Classical Authentication in Quantum Key Distribution

Edoardo Signorini

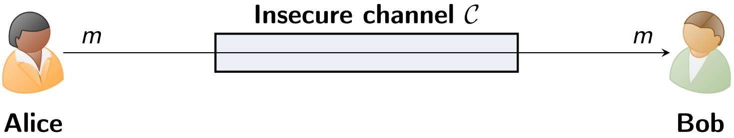
Telsy

CrypTO Conference 2021
27/05/2021

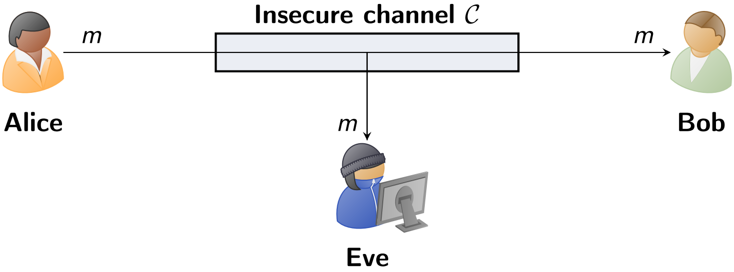
Section 1

Introduction to (ITS) authentication

Setting

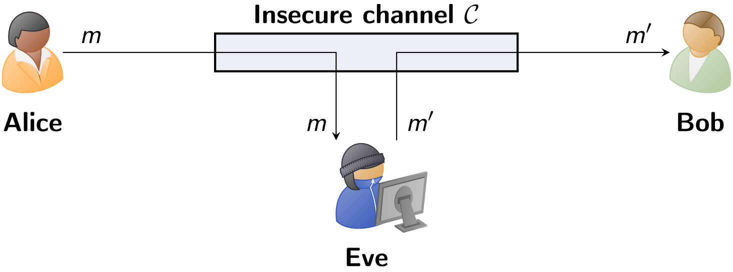


Setting



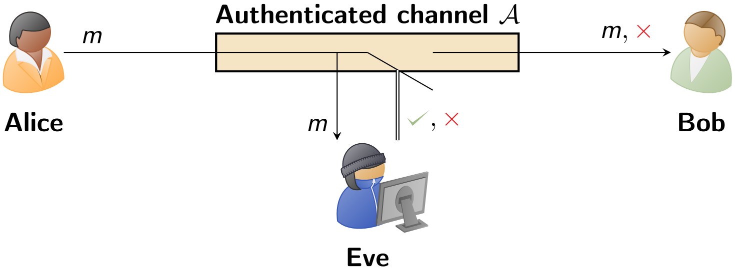
- A passive attacker can read messages on $\mathcal{C}$.

Setting



- A passive attacker can read messages on $\mathcal{C}$.
- An active attacker has complete control on $\mathcal{C}$.

Setting



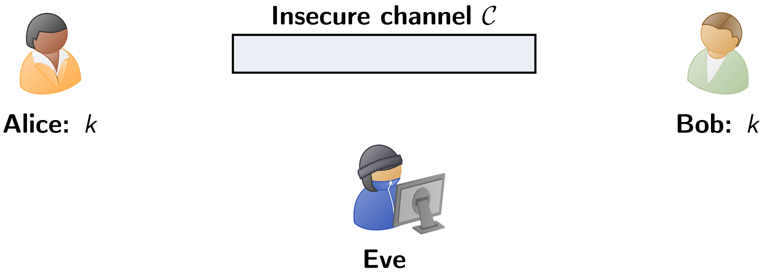
- A passive attacker can read messages on $\mathcal{C}$.
- An active attacker has complete control on $\mathcal{C}$.

We need an authenticated channel.

Message Authentication Code (MAC)

Goal: build an authenticated channel from an insecure channel and a shared secret key.

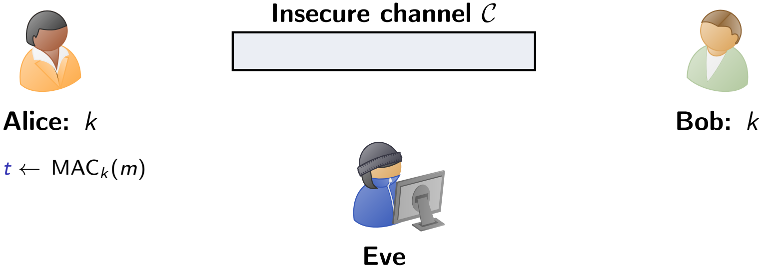
Message Authentication Code (MAC)



Goal: build an authenticated channel from an insecure channel and a shared secret key.

