On the use of pseudorandom quantum states in quantum cryptography

A. S. Trushechkin[1,2,3], **P. A. Tregubov**[2], **E. O. Kiktenko**[1,4], **Yu. V. Kurochkin**[4], **A. K. Fedorov**[4]
[1] *Steklov Mathematical Institute of Russian Academy of Sciences, 8 Gubkina St., Moscow 119991, Russia*
[2] *National Research Nuclear University MEPhI, 31 Kashirskoe Highway, Moscow 115409, Russia*
[3] *National University of Science and Technology MISiS, 2 Leninsky Avenue, Moscow 119049, Russia*
[4] *Russian Quantum Center, 100 Novaya St., Skolkovo, Moscow 143025, Russia*
e-mail: *trushechkin@mi.ras.ru*

The poster has two parts. In the first part, we present a QKD protocol of a new type and analyse its security against the intercept-resend attack. The specific feature of this protocol is the use of many bases and their pseudorandom choice, which is coincident by the legitimate parties (Alice and Bob). Alice and Bob have a common short key to generate a common pseudorandom sequence. This allows to avoid sifting and, hence, loosing a part of the key. We show that this protocol gives better secret key rates than the BB84 protocol and approximately the same rates as the asymmetric BB84 protocol.

Motivation for the development of a new protocol:

- Losses due to sifting in the BB84 protocol: how to avoid? One way is asymmetric BB84 protocol. Our new protocol is another way.
- Exploration of different ways of how we can exploit the properties of quantum information to provide information security. The novel idea of the proposed protocol is the combination of classical pseudorandomness with quantum encoding of information.
- Investigation of how the use of pseudorandom numbers instead of true random numbers affects the security of QKD protocols and, moreover, whether it can be even more advantageous.

Let Alice and Bob have a common secret key $k = (k_1, \dots, k_l) \in \{0,1\}^l$ (seed for the pseudorandom number generator). We use the notation $|\alpha\rangle = \cos\alpha\,|0\rangle + \sin\alpha\,|1\rangle$ with $\{|0\rangle, |1\rangle\}$ being the standard base. As usual in QKD, Alice and Bob use quantum channel and public authentic classical channel: an eavesdropper (Eve) freely read the communication over this channel, but cannot interfere in it. The first stages of the protocol are:

1. Alice and Bob generate a common pseudorandom (i.e., deterministic if k is known) sequence $\alpha_1(k), \dots, \alpha_N(k)$, where $\alpha_i(k) \in \{\frac{\pi j}{M}\}_{j=0}^{M-1}$.
2. Alice generates a random binary string $x = (x_1, \dots, x_N)$.
3. Alice sends the states $|\alpha_1(k) + \frac{\pi}{2}x_1\rangle, \dots, |\alpha_N(k) + \frac{\pi}{2}x_N\rangle$ to Bob over the quantum channel. Bob measures them in the bases $\{|\alpha_i(k)\rangle, |\alpha_i(k) + \frac{\pi}{2}\rangle\}$, $i = 1, \dots, n$, and writes the binary results 0 or 1 respectively in the binary variables y_i. Denote $y = (y_1, \dots, y_n)$.

The binary strings x and y are Alice's and Bob's raw keys. The rest stages of the protocol coincide with those of BB84, except that here we do not need sifting, since the bases of Alice and Bob are always consistent.

We analysed the intercept-resend attack on this protocol. The results of secret key fractions (in comparison with the BB84 and asymmetric BB84 protocol) are presented on Fig. 1.

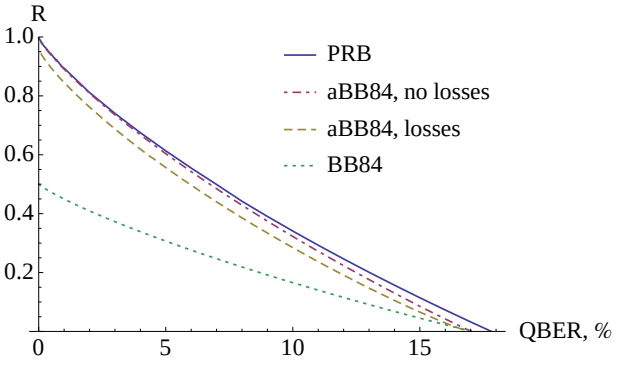

Figure 1: Secret key fractions of the presented pseudorandom bases (PRB) protocol of the presented pseudorandom bases (PRB) protocol, the BB84 protocol and the asymmetric BB84 (aBB84) protocol with and without losses in the quantum channel. It can be seen that the considered protocol gives better secret key rates than the BB84 protocol and approximately the same rates as the asymmetric BB84 protocol.

We can see that this protocol gives better secret key rates than the BB84 protocol and approximately the same rates as the asymmetric BB84 protocol. This part is based on the e-preprint arXiv: 1706.00611.

In the second part, we discuss quantum stream ciphers and, in particular, prove the impossibility of unconditionally secure quantum stream cipher.