Employing quantum cryptography for providing Byzantine fault-tolerance

2-nd International workshop "Mathematical Methods in the Problems of Quantum Technologies"

Moscow, 26 Nov 2018

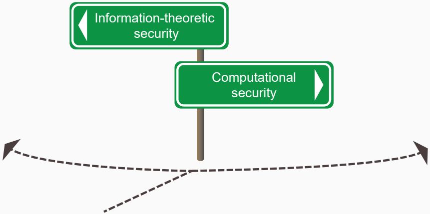
Evgeniy O. Kiktenko[1,2]

In collaboration with: Andrey A. Koziy,[2] and Aleksey K. Fedorov[2]

[1] Steklov Mathematical Institute of Russian Academy of Sciences
[2] Russian Quantum Center

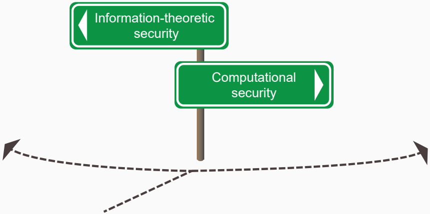
Introduction

There are two main approaches to protection against "quantum threat"



Introduction

There are two main approaches to protection against “quantum threat”

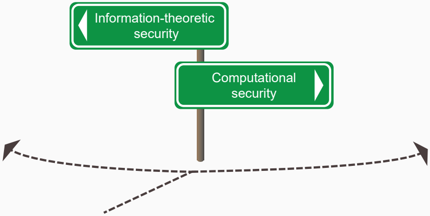


Provided with QKD

- Security proofs are available
- Expensive hardware required
- No public key crypto

Introduction

There are two main approaches to protection against "quantum threat"



Provided with QKD

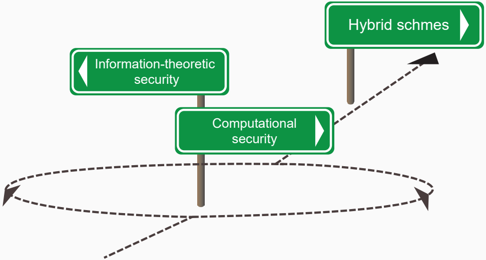
- Security proofs are available
- Expensive hardware required
- No public key crypto

Provided with PQC

- No security proofs yet
- No need in new hardware
- Public key crypto is available

Introduction

There are two main approaches to protection against "quantum threat"



Provided with QKD

- Security proofs are available
- Expensive hardware required
- No public key crypto

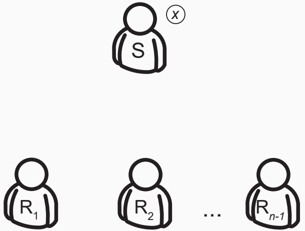
Provided with PQC

- No security proofs yet
- No need in new hardware
- Public key crypto is available

Here we consider these two approaches in the framework of providing Byzantine fault-tolerance, and show how they can be combined together in hybrid scheme in order to get benefits from both of them.

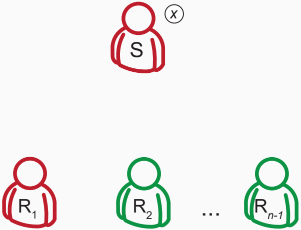
Byzantine agreement problem

[L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. 4 382 (1982)]



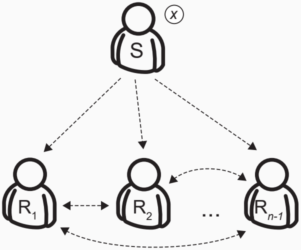
Byzantine agreement problem

[L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. 4 382 (1982)]



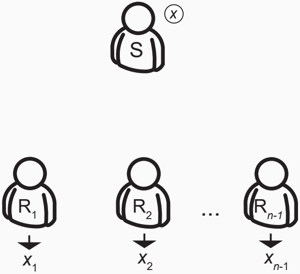
Byzantine agreement problem

[L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. 4 382 (1982)]



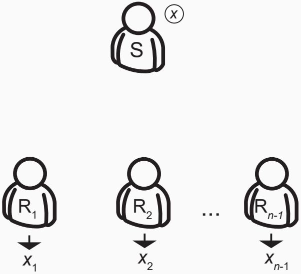
Byzantine agreement problem

[L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. 4 382 (1982)]



Byzantine agreement problem

[L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. 4 382 (1982)]



Required properties

A1. All honest receivers R_i decide the same output value $x_i = \overline{x}$ (consistency).