- Choose a tag-generation algorithm $\mathrm{MAC}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ and a verification algorithm $\mathrm{Vf}\colon \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0,1\}$.

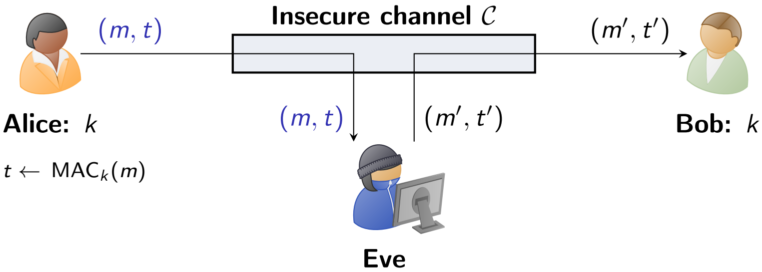
Message Authentication Code (MAC)



Goal: build an authenticated channel from an insecure channel and a shared secret key.

- Choose a tag-generation algorithm $\mathrm{MAC}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ and a verification algorithm $\mathrm{Vf}\colon \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0,1\}$.
- Given a message m and the key k a tag t is computed.

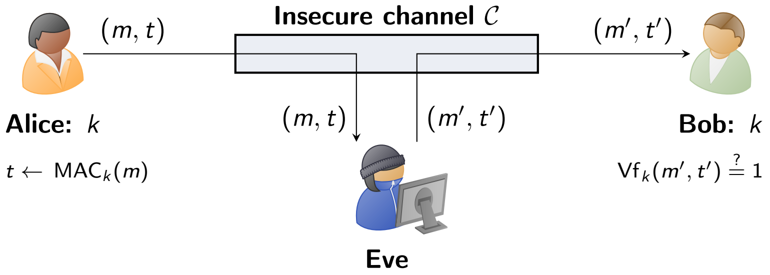
Message Authentication Code (MAC)



Goal: build an authenticated channel from an insecure channel and a shared secret key.

- Choose a tag-generation algorithm $\mathrm{MAC}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ and a verification algorithm $\mathrm{Vf}\colon \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0, 1\}$.
- Given a message m and the key k a tag t is computed.
- The couple (m, t) is sent to Bob and intercepted by Eve.

Message Authentication Code (MAC)



Goal: build an authenticated channel from an insecure channel and a shared secret key.

- Choose a tag-generation algorithm $\mathsf{MAC} \colon \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ and a verification algorithm $\mathsf{Vf} \colon \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \{0, 1\}$.
- Given a message m and the key k a tag t is computed.
- The couple (m, t) is sent to Bob and intercepted by Eve.
- Bob verifies whether the received tag t' is valid.

Real-world MACs

We require:

- Correctness: for every $k \in \mathcal{K}, m \in \mathcal{M}$, it holds $\mathsf{Vf}_k(m, \mathsf{MAC}_k(m)) = 1$.

Real-world MACs

We require:

- Correctness: for every $k \in \mathcal{K}, m \in \mathcal{M}$, it holds $\mathrm{Vf}_k(m, \mathrm{MAC}_k(m)) = 1$.
- Security: given access to an oracle of $\mathrm{MAC}_k(\cdot)$, the attacker has a negligible probability of forging a valid couple (m, t).

Real-world MACs

We require:

- Correctness: for every $k \in \mathcal{K}, m \in \mathcal{M}$, it holds $\mathrm{Vf}_k(m, \mathrm{MAC}_k(m)) = 1$.
- Security: given access to an oracle of $\mathrm{MAC}_k(\cdot)$, the attacker has a negligible probability of forging a valid couple (m, t).

Secure MACs can be built from many cryptographic primitives:

Real-world MACs

We require:

- Correctness: for every $k \in \mathcal{K}, m \in \mathcal{M}$, it holds $\mathrm{Vf}_k(m, \mathrm{MAC}_k(m)) = 1$.
- Security: given access to an oracle of $\mathrm{MAC}_k(\cdot)$, the attacker has a negligible probability of forging a valid couple (m, t).

