Post-processing procedure for industrial quantum key distribution systems

arXiv:1603.08387

E.O. Kiktenko[1,2], A.S. Trushechkin[2,3], R.U. Valiev[1,4], Y.V. Kurochkin[1], and A.K. Fedorov[1,2]

[1] Russian Quantum Center, Moscow
[2] DEPHAN, Moscow
[3] Steklov Mathematical Institute, Moscow
[4] Acronis LLC, Moscow

The work is supported by Ministry of Education and Science of the Russian Federation in the framework of the Federal Program (2015-14-579-0149-006).

Group: Theoretical support of the industrial project

- Project Leader: Aleksey Fedorov (RQC & DEPHAN).
- Postdoc: Evgeny Kiktenko (BMSTU & DEPHAN).
- Group Leader: Oleg Lychkovskiy (RQC & MIAN)
- Senior Researcher: Anton Trushechkin (MIAN).
- Scientific Advisor: Alexander Holevo (MIAN).

A.K. Fedorov

RQC

E.O. Kiktenko

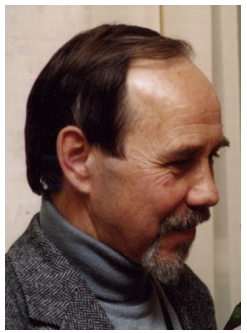
BMSTU

A.S. Lychkovksiy

MIAN

A.S. Trushechkin

MIAN



A.S. Holevo