A2. If the sender is honest then all honest receivers R_i agree on sender's value $\overline{x} = x$ (validity).

Information-theoretic treatment

- Basic assumption: all players are connected with pair-wise authentic channels.

Information-theoretic treatment

- Basic assumption: all players are connected with pair-wise authentic channels.
- It was proven in [L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. **4** 382 (1982)] that the protocol can constructed only if

$$n \geq 3m + 1,$$

where m is maximal number of faulty nodes.

Information-theoretic treatment

- Basic assumption: all players are connected with pair-wise authentic channels.
- It was proven in [L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. **4** 382 (1982)] that the protocol can constructed only if

$$n \geq 3m + 1,$$

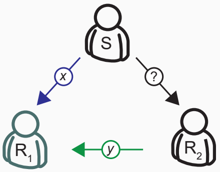
  where m is maximal number of faulty nodes.
- It is impossible to achieve broadcast even for 3 players where one is cheating.

Information-theoretic treatment

- Basic assumption: all players are connected with pair-wise authentic channels.
- It was proven in [L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. **4** 382 (1982)] that the protocol can constructed only if

$$n \geq 3m + 1,$$

  where m is maximal number of faulty nodes.
- It is impossible to achieve broadcast even for 3 players where one is cheating.

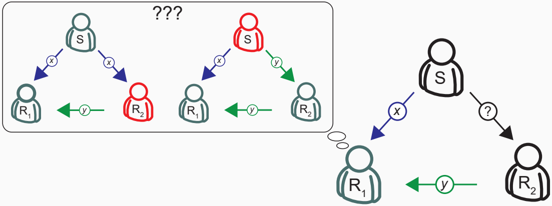


Information-theoretic treatment

- Basic assumption: all players are connected with pair-wise authentic channels.
- It was proven in [L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. **4** 382 (1982)] that the protocol can constructed only if

$$n \geq 3m + 1,$$

  where m is maximal number of faulty nodes.
- It is impossible to achieve broadcast even for 3 players where one is cheating.



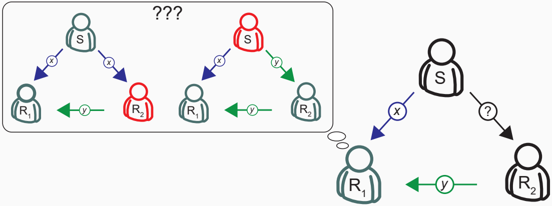
Information-theoretic treatment

- Basic assumption: all players are connected with pair-wise authentic channels.
- It was proven in [L. Lamport, R. Shostak, and M. Pease, ACM T. Progr. Lang. Sys. **4** 382 (1982)] that the protocol can constructed only if

$$n \geq 3m + 1,$$

where m is maximal number of faulty nodes.

- It is impossible to achieve broadcast even for 3 players where one is cheating.



- ITS pair-wise authentication is possible with QKD.

Quantum-secured blockchain

[E.O.K., N.O. Pozhar, M.N. Anufriev, A.S. Trushechkin, R.R. Yunusov, Y.V. Kurochkin, A.I. Lvovsky, and A.K. Fedorov, Quantum Sci. Technol. 3, 035004 (2018)]

Basic points

Quantum-secured blockchain

[E.O.K., N.O. Pozhar, M.N. Anufriev, A.S. Trushechkin, R.R. Yunusov, Y.V. Kurochkin, A.I. Lvovsky, and A.K. Fedorov, Quantum Sci. Technol. 3, 035004 (2018)]

Basic points

- ITS authentication with SU_2 family (Toeplitz hashing) and symmetric keys provided by QKD.

Quantum-secured blockchain

[E.O.K., N.O. Pozhar, M.N. Anufriev, A.S. Trushechkin, R.R. Yunusov, Y.V. Kurochkin, A.I. Lvovsky, and A.K. Fedorov, Quantum Sci. Technol. 3, 035004 (2018)]

Basic points

- ITS authentication with SU_2 family (Toeplitz hashing) and symmetric keys provided by QKD.
- ITS broadcast protocol based on pair-wise authentic channels only.