Secure MACs can be built from many cryptographic primitives:

- From block ciphers: CBC-MAC, GMAC.
- From hash functions: HMAC, KMAC.
- From pseudorandom function families: Poly1305.

Real-world MACs

We require:

- Correctness: for every $k \in \mathcal{K}, m \in \mathcal{M}$, it holds $\mathrm{Vf}_k(m, \mathrm{MAC}_k(m)) = 1$.
- Security: given access to an oracle of $\mathrm{MAC}_k(\cdot)$, the attacker has a negligible probability of forging a valid couple (m, t).

Secure MACs can be built from many cryptographic primitives:

- From block ciphers: CBC-MAC, GMAC.
- From hash functions: HMAC, KMAC.
- From pseudorandom function families: Poly1305.

All the above constructions have computational security.

Computational vs Information-Theoretic Security

- Computational security is defined in terms of a security parameter n and the presence of a probabilistic polynomial-time (PPT) adversary.

Computational vs Information-Theoretic Security

- Computational security is defined in terms of a security parameter n and the presence of a probabilistic polynomial-time (PPT) adversary.
- A PPT adversary has a bounded computational power which is polynomial in n.

Computational vs Information-Theoretic Security

- Computational security is defined in terms of a security parameter n and the presence of a probabilistic polynomial-time (PPT) adversary.
- A PPT adversary has a bounded computational power which is polynomial in n.

A cryptographic scheme is computationally secure if any PPT adversary has negligible probability (w.r.t. n) of breaking the scheme.

Computational vs Information-Theoretic Security

- Computational security is defined in terms of a security parameter n and the presence of a probabilistic polynomial-time (PPT) adversary.
- A PPT adversary has a bounded computational power which is polynomial in n.

A cryptographic scheme is computationally secure if any PPT adversary has negligible probability (w.r.t. n) of breaking the scheme.

- Information-Theoretic security (ITS) is defined in the presence of an adversary with unlimited computational power.

Computational vs Information-Theoretic Security

- Computational security is defined in terms of a security parameter n and the presence of a probabilistic polynomial-time (PPT) adversary.
- A PPT adversary has a bounded computational power which is polynomial in n.

A cryptographic scheme is computationally secure if any PPT adversary has negligible probability (w.r.t. n) of breaking the scheme.

- Information-Theoretic security (ITS) is defined in the presence of an adversary with unlimited computational power.

A cryptographic scheme is ITS if any adversary has negligible fixed probability of breaking the scheme.

Strongly universal functions

- ITS MACs are achievable if we limit the number of messages that can be authenticated with a single key.

Strongly universal functions

- ITS MACs are achievable if we limit the number of messages that can be authenticated with a single key.
- We restrict to single message authentication.

Strongly universal functions

- ITS MACs are achievable if we limit the number of messages that can be authenticated with a single key.
- We restrict to single message authentication.
- First studied by Carter and Wegman [CW79] through the use of (almost) strongly universal functions.

Strongly universal functions

- ITS MACs are achievable if we limit the number of messages that can be authenticated with a single key.
- We restrict to single message authentication.
- First studied by Carter and Wegman [CW79] through the use of (almost) strongly universal functions.

ε-Almost Strongly Universal$_2$ (ε-ASU$_2$) functions

A family of functions $\mathcal{H} = \{h\colon \mathcal{M} \to \mathcal{T}\}$ is ε-ASU$_2$, for $\varepsilon \geq 1/|\mathcal{T}|$, if

Strongly universal functions

- ITS MACs are achievable if we limit the number of messages that can be authenticated with a single key.
- We restrict to single message authentication.
- First studied by Carter and Wegman [CW79] through the use of (almost) strongly universal functions.

ε-Almost Strongly Universal$_2$ (ε-ASU$_2$) functions

A family of functions $\mathcal{H} = \{h \colon \mathcal{M} \to \mathcal{T}\}$ is ε-ASU$_2$, for $\varepsilon \geq 1/|\mathcal{T}|$, if

1. For any $m \in \mathcal{M}$ and $t \in \mathcal{T}$

$$|\{h \in \mathcal{H} \mid h(m) = t\}| = \frac{|\mathcal{H}|}{|\mathcal{T}|}$$

Strongly universal functions

- ITS MACs are achievable if we limit the number of messages that can be authenticated with a single key.
- We restrict to single message authentication.
- First studied by Carter and Wegman [CW79] through the use of (almost) strongly universal functions.

