Formal protocol verification of ETSI GS QKD 014 v1.1.1

Thomas Prévost, Bruno Martin, Olivier Alibart

I3S, Université Nice Côte d'Azur

Agenda

- What is Quantum Key Distribution?
  - Problem
  - Introduction to quantum mechanics
  - QKD: BB84
- *ETSI GS QKD 014 v1.1.1* standard proposal
  - QKD limitations
  - Standard description
- Formal verification
  - ProVerif
  - Verification results

What is Quantum Key Distribution

BB84 example

Quantum Key Distribution (QKD)

Problem: how to ensure reliable forward-secrecy against a “Harvest now, decrypt later” attacker?

Post-Quantum Cryptography?

Support already enabled in some applications (OpenSSH 9+, Google Chrome…)

```
❯ ssh -Q kex
diffie-hellman-group1-sha1
diffie-hellman-group14-sha1
diffie-hellman-group14-sha256
diffie-hellman-group16-sha512
diffie-hellman-group18-sha512
diffie-hellman-group-exchange-sha1
diffie-hellman-group-exchange-sha256
ecdh-sha2-nistp256
ecdh-sha2-nistp384
ecdh-sha2-nistp521
curve25519-sha256
curve25519-sha256@libssh.org
sntrup761x25519-sha512@openssh.com
```

TLS 1.3 hybridized Kyber support

This option enables a combination of X25519 and Kyber in TLS 1.3. – Mac, Windows, Linux, ChromeOS, Android, Fuchsia, Lacros

#enable-tls13-kyber

Enabled

Post-Quantum Safe Algorithm Candidate Cracked In An Hour On A PC

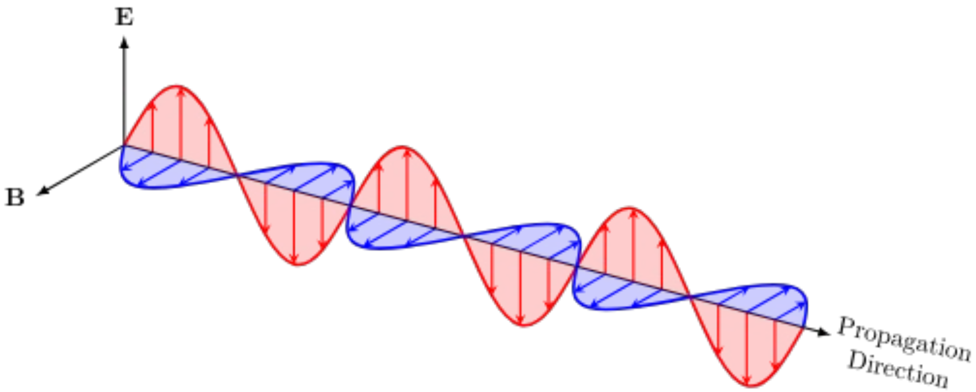
Inherent problem

Is it possible to ensure that traffic eavesdropped now cannot be decrypted later, if the encryption were broken?

We should change the whole paradigm

Short introduction to quantum mechanics: light polarization

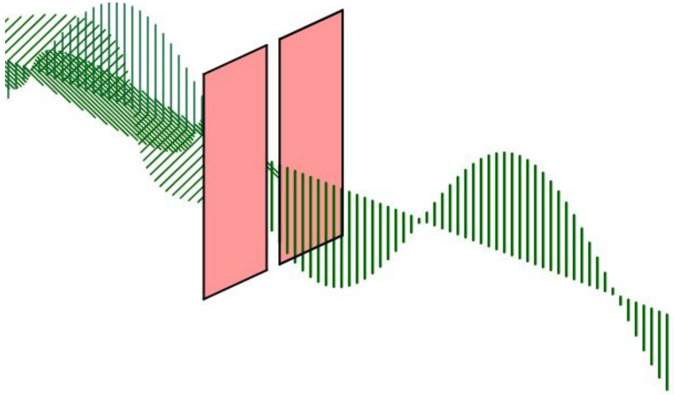
Polarization refers to the orientation of the electric field in a light wave.