Quantum-secured blockchain

[E.O.K., N.O. Pozhar, M.N. Anufriev, A.S. Trushechkin, R.R. Yunusov, Y.V. Kurochkin, A.I. Lvovsky, and A.K. Fedorov, Quantum Sci. Technol. 3, 035004 (2018)]

Basic points

- ITS authentication with SU_2 family (Toeplitz hashing) and symmetric keys provided by QKD.
- ITS broadcast protocol based on pair-wise authentic channels only.
- No signature schemes are used.

Quantum-secured blockchain

[E.O.K., N.O. Pozhar, M.N. Anufriev, A.S. Trushechkin, R.R. Yunusov, Y.V. Kurochkin, A.I. Lvovsky, and A.K. Fedorov, Quantum Sci. Technol. 3, 035004 (2018)]

Basic points

- ITS authentication with SU_2 family (Toeplitz hashing) and symmetric keys provided by QKD.
- ITS broadcast protocol based on pair-wise authentic channels only.
- No signature schemes are used.
- The broadcast protocol is launching for unconfirmed transaction appeared during fixed period of time.

Quantum-secured blockchain

[E.O.K., N.O. Pozhar, M.N. Anufriev, A.S. Trushechkin, R.R. Yunusov, Y.V. Kurochkin, A.I. Lvovsky, and A.K. Fedorov, Quantum Sci. Technol. 3, 035004 (2018)]

Basic points

- ITS authentication with SU_2 family (Toeplitz hashing) and symmetric keys provided by QKD.
- ITS broadcast protocol based on pair-wise authentic channels only.
- No signature schemes are used.
- The broadcast protocol is launching for unconfirmed transaction appeared during fixed period of time.
- The "block" with newly confirmed transactions is constructed for all users simultaneously.

Some technical details

[E.O.K., N.O. Pozhar, M.N. Anufriev, A.S. Trushechkin, R.R. Yunusov, Y.V. Kurochkin, A.I. Lvovsky, and A.K. Fedorov, Quantum Sci. Technol. 3, 035004 (2018)]

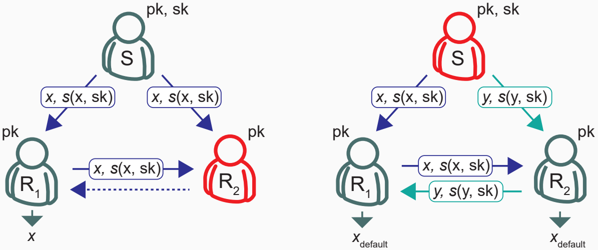
Number of nodes in the network	$n = 4$
Upper bound on the number of faulty nodes	$m = 1$
Number of rounds in the broadcast protocol	2
Duration of broadcast protocol	$<$ 10 sec
Time between block generation events	5 min
Authentication hash length	40 bit
Quantum key consumption in the initial broadcast of a transaction	40 bit
Quantum key consumption in the broadcast protocol	80 bit
Average quantum key consumption required for a transaction rate of 10 per minute	$<$ 7 bit/s

Employing signatures

The restriction on the number of faulty nodes ($n \geq 3m + 1$) can be overcomed by using signatures schemes.

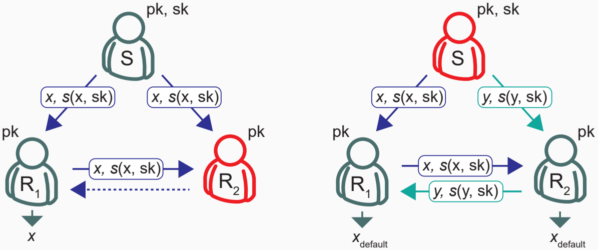
Employing signatures

The restriction on the number of faulty nodes ($n \geq 3m + 1$) can be overcomed by using signatures schemes.



Employing signatures

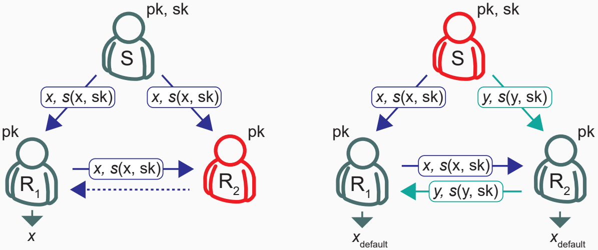
The restriction on the number of faulty nodes ($n \geq 3m + 1$) can be overcomed by using signatures schemes.



- Pre-broadcast step for establishing public key infrastructure is required.