ε-Almost Strongly Universal$_2$ (ε-ASU$_2$) functions

A family of functions $\mathcal{H} = \{h\colon \mathcal{M} \to \mathcal{T}\}$ is ε-ASU$_2$, for $\varepsilon \geq 1/|\mathcal{T}|$, if

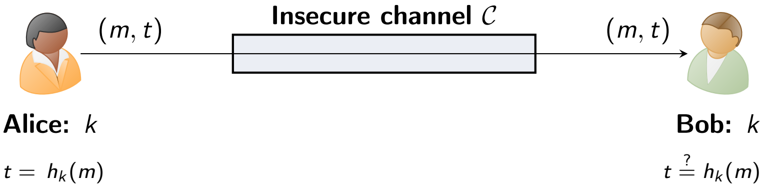
1. For any $m \in \mathcal{M}$ and $t \in \mathcal{T}$

$$|\{h \in \mathcal{H} \mid h(m) = t\}| = \frac{|\mathcal{H}|}{|\mathcal{T}|}$$

2. For any $m, m' \in \mathcal{M}$, $m \neq m'$ and $t, t' \in \mathcal{T}$

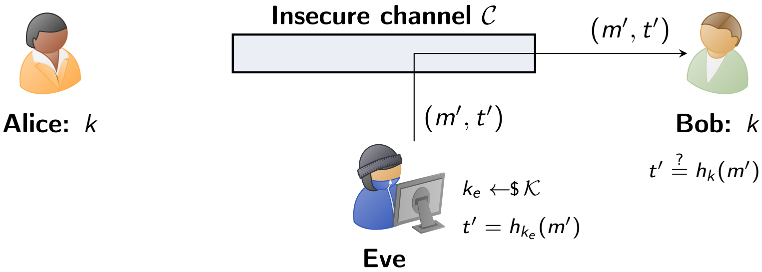
$$|\{h \in \mathcal{H} \mid h(m) = t, h(m') = t'\}| \leq \varepsilon \frac{|\mathcal{H}|}{|\mathcal{T}|}$$

One-time MAC



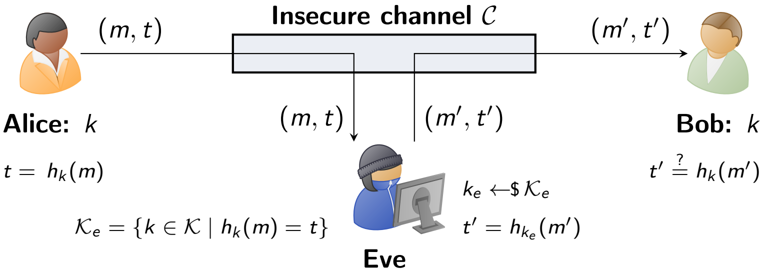
- Let $\mathcal{H} = \{h_k\}_{k \in \mathcal{K}}$, given a shared random key k, the tag t on message m is obtained as $t = h_k(m)$.

One-time MAC



- Let $\mathcal{H} = \{h_k\}_{k \in \mathcal{K}}$, given a shared random key k, the tag t on message m is obtained as $t = h_k(m)$.
- The attacker can try to:
  1. Impersonate Alice, succeeding with probability $1/|\mathcal{T}|$.

One-time MAC

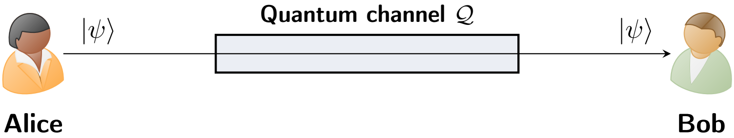


- Let $\mathcal{H} = \{h_k\}_{k \in \mathcal{K}}$, given a shared random key k, the tag t on message m is obtained as $t = h_k(m)$.
- The attacker can try to:
  1. Impersonate Alice, succeeding with probability $1/|\mathcal{T}|$.
  2. Substitute Alice, succeeding with probability at most ε.

