, France
olivier.alibart@univ-cotedazur.fr

***Abstract*—Cet article présente une vérification formelle par ProVerif de la proposition de l'ETSI (European Telecommunications Standards Institute) pour standardiser l'utilisation d'un protocole de distribution quantique de clé.**

***Index Terms*—ETSI, QKD, formal verification, ProVerif.**

I. Introduction

Les protocoles d'échange de clé permettent à deux parties qui ne se connaissaient pas au préalable de partager une clé cryptographique commune, afin d'échanger des messages chiffrés symétriquement par la suite.

Les mécanismes utilisés jusqu'à présent pour échanger une clé symétrique entre deux parties se basent sur la cryptographie à clé publique. La sécurité des protocoles repose donc sur la difficulté, connaissant la clé publique, de retrouver la clé privée ou la clé chiffrée avec la clé publique. Les algorithmes de chiffrement asymétriques actuels ne nous offriront plus de telles garanties avec l'avènement de l'ordinateur quantique [1].

La sécurité d'un protocole de transmission quantique de clé (*Quantum Key Distribution*, QKD) repose sur les propriétés de la physique quantique, notamment sur le théorème de non-clonage [2]. Ce théorème stipule qu'il est impossible de cloner parfaitement l'état quantique d'une particule, un qubit. Si un attaquant essaie de lire les qubits échangés par les deux participants (le plus souvent un état de polarisation d'un photon), alors il va nécessairement modifier l'état quantique et pourra donc être détecté à la volée. La sécurité du protocole repose donc sur la possibilité de détection à la volée d'un attaquant qui écouterait les communications.

Une des limitations de la QKD demeure néanmoins la distance géographique maximale à laquelle deux parties peuvent échanger, pour le moment quelques centaines de kilomètres [3].

L'ETSI (European Telecommunications Standards Institute) a proposé un standard de protocole pour un réseau de QKD [4]. Nous en proposons ici une vérification formelle en utilisant l'outil ProVerif.

II. Le standard de l'ETSI

A. Les différentes parties présentes sur le réseau

Dans le protocole de l'ETSI, deux types d'entités cohabitent : les *Key Management Entities* (KME) en charge de l'échange quantique de clé et les *Secure Application Entitites* (SAE). Ces dernières correspondent aux applications finales qui vont par la suite utiliser les clés échangées.

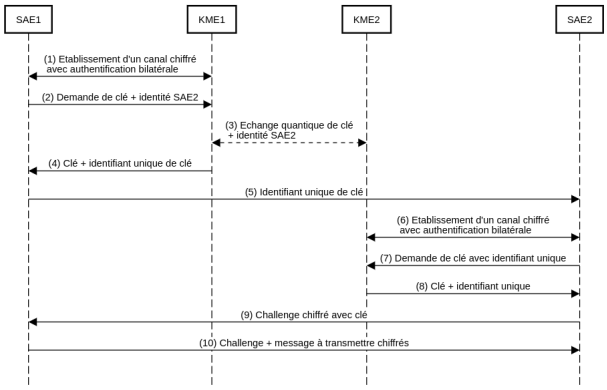

Fig. 1. Schéma simplifié du protocole de QKD de l'ETSI tel que nous l'avons vérifié avec ProVerif.

Le protocole considère que plusieurs SAEs peuvent partager un même KME dans une zone "sécurisée", c'est-à-dire dans laquelle il est possible d'utiliser des protocoles cryptographiques standards (ici le protocole TLS, avec une transmission de clé protégée par des algorithmes classiques). Le protocole fait en outre l'hypothèse que l'ensemble des KMEs sur le réseau global sont "de confiance".

La proposition de standard permet également un échange de clé entre trois SAEs ou davantage. En outre, il est possible de spécifier des "options" au moment de la requête de clé, permettant de restreindre la distribution de clé aux SAEs disposant par exemple des versions à jour des logiciels. Nous avons choisi d'ignorer ces aspects dans notre vérification formelle, puisqu'ils ne changent rien au fonctionnement du protocole.

B. Déroulé du protocole simplifié

On appelle "maître" le SAE souhaitant réaliser un échange quantique de clé avec un pair distant. On suppose ici que les deux SAEs se trouvent dans des zones sécurisées distinctes, et donc ne dépendant pas du même KME.

Comme décrit dans la Fig. 1, le SAE 1 (maître) commence par contacter le KME de sa zone, au travers du protocole

HTTPS avec une authentification bilatérale par certificats (1). Il communique au KME 1 un identifiant désignant le SAE 2 (esclave) (2). À ce moment (3), le KME 1 (maître) va échanger une clé commune avec le KME 2, et va lui indiquer l'identifiant du SAE 2. L'échange de clé se fait via une liaison quantique, au travers d'une fibre noire, c'est à dire une fibre protégée contre les interférences extérieures, adaptée au transport de photons uniques. Finalement, le KME 1 va retourner au SAE 1 un identifiant unique pour la clé qui vient d'être échangée (4).