Employing signatures

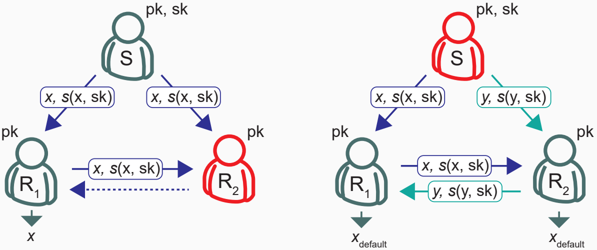
The restriction on the number of faulty nodes ($n \geq 3m + 1$) can be overcomed by using signatures schemes.



- Pre-broadcast step for establishing public key infrastructure is required.
- Can be established using common or post-quantum CS algorithms or ITS schemes (e.g. see [B. Pfitzmann, M. Waidner Reserch reprot (#908RZ 2882 (#90830)) (1996)]).

Employing signatures

The restriction on the number of faulty nodes ($n \geq 3m + 1$) can be overcomed by using signatures schemes.



- Pre-broadcast step for establishing public key infrastructure is required.
- Can be established using common or post-quantum CS algorithms or ITS schemes (e.g. see [B. Pfitzmann, M. Waidner Reserch reprot (#908RZ 2882 (#90830)) (1996)]).
- Of particular interest are the post-quantum hash-based signatures.

Lamport one-time signature

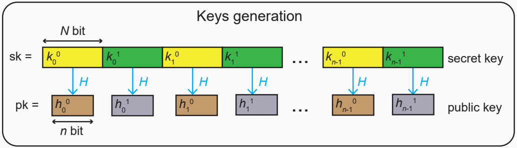
[L. Lamport, Technical Report SRI-CSL-98, SRI International Computer Science Laboratory (1979)]

Let $H : \{0, 1\}^* \to \{0, 1\}^n$ be a cryptographic hash function. Consider a following variation of L-OTS.

Lamport one-time signature

[L. Lamport, Technical Report SRI-CSL-98, SRI International Computer Science Laboratory (1979)]

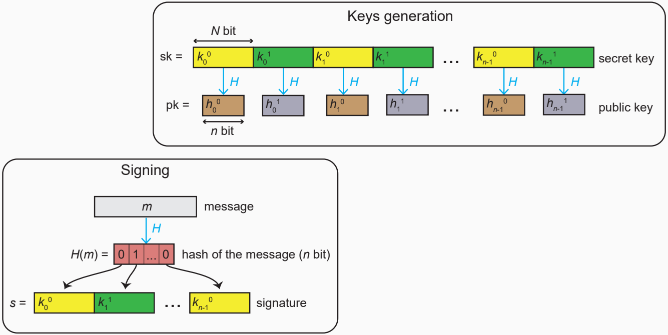
Let $H : \{0,1\}^* \to \{0,1\}^n$ be a cryptographic hash function. Consider a following variation of L-OTS.



Lamport one-time signature

[L. Lamport, Technical Report SRI-CSL-98, SRI International Computer Science Laboratory (1979)]

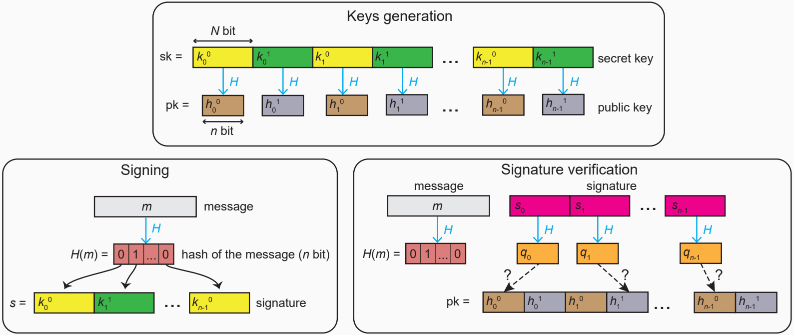
Let $H : \{0,1\}^* \to \{0,1\}^n$ be a cryptographic hash function. Consider a following variation of L-OTS.