Section 2

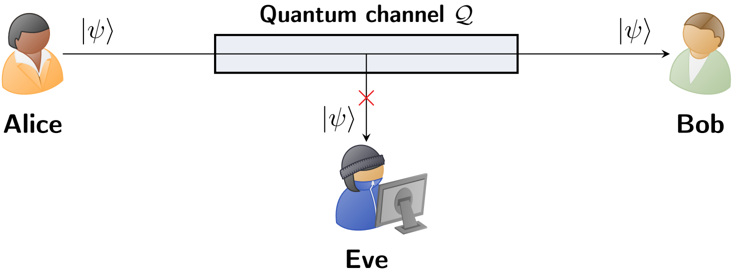
Authentication in Quantum Key Distribution

QKD in two slides I



Goal: build an ITS key exchange from a quantum channel.

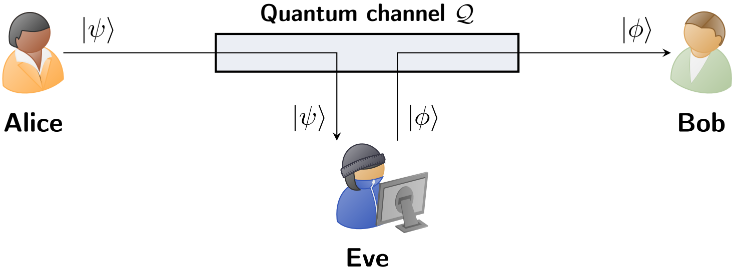
QKD in two slides I



Goal: build an ITS key exchange from a quantum channel.

- No-cloning theorem prevents a passive attacker on the quantum channel.

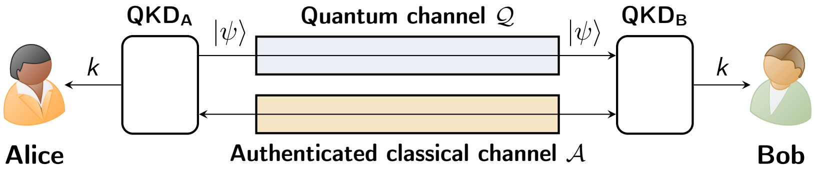
QKD in two slides I



Goal: build an ITS key exchange from a quantum channel.

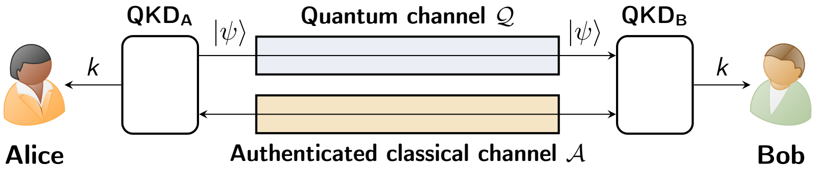
- No-cloning theorem prevents a passive attacker on the quantum channel.
- An active attacker can perform a man-in-the-middle attack.

QKD in two slides II



Quantum Key Distribution (QKD) builds an ITS key exchange from a quantum channel and an authenticated classical channel.

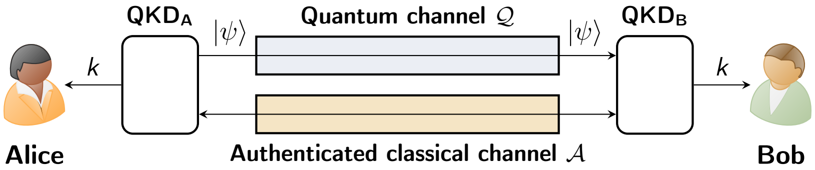
QKD in two slides II



Quantum Key Distribution (QKD) builds an ITS key exchange from a quantum channel and an authenticated classical channel.

- QKD protocols involve the use of classical authentication schemes.

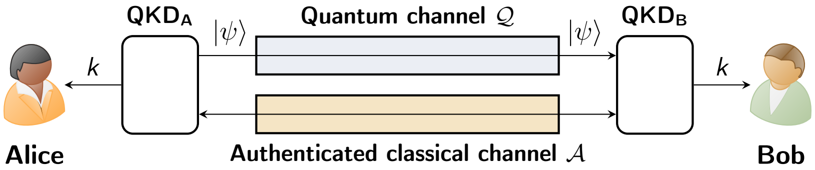
QKD in two slides II



Quantum Key Distribution (QKD) builds an ITS key exchange from a quantum channel and an authenticated classical channel.

