Lightweight authentication for quantum key distribution

E.O. Kiktenko, A.O. Malyshev, M.A. Gavreev, A.A. Bozhedarov, N.O. Pozhar, M.N. Anufriev, and A.K. Fedorov

1Russian Quantum Center, Skolkovo, Moscow 143025, Russia
2Bauman Moscow State Technical University, Moscow 105005, Russia
3Steklov Mathematical Institute of Russian Academy of Sciences, Moscow 119991, Russia
4Moscow Institute of Physics and Technology, Dolgoprudny 141700, Russia
5Institute for Information Transmission Problems of the Russian Academy of Sciences, Moscow 127051, Russia
6Skolkovo Institute of Science and Technology, Moscow 121205, Russia

Quantum key distribution (QKD) enables unconditionally secure communication between distinct parties using a quantum channel and an authentic public channel. Reducing the portion of quantum-generated secret keys, that is consumed during the authentication procedure, is of significant importance for improving the performance of QKD systems. In the present work, we develop a lightweight authentication protocol for QKD based on a ‘ping-pong’ scheme of authenticity check for QKD. An important feature of this scheme is that the only one authentication tag is generated and transmitted during each of the QKD post-processing rounds. For the tag generation purpose, we design an unconditionally secure procedure based on the concept of key recycling. The procedure is based on the combination of almost universal\textsubscript{2} polynomial hashing, XOR universal\textsubscript{2} Toeplitz hashing, and one-time pad (OTP) encryption. We also demonstrate how to minimize both the length of the recycled key and the size of the authentication key, that is required for OTP encryption. Finally, we provide a security analysis of the full key growing process in the framework of universally composable security.

I. INTRODUCTION

QKD is a method for distributing provably secure cryptographic keys in insecure communications networks [1]. For this purpose, QKD systems encode information in quantum states of photons and transmit them through optical channels [2, 3]. The security of QKD is then based on laws of quantum physics rather than on computational complexity as is usually the case for public key distribution systems. This technology has attracted a significant amount of interest last decades, and industrial QKD systems are now available at retail [4–6].

Besides transmitting quantum states in the QKD technology, legitimate users also employ post-processing via an authentic public channel [2–5]. The post-processing procedure consists of key sifting, information reconciliation, privacy amplification, and other supplemental steps. The authentic classical channel, which prevents malicious modification of the transmitted classical data by an eavesdropper, is essential in order to prevent man-in-the-middle attacks. The problem of providing classical channel authenticity for QKD has been considered in various aspects in [7–14].

A conventional approach to the authentication problem in QKD systems is to use the Wegman-Carter scheme [15, 16]. This scheme provides unconditional security needed for QKD purposes but at the same time needs a pair of symmetric secret keys by itself. Moreover, like in the case of the unconditionally secure encryption with OTP [17], fresh secret keys are required for each use of the authenticated channel. From this perspective, the QKD workflow appears to be a key growing scheme, since the parties already need to have a short pair of pre-distributed keys before the launching the first QKD round. For authentication in the second and subsequent rounds, the parts of quantum-generated secret keys from the previous round could be used. In Ref. [18] it was shown that such an approach provides provable composable security of the whole key growing scheme. However, an increase in the performance of QKD systems faces a number of challenges, which include the reduction of authentication costs. An important task is then to find hash functions that allow minimizing the secret key consumption.

In our contribution, we present a practical authentication protocol specially designed for minimizing a secret its key consumption. This goal is achieved by (i) reducing the number of required generations of authentication tags down to one per QKD round and (ii) reducing the size of a secret key consumed by each tag generation. The reducing tag generations is achieved by modification of delayed authentication scheme considered in Ref. [7, 19, 20] into a ‘ping-pong’ scheme, in which the direction of the tag transmission alters each following QKD round. In order to minimize the secret key consumption on the tag generation, we consider a combining almost universal\textsubscript{2} polynomial hashing with XOR universal\textsubscript{2} Toeplitz hashing followed by the OTP encryption. This construction allows employing a key recycling approach [14, 15], where a permanent “recycling” key is used for polynomial and Toeplitz hashing, while only keys for the OTP encryption require an update. We pay special attention to the problem of optimizing a length of OTP keys since they are subtracted from previously quantum-generated keys and thus a need in their consumption reduces an effective QKD rate. Meanwhile, we show how to achieve a quite reasonable length of the used recycled key, which has to be picked up at once from the first QKD round.
We also study how the security of pre-shared authentication keys impacts further quantum-generated keys in the universally composable security framework. As a result, we obtain the authentication protocol with very low key consumption, that is why we refer to it as a lightweight authentication protocol.

Our work is organized as follows. In Sec. II, we describe the unconditionally secure message authentication code (MAC) scheme with the use of universal families. We also review some known approaches to improving their performance. In Sec. III, we present our lightweight authentication protocol and show how to derive its parameters with respect to the required security level. In Sec. IV we provide security analysis of the full key-growing QKD scheme. Finally, we summarize the main results and give an outlook in Sec. V.

II. UNCONDITIONALLY SECURE AUTHENTICATION

As soon as two legitimate parties, Alice and Bob, have an urge to communicate one each other, they almost inevitably face the problem of assuring that (i) received messages are indeed sent by a claimed sender; (ii) no adversary can forge messages such that fraud remains undetected. This problem is referred to as the authentication problem and it is known in cryptography for ages. For example, signatures and seals are ancient yet eligible solutions to authenticate handwritten documents and they provide enough intuition about what authentication is. Nevertheless, since the topic of this paper lies within the area of digital communication, further we restrict ourselves only to cryptographic authentication schemes [21].