Short introduction to quantum mechanics: light polarization

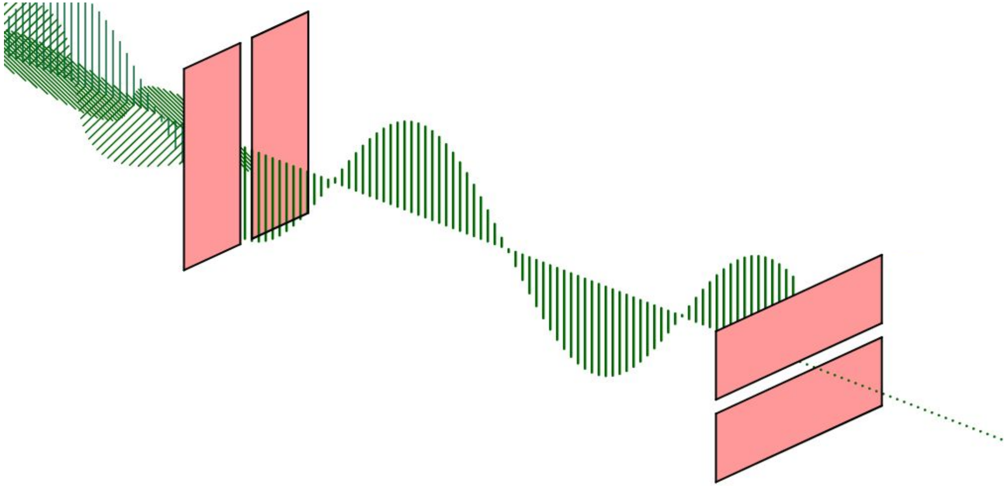
Polarizer: device that selectively transmits light of a specific polarization and blocks light of other polarizations.

A linear polarizers transmit light in a single plane of polarization.



Short introduction to quantum mechanics: light polarization

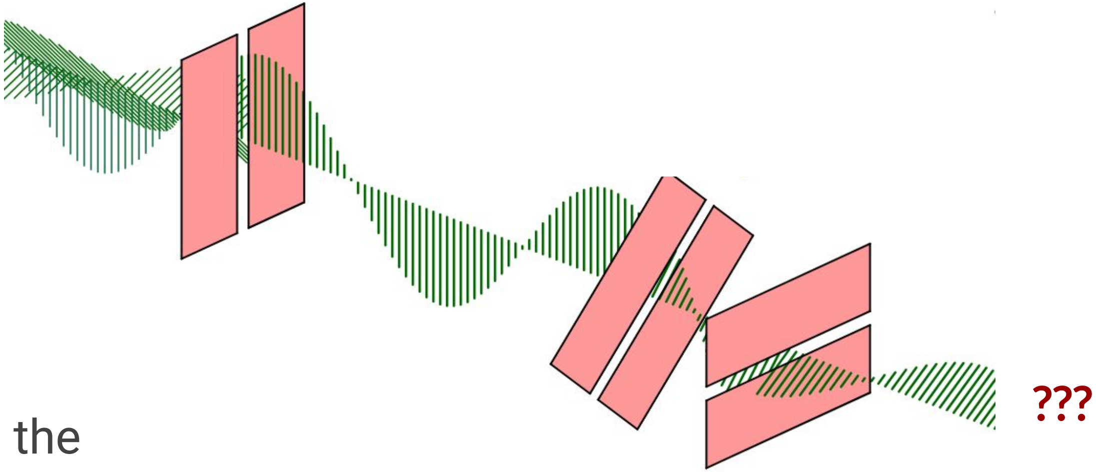
What happens with 2 orthogonal polarizers?



Obviously light is blocked

Short introduction to quantum mechanics: light polarization

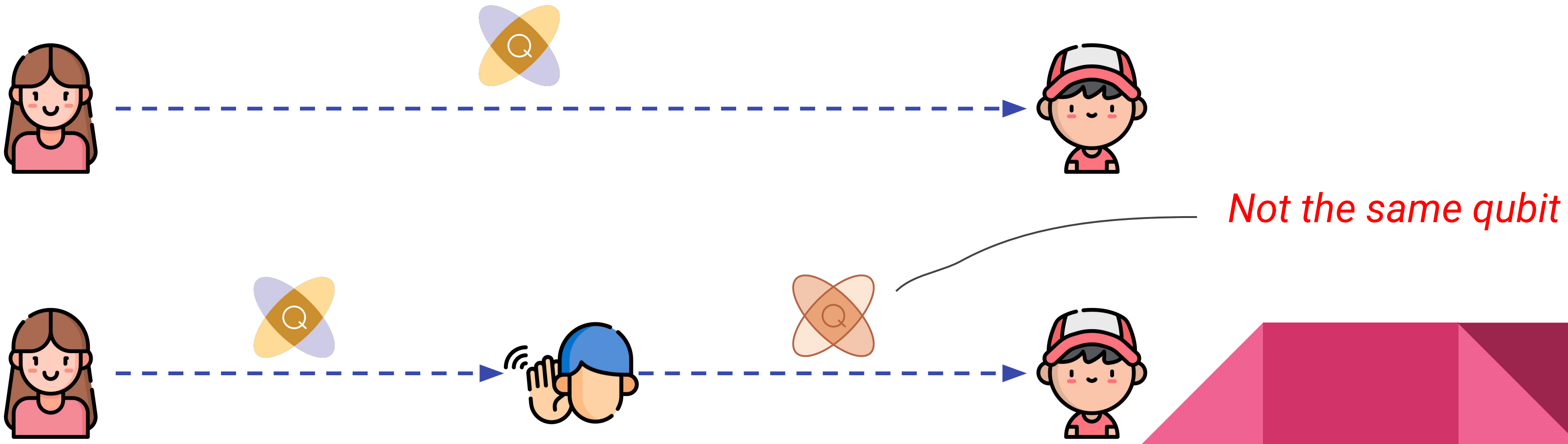
What if we insert a 45° polarizer in the middle?