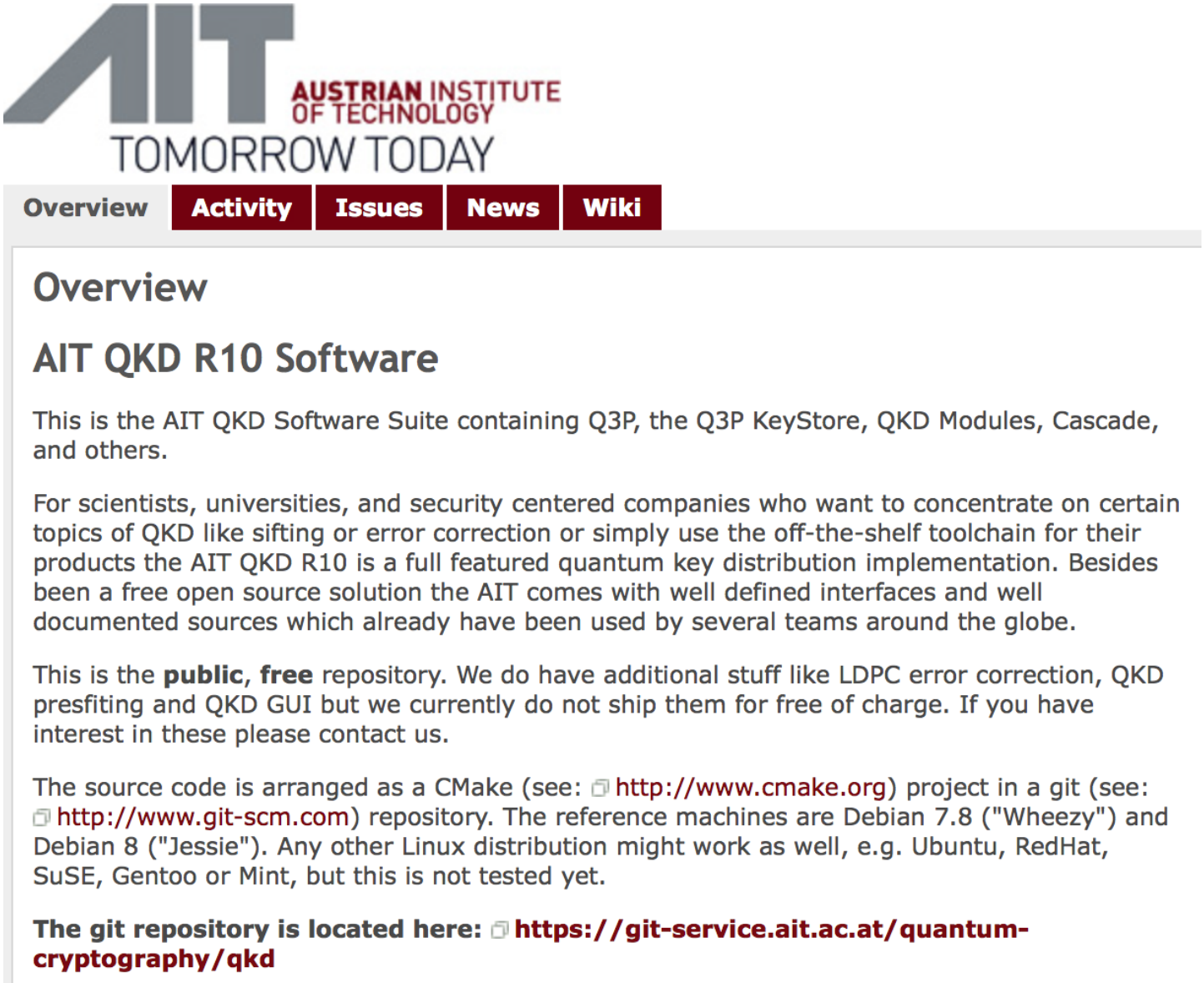
MIAN

Problem statment. I

- The industrial quantum key distribution system under development is based on the BB84 protocol.
- The BB84 protocol uses a public classical channel.
- Errors in quantum keys after distributions come from non-ideal work of technical apparatus and possible attacks.
- Typical level of errors is too high for practical applications.
- Security model relies on the fact that all errors in quantum keys comes from attacks.

Problem statement: development of a post-processing procedure, which allows one to transform quantum keys from industrial system to final keys for final applications.

World practice: AIT Austrian Institute of Technology

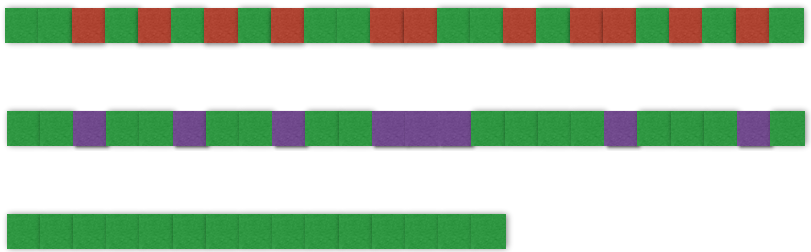


Problem statment. II

Problem statement: development of a post-processing procedure, which allows one to transform quantum keys from industrial system to final keys for final applications.