Lamport one-time signature

[L. Lamport, Technical Report SRI-CSL-98, SRI International Computer Science Laboratory (1979)]

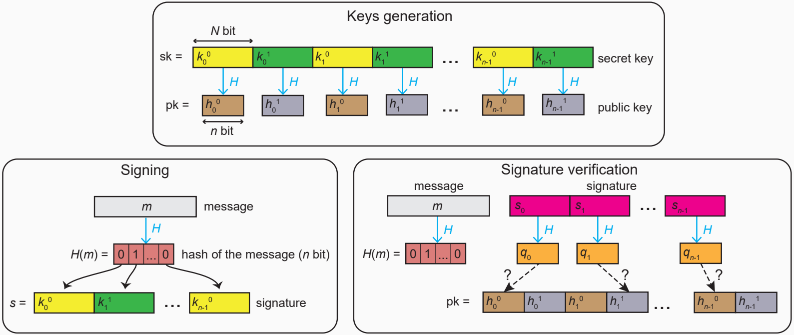
Let $H : \{0,1\}^* \to \{0,1\}^n$ be a cryptographic hash function. Consider a following variation of L-OTS.



Lamport one-time signature

[L. Lamport, Technical Report SRI-CSL-98, SRI International Computer Science Laboratory (1979)]

Let $H : \{0,1\}^* \to \{0,1\}^n$ be a cryptographic hash function. Consider a following variation of L-OTS.



Note: signature includes a half of secret key!

Security of hash-based signatures

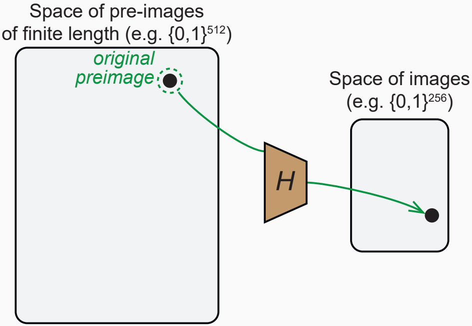
- It is *commonly accepted* that the best one can do with quantum computer is to employ Grover's algorithm (quadratic speed-up) to break the security of the scheme.

Security of hash-based signatures

- It is *commonly accepted* that the best one can do with quantum computer is to employ Grover's algorithm (quadratic speed-up) to break the security of the scheme.
- However, let's imagine that an adversary founded a way how to invert the employed hash function in some clever way OR got an access to some enormous computational resources.

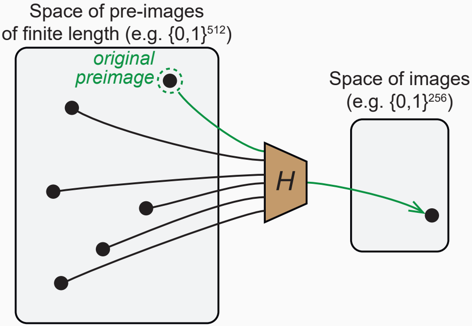
Security of hash-based signatures

- It is *commonly accepted* that the best one can do with quantum computer is to employ Grover's algorithm (quadratic speed-up) to break the security of the scheme.
- However, let's imagine that an adversary founded a way how to invert the employed hash function in some clever way OR got an access to some enormous computational resources.



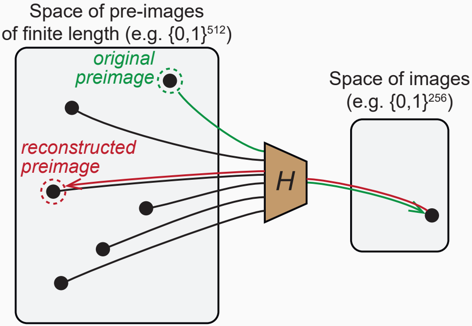
Security of hash-based signatures

- It is *commonly accepted* that the best one can do with quantum computer is to employ Grover's algorithm (quadratic speed-up) to break the security of the scheme.
- However, let's imagine that an adversary founded a way how to invert the employed hash function in some clever way OR got an access to some enormous computational resources.



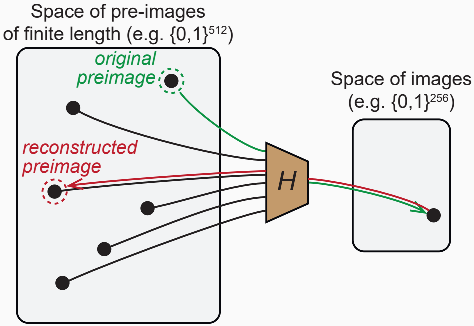
Security of hash-based signatures

- It is *commonly accepted* that the best one can do with quantum computer is to employ Grover's algorithm (quadratic speed-up) to break the security of the scheme.
- However, let's imagine that an adversary founded a way how to invert the employed hash function in some clever way OR got an access to some enormous computational resources.