The measurement modifies the polarization state of the photon, as photon polarization is a quantum state

Quantum encryption security: no-cloning theorem

Since the measurement modifies the quantum state, **it is impossible to create an independent and identical copy** of an arbitrary unknown quantum state.



How to detect that someone is eavesdropping the traffic (BB84)?

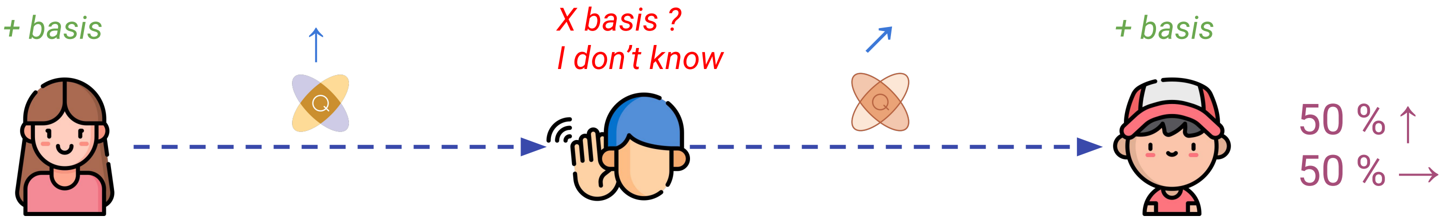
- Let’s keep the photon polarization as qubit state
- First we define 2 orthogonal basis:
  - \+ : 0 = ↑ 1 = →
  - X : 0 = ↗ 1 = ↘
- We need a quantum channel and an **authenticated clear channel**: we can use classical ciphers for encryption

How to detect that someone is eavesdropping the traffic (BB84)?

Alice's random bit	0	1	1
Alice's random sending basis	+	+	X
Sent polarization	↑	→	↘
Bob's random measuring basis	+	X	X
Measured polarization	↑	↗	↘
Basis reconciliation on public authenticated channel			
Shared bits	0	?	1

How to detect that someone is eavesdropping the traffic (BB84)?

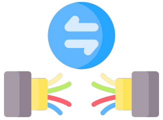
As Eve doesn't know the basis, she will change 50% of qubits



So Alice and Bob would be able to detect Eve on-the-fly by checking some random bit samples, and accordingly **abort the key exchange**

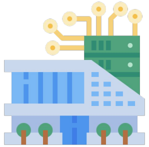
ETSI GS QKD 014 v1.1.1 standard proposal

QKD limitations



- Need a single direct fiber between the 2 sites

- Distance limitations due to fiber losses (~200 km today)