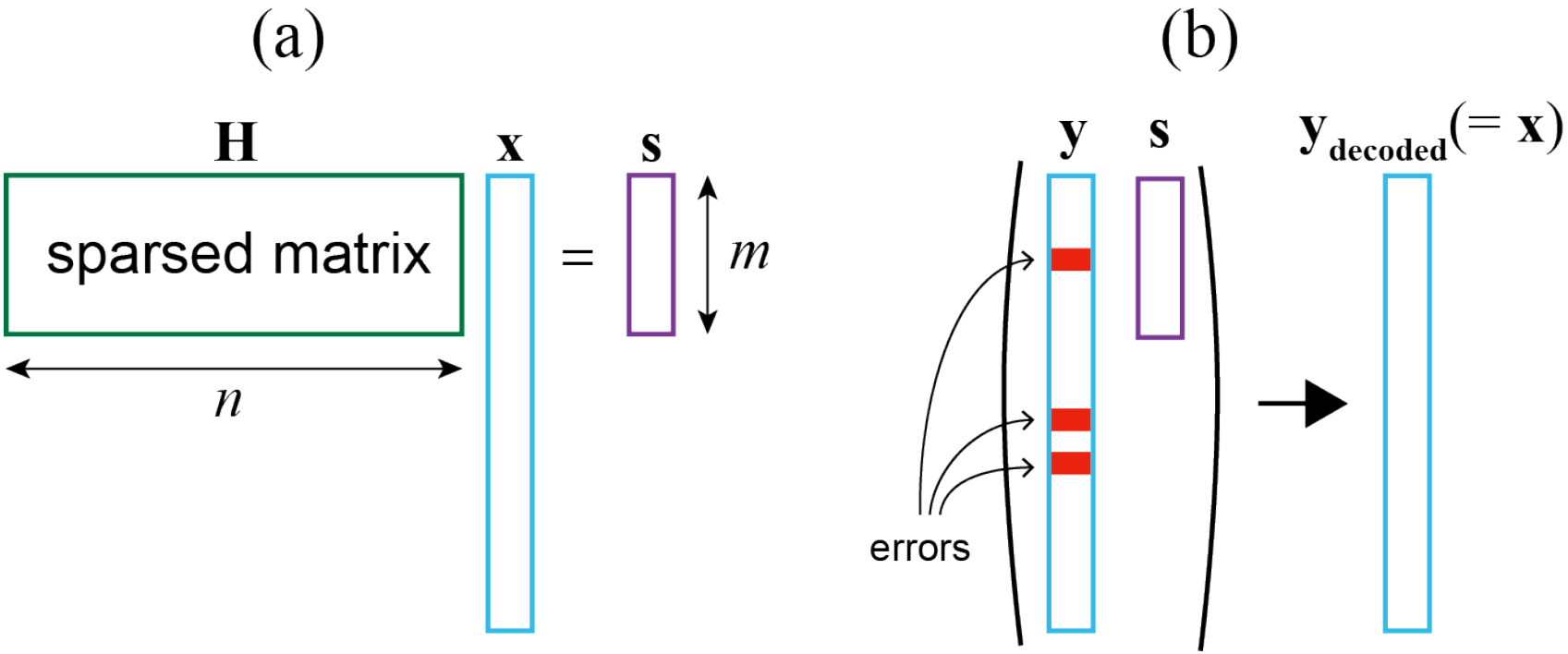
- Error correction.
- Parameter estimation.
- Privacy amplification.

Parties have to use authentication for all communications over public channel.



Error correction

Alice and Bob share a pool of LDPC parity-matrices with code-rates $R = (1 - m/n)$ from 0.9 up to 0.5 (with step being equal to 0.05) and the frame size (length of processed strings) being equal to $n = 4096$.



Error correction

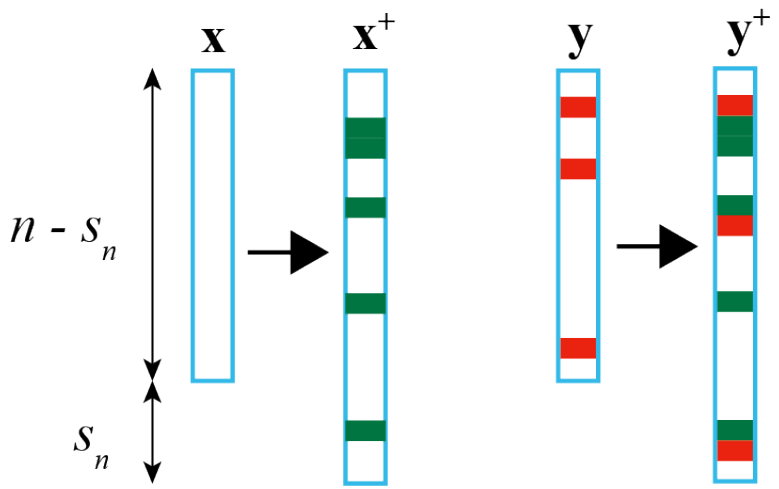
For each coding and decoding process parties employ code with the minimal rate R, which satisfies the following condition:

$$\frac{1-R}{h_{\mathrm{b}}(\mathrm{QBER}_{\mathrm{est}})} \leq f_{\mathrm{crit}} \tag{1}$$

where h_{b} is the standard binary entropy function, $\mathrm{QBER}_{\mathrm{est}}$ is the estimated level of QBER (see below), and $f_{\mathrm{crit}} = 1.22$ is the critical efficiency parameter in our setup, that is the tolerable ratio between level of disclosed information about sifted key and theoretical limit for successive error correction, which is predicted by the classical information theory.