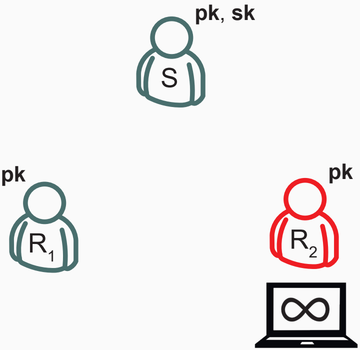
Security of hash-based signatures

- It is *commonly accepted* that the best one can do with quantum computer is to employ Grover's algorithm (quadratic speed-up) to break the security of the scheme.
- However, let's imagine that an adversary founded a way how to invert the employed hash function in some clever way OR got an access to some enormous computational resources.

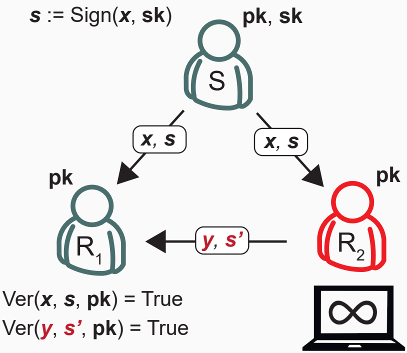


- We can use a resulting collision as an evidence of a forgery event.

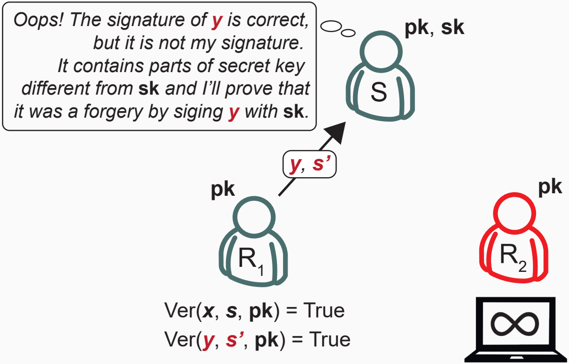
Proving the fact of a forgery



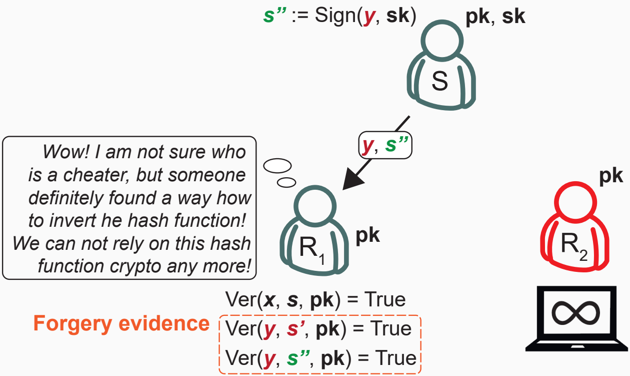
Proving the fact of a forgery



Proving the fact of a forgery



Proving the fact of a forgery



Broadcast with detection of signature forgery

- Before a start of the protocol each of the players initialize a flag $\texttt{forgery\_detected}_i := 0$.

Broadcast with detection of signature forgery

- Before a start of the protocol each of the players initialize a flag $\texttt{forgery\_detected}_i := 0$.

Required properties

B1. If no one has an ability to forge anyones signature, then the standard broadcast Byzantine agreement properties (consistency and validity) hold, and all the honest players end protocol with $\texttt{forgery\_detected}_i = 0$.

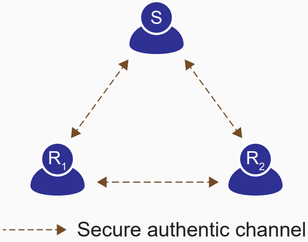
B2. If anyone applies the ability to forge signature, then all the honest players end up the protocol with flags $\texttt{forgery\_detected}_i = 1$.

Sketch of the protocol for tripartite case

Main ideas

1. Using hash-based signatures (PQC) + ITS authentication (provided with QKD);
2. Making a check if there is suspicion of a forgery.
3. Using ITS (pseudo-)signatures (provided with QKD) for broadcasting the evidence.

Pre-broadcast stage: establishing PKI and keys for ITS signatures.