Le SAE 1 va ensuite, via une liaison classique, communiquer directement au SAE 2 l'identifiant unique de clé (5). Le SAE 2 va alors contacter son KME (6), via une API REST en HTTPS, pour lui demander la clé correspondant à l'identifiant (7). Le KME 2 authentifie le SAE via un certificat TLS client. Finalement, il lui retourne la clé (8), les deux SAEs partagent alors une clé cryptographique commune parfaitement aléatoire.

Pour une bonne authentification entre les deux SAEs, le SAE 2 commence par envoyer au SAE 1 un défi sous la forme d'un jeton aléatoire, chiffré avec la clé fournie par le KME 2 (9). Le SAE 1 va alors répondre un acquittement, son message final accompagné du jeton chiffré avec cette même clé (10).

III. Hypothèses et vérification formelle

Nous avons utilisé le logiciel ProVerif, qui traduit le protocole en contraintes logiques et tente d'exhiber un contre-exemple et d'en inférer une attaque [5] [6]. La vérification effectuée par ProVerif est prouvée complète.

Nous avons employé plusieurs primitives offertes par ProVerif afin de modéliser le comportement des différents acteurs de notre protocole.

Nous avons considéré des algorithmes de chiffrement parfaits; par exemple on exprime ci-dessous que tout ce qui est chiffré par *senc* est déchiffrable par *sdec* dans le langage de ProVerif.

```
type key.
fun senc(bitstring, key): bitstring.
reduc forall m: bitstring, k: key;
    sdec(senc(m, k), k) = m.
```

Afin de modéliser le canal de transmission quantique de clé, nous avons utilisé deux canaux : un canal public et un canal privé. Le canal privé est utilisé pour transmettre la clé, et le canal public transmet tout le reste (identifiant de la clé et du destinataire). Nous avons considéré cette hypothèse valide puisque nous cherchons à prouver la sécurité du protocole de l'ETSI, pas celle de la QKD en général.

La confidentialité persistante est démontrée par la modélisation d'une fuite des clés privées des différents acteurs de l'échange qui surviendrait postérieurement à la phase d'échange de clé. Ces clés privées sont utilisées pour mettre en place les canaux chiffrés et authentifiés entre SAEs et KMEs dans les zones sécurisées.

Finalement, nous avons modélisé un nombre infini d'itérations de protocole.

IV. Résultats

La proposition de protocole de l'ETSI pour la QKD s'est révélée résistante aux attaques passives et actives dans notre modèle, y compris pour les tests de confidentialité persistante.

Il est cependant essentiel d'assurer l'authentification bilatérale entre SAEs et KMEs ainsi qu'entre les SAEs. Pour ce faire, les entités doivent proposer un défi à leur correspondant via un jeton à retourner chiffré avec la clé partagée. La Fig. 1 montre les différentes étapes d'authentification entre les parties :

- Étapes (1) et (6) : les SAEs s'authentifient auprès de leurs KMEs respectifs ;
- Étapes (9) et (10) : une fois le partage de clé terminé, les SAEs s'authentifient entre eux, pour s'assurer qu'ils partagent bien la même clé.

L'ensemble du code ProVerif avec les détails du protocole peut être consulté depuis l'URL suivante : https://gist.github.com/thomasarmel/c2bfc851bb3b19348bf1df90ed041fac

V. Améliorations futures

Dans notre modélisation, nous avons considéré sûrs l'ensemble des KMEs sur le réseau. Il pourrait être intéressant de considérer la possibilité qu'un sous-ensemble minoritaire de KMEs puisse être compromis. La solution serait alors d'imaginer un protocole permettant de router des morceaux de clé par des chemins différents.

VI. Conclusion

Dans cet article, nous avons formalisé la proposition de protocole de l'ETSI pour la QKD. Nous avons vérifié, via l'analyseur ProVerif, que ce dernier était sécurisé contre les attaquants actifs et passifs, y compris dans un modèle de confidentialité persistante.

Il faut cependant assurer une authentification correcte de toutes les parties pendant tout le déroulé du protocole, via une vérification des certificats TLS ou le partage de défis cryptographiques.

References

[1] Bhatia, V., Ramkumar, K. R. (2020). An efficient quantum computing technique for cracking RSA using Shor's algorithm. In 2020 IEEE 5th international conference on computing communication and automation (ICCCA) (pp. 89-94). IEEE.

[2] Zygelman, B., Zygelman, B. (2018). No-cloning theorem, quantum teleportation and spooky correlations. A First Introduction to Quantum Computing and Information, 125-147. Springer International Publishing.

[3] Hiskett, P. A., Rosenberg, D., Peterson, C. G., Hughes, R. J., Nam, S., Lita, A. E., Nordholt, J. E. (2006). Long-distance quantum key distribution in optical fiber. New Journal of Physics, 8(9), 193.

[4] ETSI (2019), Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API.

[5] Blanchet, B., Smyth, B., Cheval, V., Sylvestre, M. (2018). ProVerif 2.00: automatic cryptographic protocol verifier, user manual and tutorial. Version from, 05-16.

[6] Blanchet, B. (2012). Automatic verification of security protocols in the symbolic model: The verifier proverif. In International School on Foundations of Security Analysis and Design (pp. 54-87). Springer International Publishing.