Error correction: shortening technique

To decrease a probability of unsuccessive decoding in the constraint that the resulting efficiency is not greater than f_{crit}, Alice and Bob use the shortening technique.



The number of shortened bits n_{s} is obtained from the following expression

$$n_{\text{s}} = \lfloor n - \frac{m}{f_{\text{crit}} h_{\text{b}}(\text{QBER}_{\text{est}})} \rfloor, \qquad (2)$$

where $\lfloor x \rfloor$ stands for the maximal integer less than x.

Error correction: coding/decoding

- Alice and Bob construct $\mathcal{N} = 256$ strings of length n possessing $n - n_s$ bits of their sifted keys K^A_{sift} and K^B_{sift} and n_s shortened bits, whose positions and values come from synchronised pseudo-random generator.
- Bob multiplies the chosen parity-check matrix on the constructed strings to obtain syndromes that are sent to Alice.
- For the LDPC syndrome decoding, Alice applies iterative sum-product algorithm, which uses log-likelihood ratios for messages between symbol and parity-check nodes of the corresponding Tanner graph. Then Alice removes shortened bits obtaining corrected key.

Error correction: verification

- If the algorithm of decoding does not converge for particular block in a specified number of iterations, then this block is considered as unverified. However, rarely decoding process converges to a wrong result, i.e., to a incorrect key but still with proper syndrome.
- We use comparison of hash-tags constructed with ε-universal polynomial hashing. In particular, we use modified 50-bit variant of PolyR hash function that provide collision probability for a $\mathcal{N}$ blocks of $n - n_s$ bits on the level of $\varepsilon_{\text{ver}} < 2 \times 10^{-12}$.
- If the hash tags on the both sides match, then the corresponding blocks are concerned to be identical on the both sides and are added to verified keys K_{ver}.
- We note that K_{ver} is a part of K^B_{sift} as all modifications of the key are performed on the Alice's side.

Parameter estimation

- The purpose of the parameter estimation procedure is to determine QBER that is a probability of bit-flipping in a quantum channel. This problem could be resolved by comparison of input K^A_{sift} and output K_{ver} keys of the error correction, because as it was noted all changes of the key were performed exclusively by Alice.
- The ratio of corrected bits is averaged over a set of $\mathcal{N}$ the error correction blocks, and for unverified blocks a conservative maximal value $1/2$ is assumed. If we denote all verified blocks in the set as $\mathcal{V}$ than the estimated QBER reads

$$\text{QBER}_{\text{est}} = \mathcal{N}^{-1}\left(\sum\nolimits_{i\in\mathcal{V}} \text{QBER}_i + \left|\overline{\mathcal{V}}\right|/2\right), \tag{3}$$

where QBER_i is the ratio of bit-flips in ith block and $|\overline{\mathcal{V}}|$ is number on unverified blocks.

Privacy amplification

- After these procedures, both sides have identical bit strings.
- Eve may have some amount of information about them.
- The privacy amplification procedure is used to reduce this potential information of an adversary to a negligible quantity.
- This is achieved by a contraction of the input bit string into a shorter string.
- The output shorter string is a final private key K_{sec} of length l_{sec}.

Privacy amplification

Our algorithm of the privacy amplification firstly checks whether it is possible to distill private key with the given length and security parameter ε_{pa} (which is set to $\varepsilon_{\text{pa}} = 10^{-12}$). Namely, define the quantity

$$\nu = \sqrt{\frac{2(l_{\text{ver}} + k)(k+1)\ln\frac{1}{\varepsilon_{\text{pa}}}}{l_{\text{ver}}k^2}}, \tag{4}$$

where k in the number of bits used to estimate the QBER. Then if the inequality

$$2^{-\frac{1}{5}(l_{\text{ver}}(1-h_{\text{b}}(\delta+\nu))-r-t-l_{\text{sec}})} \leq \varepsilon_{\text{pa}} \tag{5}$$

is satisfied, then the generation of a private key of length l_{sec} and security parameter ε_{pa} is possible. Here r is the total length of the syndromes in the error correction procedure, t is the total length of the verification hashes and δ is a critical level of QBER.