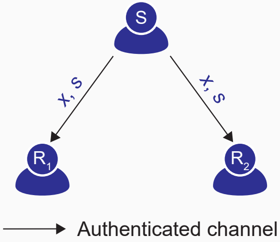


Sketch of the protocol for tripartite case

Main ideas

1. Using hash-based signatures (PQC) + ITS authentication (provided with QKD);
2. Making a check if there is suspicion of a forgery.
3. Using ITS (pseudo-)signatures (provided with QKD) for broadcasting the evidence.

Step 1: initial sending of the message by Sender.

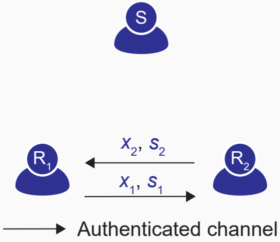


Sketch of the protocol for tripartite case

Main ideas

1. Using hash-based signatures (PQC) + ITS authentication (provided with QKD);
2. Making a check if there is suspicion of a forgery.
3. Using ITS (pseudo-)signatures (provided with QKD) for broadcasting the evidence.

Step 2: exchanging messages by Receivers.

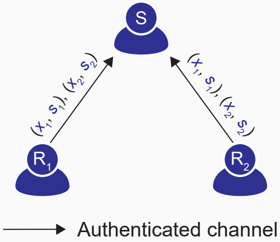


Sketch of the protocol for tripartite case

Main ideas

1. Using hash-based signatures (PQC) + ITS authentication (provided with QKD);
2. Making a check if there is suspicion of a forgery.
3. Using ITS (pseudo-)signatures (provided with QKD) for broadcasting the evidence.

Step 3: asking for clarifications (if needed).

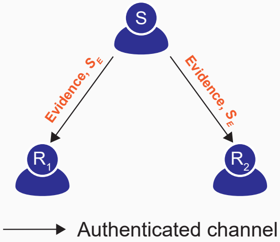


Sketch of the protocol for tripartite case

Main ideas

1. Using hash-based signatures (PQC) + ITS authentication (provided with QKD);
2. Making a check if there is suspicion of a forgery.
3. Using ITS (pseudo-)signatures (provided with QKD) for broadcasting the evidence.

Step 4: sending the evidence of forgery (if available).

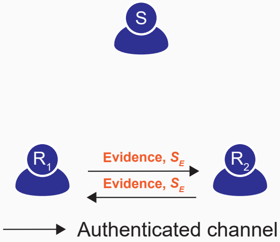


Sketch of the protocol for tripartite case

Main ideas

1. Using hash-based signatures (PQC) + ITS authentication (provided with QKD);
2. Making a check if there is suspicion of a forgery.
3. Using ITS (pseudo-)signatures (provided with QKD) for broadcasting the evidence.

Step 5: exchanging the evidence between Receivers (if available).

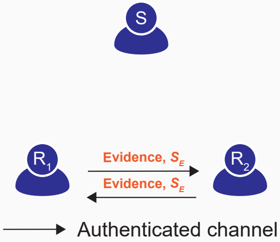


Sketch of the protocol for tripartite case

Main ideas

1. Using hash-based signatures (PQC) + ITS authentication (provided with QKD);
2. Making a check if there is suspicion of a forgery.
3. Using ITS (pseudo-)signatures (provided with QKD) for broadcasting the evidence.

Step 5: exchanging the evidence between Receivers (if available).



Details to appear on arXiv soon!

Conclusion

- There are two ways of thinking about security in post-quantum era: ITS and CS.

Conclusion

- There are two ways of thinking about security in post-quantum era: ITS and CS.
- It's possible to construct ITS distributed ledgers with QKD.

Conclusion

- There are two ways of thinking about security in post-quantum era: ITS and CS.
- It's possible to construct ITS distributed ledgers with QKD.
- Features of hash-based post-quantum signatures allows proving event of their forgery.

Conclusion

- There are two ways of thinking about security in post-quantum era: ITS and CS.
- It's possible to construct ITS distributed ledgers with QKD.
- Features of hash-based post-quantum signatures allows proving event of their forgery.
- Combining with ITS cryptographic primitives, provided with QKD, allows constructing new type of broadcast protocol with detection of signature forgery (detailed description of the protocol with its security proof to appear on arXiv soon).