➢ More suited for cross-data-centers communication

ETSI standard representation

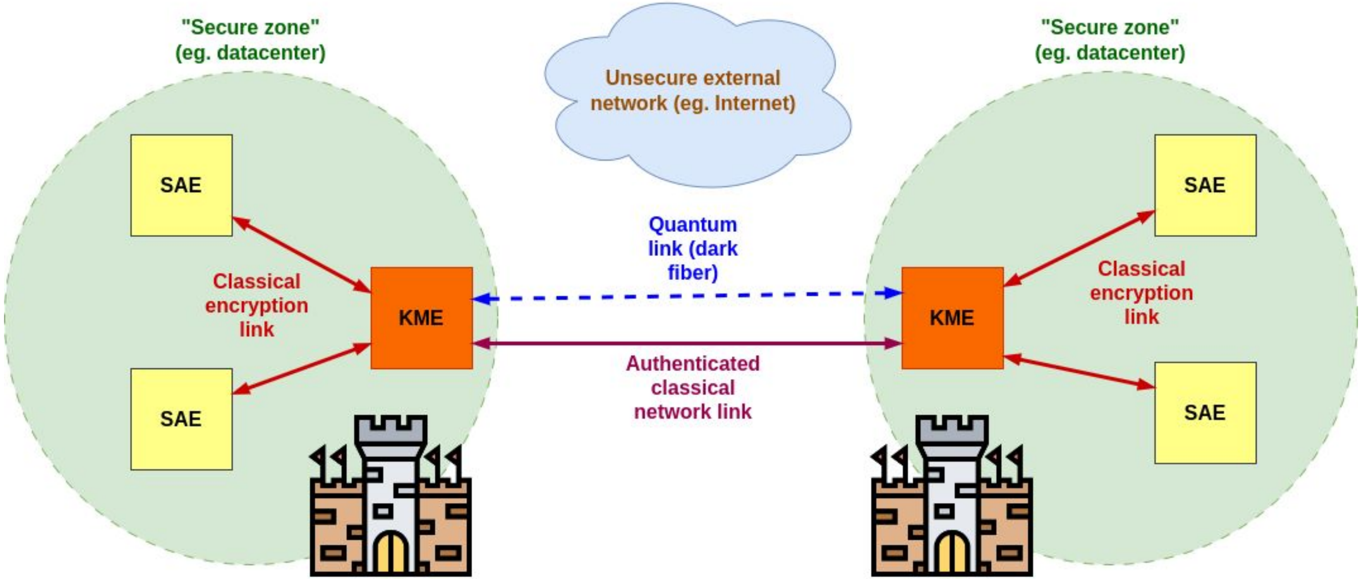
Zones

- Secure zone: inside datacenter, classical encryption is allowed
- Outside (eg Internet): We must assume that communications are eavesdropped

Entities

- KME: Key Management Entity: at least 1 / secure zone
- SAE: Secure Application Entity

ETSI standard representation



What does the standard say?

- SAEs are authenticated to KMEs via client SSL certificates
- Defines some REST routes for SAEs requests to KMEs:
  - POST /api/v1/keys/{slave SAE id}/enc_keys
  - GET /api/v1/keys/{slave SAE id}/status
  - POST /api/v1/keys/{master SAE id}/dec_keys

- And the rest is “outside the scope” (like how keys are actually exchanged between KMEs…)

Formal verification

ProVerif

ProVerif

- Takes abstract representation of a protocol and its cryptographic primitives (in the form of equations)
- Assumes Dolev-Yao model
- Translates protocol into Horn clauses
- Tries to find constraint contradiction to infer an attack

- Proven **complete** (cannot be a false negative)
- Pretty fast

ProVerif

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


process
      new my_key:key;
      event start(my_key);
      let encrypted_secret = senc(the_secret, my_key) in
      out(public_channel_1, encrypted_secret);
      in(private_channel_1, another_secret:bitstring);
      event stop(my_key);


query attacker(the_secret).
query k:key; event(stop(k)) ==> event(start(k)).
```

Verification results

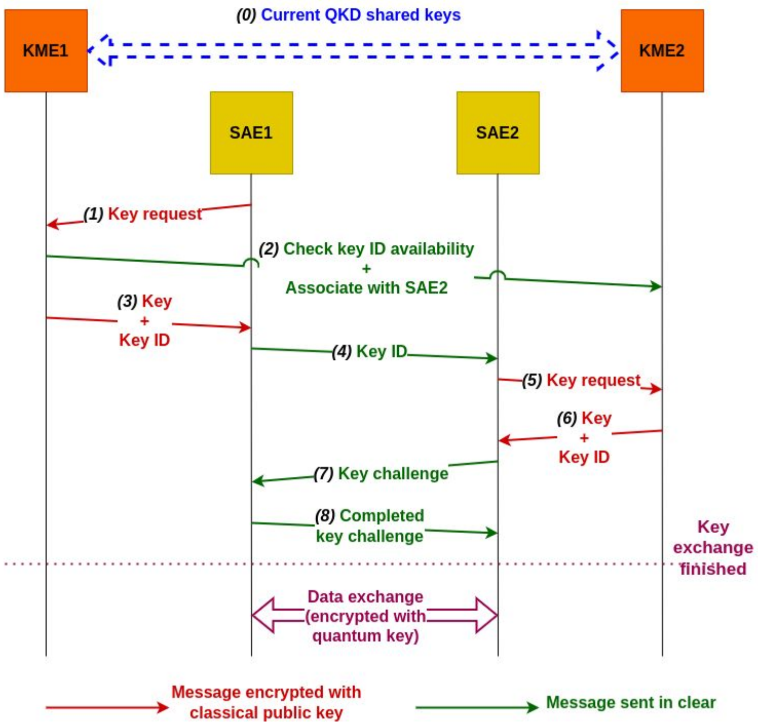
Standard appeared to be secured for both secrecy and authentication

At these conditions:

- All messages exchanged between KMEs are authenticated
- Slave (2nd) SAE must send a cryptographic challenge to master (1st) SAE to ensure proper authentication

Find the whole ProVerif code at
https://gist.github.com/thomasarmel/c2bfc851bb3b19348bf1df90ed041fac

Detailed protocol conception



Actual implementations:
https://github.com/thomasarmel/qkd_kme_server
https://github.com/thomasarmel/rustls/tree/qkd

Thanks!

Questions?