Privacy amplification

- If formula (5) gives the positive answer on the question about the possibility of key generation with the desired parameters, then the final private key is computed as a hash function of the input bit string.
- The family of hash functions is required to be 2-universal. The generalization to almost universal families of hash functions is also possible.

Privacy amplification

- A hash function is chosen randomly from the Toeplitz universal family of hash functions. A matrix T of dimensionality $l_{\text{sec}} \times l_{\text{ver}}$ is a Toeplitz matrix if $T_{ij} = T_{i+1,j+1} = s_{j-i}$ for all $i = 1, \ldots, l_{\text{sec}} - 1$ and $j = 1, \ldots, l_{\text{ver}} - 1$.
- To generate a Toeplitz matrix, we need a random bit string $S = (s_{1-l_{\text{sec}}}, s_{1-l_{\text{sec}}+1}, \ldots, s_{l_{\text{ver}}-1})$ of length $l_{\text{ver}} + l_{\text{sec}} - 1$.
- The string S is generated randomly by one side (say, Alice) and sent to another side (Bob) by public channel. Let us denote the corresponding Toeplitz matrix as T_S. Then the final private key is computed as

$$K_{\text{sec}} = T_S K_{\text{ver}}$$

(with multiplication and addition modulo 2).

Privacy amplification

- It is possible to use not a random, but a pseudo-random bit string, which uses a shorter random seed, to specify the Toeplitz matrix.
- In this case, the family of hash functions is not universal, but almost universal, which is also acceptable for privacy amplification with some modifications in formula (5).
- However, neither the rate of random number generator, nor the amount of publicly amount information are critical parameters of our setup. Thus, we adopt the standard family of Topelitz functions.

Authentication

- The purpose of the authentication is to ensure that messages received by each side via public channel were sent by the other legitimate side (not by the adversary) and were not changed during the transmission.
- The hash function requires to assure that, whenever an eavesdropper does not know the private key, he cannot modify the message of send his own message and guess the correct hash tag of the message except for negligible probability (we require it not to exceed $\varepsilon_{\text{aut}} = 10^{-12}$).
- For unconditional security, the hash function must be chosen from some universal family or GOST.
- We also consider the GOST authentication method.

Workflow of the post-processing procedure. I

- Sifted keys go through the error correction that is adjusted on the current value of QBER.
- After accumulation of necessary number of blocks they input to the parameter estimation (together with their versions before the error correction).
  - If an estimated value of QBER given by (3) is higher than the critical value needed for efficient privacy amplification, the parties receive warning message about possible of eavesdropping.
  - Otherwise, verified blocks input privacy amplification and estimated QBER is used in next round of the error correction algorithm.

Workflow of the post-processing procedure. II

- The overall (in)security parameter of the quantum key distribution system is

$$\begin{aligned} \varepsilon_{\text{qkd}} &= \varepsilon_{\text{ver}} + \varepsilon_{\text{pa}} + \varepsilon_{\text{aut}} \\ &= 2 \times 10^{-11} + 10^{-12} + 10^{-12} < 3 \times 10^{-11}. \end{aligned} \tag{6}$$

  This parameter majorizes both the probability that the keys of Alice and Bob do not coincide and the probability of guessing the common key by Eve. If this parameter exceeds a critical value, then the protocol is terminated.

- After the privacy amplification procedure, a fraction of the key is used for authentication in the next rounds. In our setup, the fraction of generated private key consumed by authentication procedure does not exceed 15%.

Open questions

- Pilot project on quantum communications: a hybrid study.
- Certification of quantum key distribution devices.
- Standardisation of post-processing procedures.

Thank you for the attention!