- QKD protocols involve the use of classical authentication schemes.
- Overall unbounded security requires ITS MACs.

QKD in two slides II



Quantum Key Distribution (QKD) builds an ITS key exchange from a quantum channel and an authenticated classical channel.

- QKD protocols involve the use of classical authentication schemes.
- Overall unbounded security requires ITS MACs.
- A portion of the exchanged key can be used as the one-time authentication key for the next round.

QKD composability

Composability principle

The composition of secure cryptographic schemes should also be secure.

QKD composability

Composability principle

The composition of secure cryptographic schemes should also be secure.

- QKD produces keys that are not uniformly distributed for the attacker.

QKD composability

Composability principle

The composition of secure cryptographic schemes should also be secure.

- QKD produces keys that are not uniformly distributed for the attacker.
- ASU_2-based one-time MACs are originally formulated in terms of uniformly distributed keys.

QKD composability

Composability principle

The composition of secure cryptographic schemes should also be secure.

- QKD produces keys that are not uniformly distributed for the attacker.
- ASU_2-based one-time MACs are originally formulated in terms of uniformly distributed keys.

Both QKD [Ben+05] and ASU_2-based one-time MACs [AL14] are proved to be secure in the Universally Composable (UC) framework.

Key length

The length of the authentication key directly impacts the QKD key rate.

Key length

The length of the authentication key directly impacts the QKD key rate.

- The optimal case for $\varepsilon = 1/|\mathcal{T}|$ is impractical.

Key length

The length of the authentication key directly impacts the QKD key rate.

- The optimal case for $\varepsilon = 1/|\mathcal{T}|$ is impractical.

Let $\mathcal{H} = \{h_k : \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ be ε-ASU$_2$, $\varepsilon > 1/|T|$. If $|\mathcal{M}| \gg |\mathcal{T}|$, then

$$\log|\mathcal{K}| \geq 2\log(|\mathcal{T}| - 1) - \log(\varepsilon|\mathcal{T}| - 1)$$

Key length

The length of the authentication key directly impacts the QKD key rate.

- The optimal case for $\varepsilon = 1/|\mathcal{T}|$ is impractical.

Let $\mathcal{H} = \{h_k : \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ be ε-ASU$_2$, $\varepsilon > 1/|T|$. If $|\mathcal{M}| \gg |\mathcal{T}|$, then

$$\log|\mathcal{K}| \geq 2\log(|\mathcal{T}| - 1) - \log(\varepsilon|\mathcal{T}| - 1)$$

- In many constructions both ε and $|\mathcal{K}|$ depends on $|\mathcal{M}|$.

Key length

The length of the authentication key directly impacts the QKD key rate.

- The optimal case for $\varepsilon = 1/|\mathcal{T}|$ is impractical.

Let $\mathcal{H} = \{h_k : \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ be ε-ASU$_2$, $\varepsilon > 1/|T|$. If $|\mathcal{M}| \gg |\mathcal{T}|$, then

$$\log|\mathcal{K}| \geq 2\log(|\mathcal{T}| - 1) - \log(\varepsilon|\mathcal{T}| - 1)$$

- In many constructions both ε and $|\mathcal{K}|$ depends on $|\mathcal{M}|$.

Idea [WC81]: recycle part of the key.

Key recycling

Let $\mathcal{T} = (\mathbb{F}_2)^t$ and let $\mathcal{H} = \{h_k : \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ be ε-ASU$_2$. Then

$$\{g_{k_1,k_2}(\cdot) := h_{k_1}(\cdot) \oplus k_2 \mid (k_1, k_2) \in \mathcal{K} \times \mathcal{T}\}$$

is ε-ASU$_2$.

Key recycling

Let $\mathcal{T} = (\mathbb{F}_2)^t$ and let $\mathcal{H} = \{h_k \colon \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ be ε-ASU$_2$. Then

$$\{g_{k_1,k_2}(\cdot) := h_{k_1}(\cdot) \oplus k_2 \mid (k_1, k_2) \in \mathcal{K} \times \mathcal{T}\}$$

is ε-ASU$_2$.

- The tag for message m is obtained as $h_{k_1}(m) \oplus k_2$.

