Formal verification of the ETSI proposal on a standard QKD protocol

Thomas Prévost, Bruno Martin, Olivier Alibart

Formal verification of the ETSI proposal on a standard QKD protocol

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*—This article presents a formal verification by ProVerif of the ETSI[1] proposal to standardize the use of a quantum key distribution protocol.**

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

I. Introduction

Key exchange protocols allow two parties who did not know each other beforehand to share a common cryptographic key, in order to subsequently exchange symmetrically encrypted messages.

Current key exchange protocols are based on public key cryptography. Their security is therefore based on the difficulty, knowing the public key, of finding the private key or the key encrypted with the public key. Current asymmetric algorithms will no longer offer such guarantees with the advent of quantum computers [1].

The security of a quantum key distribution protocol (*Quantum Key Distribution*, QKD) is based on the properties of quantum physics, in particular on the non-cloning theorem [2]; which states that it is impossible to perfectly clone the quantum state of a particle, a qubit. If an attacker tries to read the qubits exchanged by the two participants (most often a polarization state of a photon), then she will necessarily modify the quantum state and so can be detected on the fly.

One of the limitations of QKD nevertheless remains the maximum geographic distance at which two parties can exchange, for the moment a few hundred kilometers [3].

ETSI proposed a protocol standard for a QKD network [4]. Here we propose its formal verification using the ProVerif tool.

II. The ETSI standard

A. The different parties on the network

In the ETSI protocol, two types of entities coexist: *Key Management Entities* (KME) in charge of quantum key exchange and *Secure Application Entities* (SAE), the clients. These correspond to the final applications which will subsequently use the exchanged keys.

[1]European Telecommunications Standards Institute

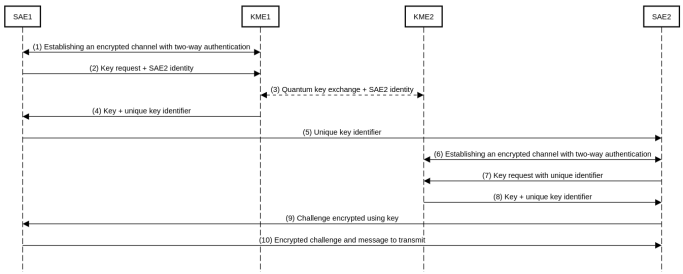

Fig. 1. Simplified diagram of the ETSI QKD protocol as we verified it with ProVerif.

B. Simplified protocol procedure

As described in Fig. 1, SAE 1 (master) begins by contacting the KME in its zone, through HTTPS with bilateral authentication by certificates (1). It communicates to KME 1 an identifier designating SAE 2 (slave) (2). At this moment (3), KME 1 (master) will exchange a common key with KME 2, and will tell it the identifier of SAE 2. The key exchange is done via a quantum link, adapted to the transport of single photons. Finally, KME 1 will return to SAE 1 a unique identifier for the key which has just been exchanged (4).

SAE 1 will then, via a conventional link, communicate directly to SAE 2 the unique key identifier (5). SAE 2 will then contact its KME (6), via a REST API in HTTPS, to ask for the key corresponding to the identifier (7). KME 2 authenticates SAE via a client TLS certificate. Finally, it returns the key (8), the two SAEs then share a perfectly random common cryptographic key.

For proper authentication between the two SAEs, SAE 2 begins by sending SAE 1 an encrypted challenge (9). SAE 1 will then acknowledge and reply to the challenge (10).

III. Assumptions and formal verification

We used ProVerif, which translates the protocol into logical constraints and attempts to exhibit a counterexample and infer an attack [5], [6].

In order to model the quantum key transmission channel, we used a public and a private channel. The private channel is used to transmit the key, and the public channel transmits everything else (key and recipient ID). We considered this

assumption valid since we aim to prove the security of the ETSI protocol, not that of QKD in general.

Forward secrecy is demonstrated by modeling a leak of the private keys of the different participants in the exchange which would occur after the key exchange. These private keys are used to set up encrypted and authenticated channels between SAEs and KMEs in secure areas.

Finally, we modeled an infinite number of protocol iterations.

IV. Results

ETSI's proposed protocol for QKD was found to be resilient to passive and active attacks in our model, including forward secrecy testing.

However, it is essential to ensure bilateral authentication between SAEs and KMEs as well as between SAEs. To do this, the entities must offer a challenge to their correspondent via a token to be returned encrypted with the shared key.

The entire ProVerif code with protocol details can be viewed from the following URL: https://gist.github.com/thomasarmel/c2bfc851bb3b19348bf1df90ed041fac

V. Conclusion

In this article, we have formalized the ETSI protocol proposal for QKD. We verified, via the ProVerif analyzer, that it was secure against active and passive attackers, including in a forward secrecy model.

However, it is necessary to ensure correct authentication of all parties throughout the protocol, via verification of TLS certificates and the sharing of cryptographic challenges.

References

[1] Bhatia, V., Ramkumar, K. R. (2020). An efficient quantum computing technique for cracking RSA using Shor's algorithm. In 2020 IEEE (pp. 89-94).

[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.

[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.

[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.