Conclusion

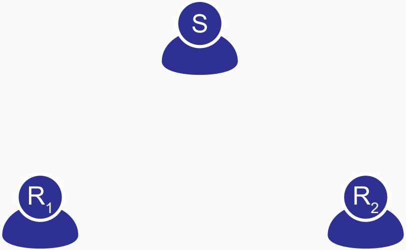
- There are two ways of thinking about security in post-quantum era: ITS and CS.
- It's possible to construct ITS distributed ledgers with QKD.
- Features of hash-based post-quantum signatures allows proving event of their forgery.
- Combining with ITS cryptographic primitives, provided with QKD, allows constructing new type of broadcast protocol with detection of signature forgery (detailed description of the protocol with its security proof to appear on arXiv soon).
- Open questions:
  - extending protocol on arbitrary number of players;
  - employing modern hash-based many-time signatures (`SPHINCS`, `XMSS`, etc.).

Thank You!
Any questions?

e.kiktenko@rqc.ru

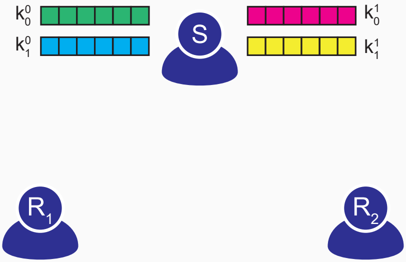
ITS signature scheme

[P. Wallden, V. Dunjko, A. Kent, E. Andersson Phys. Rev. A 91, 042304 (2014)]



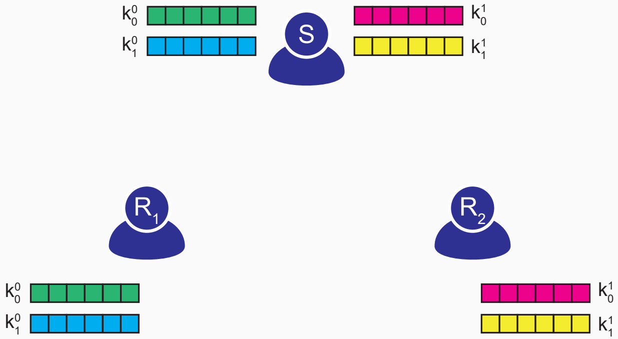
ITS signature scheme

[P. Wallden, V. Dunjko, A. Kent, E. Andersson Phys. Rev. A 91, 042304 (2014)]



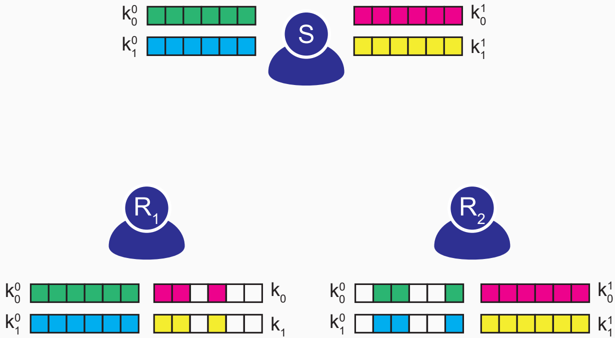
ITS signature scheme

[P. Wallden, V. Dunjko, A. Kent, E. Andersson Phys. Rev. A 91, 042304 (2014)]



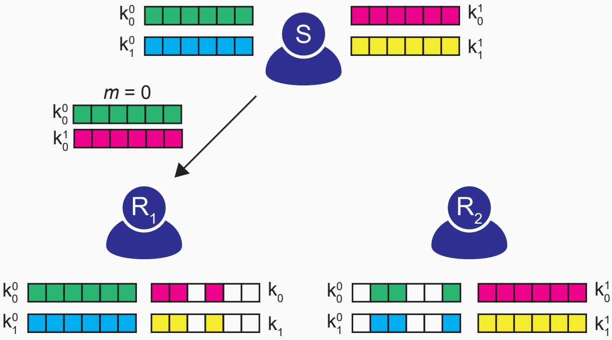
ITS signature scheme

[P. Wallden, V. Dunjko, A. Kent, E. Andersson Phys. Rev. A 91, 042304 (2014)]



ITS signature scheme

[P. Wallden, V. Dunjko, A. Kent, E. Andersson Phys. Rev. A 91, 042304 (2014)]



ITS signature scheme

[P. Wallden, V. Dunjko, A. Kent, E. Andersson Phys. Rev. A 91, 042304 (2014)]