Key recycling

Let $\mathcal{T} = (\mathbb{F}_2)^t$ and let $\mathcal{H} = \{h_k \colon \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ be ε-ASU$_2$. Then

$$\{g_{k_1,k_2}(\cdot) := h_{k_1}(\cdot) \oplus k_2 \mid (k_1, k_2) \in \mathcal{K} \times \mathcal{T}\}$$

is ε-ASU$_2$.

- The tag for message m is obtained as $h_{k_1}(m) \oplus k_2$.
- The OTP key k_2 "hides" the value $h_{k_1}(m)$.

Key recycling

Let $\mathcal{T} = (\mathbb{F}_2)^t$ and let $\mathcal{H} = \{h_k : \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ be ε-ASU$_2$. Then

$$\{g_{k_1,k_2}(\cdot) := h_{k_1}(\cdot) \oplus k_2 \mid (k_1, k_2) \in \mathcal{K} \times \mathcal{T}\}$$

is ε-ASU$_2$.

- The tag for message m is obtained as $h_{k_1}(m) \oplus k_2$.
- The OTP key k_2 "hides" the value $h_{k_1}(m)$.
- k_1 can be recycled in subsequent authentication rounds.

Key recycling

Let $\mathcal{T} = (\mathbb{F}_2)^t$ and let $\mathcal{H} = \{h_k : \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ be ε-ASU$_2$. Then

$$\{g_{k_1,k_2}(\cdot) := h_{k_1}(\cdot) \oplus k_2 \mid (k_1, k_2) \in \mathcal{K} \times \mathcal{T}\}$$

is ε-ASU$_2$.

- The tag for message m is obtained as $h_{k_1}(m) \oplus k_2$.
- The OTP key k_2 "hides" the value $h_{k_1}(m)$.
- k_1 can be recycled in subsequent authentication rounds.

Key length consumption is asymptotically the optimal value $\log|\mathcal{T}| = t$.

Key recycling

Let $\mathcal{T} = (\mathbb{F}_2)^t$ and let $\mathcal{H} = \{h_k : \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ be ε-ASU$_2$. Then

$$\{g_{k_1,k_2}(\cdot) := h_{k_1}(\cdot) \oplus k_2 \mid (k_1, k_2) \in \mathcal{K} \times \mathcal{T}\}$$

is ε-ASU$_2$.

- The tag for message m is obtained as $h_{k_1}(m) \oplus k_2$.
- The OTP key k_2 “hides” the value $h_{k_1}(m)$.
- k_1 can be recycled in subsequent authentication rounds.

Key length consumption is asymptotically the optimal value $\log|\mathcal{T}| = t$.

Previous results on composability do not apply directly to this scheme.

Conclusions

- "Classical" information theory and cryptography are fundamental components of QKD.

Conclusions

- "Classical" information theory and cryptography are fundamental components of QKD.
- QKD literature often underestimates the role of authentication.

Conclusions

- "Classical" information theory and cryptography are fundamental components of QKD.
- QKD literature often underestimates the role of authentication.
- Non-definitive results on the authentication method based on key recycling.

Conclusions

- “Classical” information theory and cryptography are fundamental components of QKD.
- QKD literature often underestimates the role of authentication.
- Non-definitive results on the authentication method based on key recycling.
- Risk of security gap between theoretical model and practical realization.

References

[AL14] Aysajan Abidin and Jan-Åke Larsson. "Direct Proof of Security of Wegman–Carter Authentication with Partially Known Key". In: *Quantum Information Processing* 13.10 (2014), pp. 2155–2170.

[Ben+05] Michael Ben-Or, Michał Horodecki, Debbie W. Leung, Dominic Mayers, and Jonathan Oppenheim. "The Universal Composable Security of Quantum Key Distribution". In: *Theory of Cryptography*. Vol. 3378. 2005, pp. 386–406.

[CW79] J.Lawrence Carter and Mark N. Wegman. "Universal Classes of Hash Functions". In: *Journal of Computer and System Sciences* 18.2 (1979), pp. 143–154.

[WC81] Mark N. Wegman and J.Lawrence Carter. "New Hash Functions and Their Use in Authentication and Set Equality". In: *Journal of Computer and System Sciences* 22.3 (1981), pp. 265–279.

Thanks for your attention

edoardo.signorini@telsy.it