From the very beginning, we put ourselves in the framework of unconditional (information-theoretic) security, where we do not rely on any assumption about the computational abilities of an adversary (Eve).

A. Message authentication codes with strongly universal hashing

A way for providing the authentication is to employ the MAC scheme. The main idea behind is as follows. Suppose that Alice and Bob have a common secret key $k$. Then they use the following procedure:

(i) If Alice wants to authenticate message $m$, she generates a MAC or, simply, a tag $t$, which is calculated based on $m$ and $k$, and then sends a pair $(m, t)$ to Bob.

(ii) Given a received pair $(m', t')$, which could different from $(m, t)$ because of an attack by Eve, Bob generates the corresponding tag $t_{\text{check}}$ from $m'$ and $k$, and checks whether the obtained $t_{\text{check}}$ is equal to $t'$.

(iii) If so, he supposes that the message was indeed sent by Alice and $m' = m$ (and $t' = t$).

The main point behind this protocol is that, having the intercepted a pair $(m, t)$ but not possessing $k$, for Eve it should be practically impossible to generate an alternative valid message-tag pair $(m', t')$ with $m \neq m'$ in order to cheat Bob. At the same time, it should be impossible for Eve to generate a pair $(m', t')$ for any $m'$ without any authenticated message previously transmitted by Alice.

The described protocol can be realized with the tag generation scheme on the basis of a strongly universal hash family. Strongly universal hash-functions have been proposed in Ref. [15] and then formally defined in Ref. [16]. A peculiar feature of this method is to employ a whole family of functions with desired properties, but no a single hash-function.

Let us consider a set of all possible messages $\mathcal{M}$, the set of all possible tags $\mathcal{T}$, and the set of keys $\mathcal{K}$. Each key $k \in \mathcal{K}$ defines a function from the family $h_k : \mathcal{M} \rightarrow \mathcal{T}$.

Definition 1 ($\varepsilon$-almost strong universal$_2$ family of functions). A family of functions

$$\mathcal{H} = \{ h_k : \mathcal{M} \rightarrow \mathcal{T} \}_{k \in \mathcal{K}}$$

is called $\varepsilon$-almost strongly universal$_2$ ($\varepsilon$-ASU$_2$) if for any distinct messages $m, m' \in \mathcal{M}$ and any tags $t, t' \in \mathcal{T}$ the following two conditions are satisfied:

$$\Pr_{k \leftarrow \mathcal{K}} [h_k(m) = t] = \frac{1}{|\mathcal{T}|}, \quad (2)$$

$$\Pr_{k \leftarrow \mathcal{K}} [h_k(m) = t, h_k(m') = t'] \leq \varepsilon \frac{1}{|\mathcal{T}|}. \quad (3)$$

If $\varepsilon = |\mathcal{T}|^{-1}$, then $\mathcal{H}$ is called strongly universal$_2$ (SU$_2$).

We note that that sometimes condition (2) is omitted. Here we use $k \leftarrow \mathcal{K}$ for defining a uniformly random generation of an element $k$ from the set $\mathcal{K}$.

In Ref. [15] it has been demonstrated that if one employs this family, unconditionally secure authentication can be achieved in the following way. Alice calculates a tag for message $m$ as $t := h_k(m)$, where $k$ is a secret key shared by Alice and Bob. Due to the fact that from the Eve’s perspective, the secret key is a uniform random variable, the probability of the impersonation attack, where Eve tries to generate a valid pair $(m', t')$ without any message sent by Alice, is limited by $|\mathcal{T}|^{-1}$ according to Eq. (2). It is directly follows from Eqs. (2) and (3) that for $m' \neq m$ the following expression holds:

$$\Pr_{k \leftarrow \mathcal{K}} [h_k(m') = t' | h_k(m) = t] \leq \varepsilon. \quad (4)$$

Therefore, the probability of the successful realization of the substitution attack, where Eve tries to modify a message (and probably a tag) originally sent by Alice, is limited by $\varepsilon$.

Thus, the use of an $\varepsilon$-ASU$_2$ family with small enough values of $\varepsilon$ allows legitimate parties Alice and Bob to achieve unconditionally secure authentication. The construction issues of appropriate $\varepsilon$-ASU$_2$ families was studied in various aspects [15, 22–26]. Each of these approach provides a family of hash-functions with its own trade-offs.
between sizes of $\mathcal{M}$ and $\mathcal{K}$, security parameter $\varepsilon$, and efficiency. Below we consider several common approaches for improving the performance of the authentication scheme based on $\varepsilon$-ASU$_2$ family.

### B. Key recycling

Here a point of concern is an amount of a key, which is distributed by legitimate parties and required for $\varepsilon$-ASU$_2$ hashing. It turns out that in order to achieve unconditional security, Alice and Bob have to use a distinct key $k$ for each message, i.e. the key consumption is significant. This shortcoming can be avoided. In Ref. [15] Wegman and Carter have proposed to choose a key $k$ once. For each generated tag, we use the XOR operation (bitwise modulo-2 addition) of the resulting tag with a new one-time pad (OTP) key $k_{\text{OTP}}$ of $\tau$ bits length, where $\tau$ is the tag size (here we assume that $T = \{0,1\}^\tau$) and $\tau \ll \log |\mathcal{K}|$. Consequently, in order to authenticate each new message legitimate parties recycle the key $k$ and review OTP keys $k_{\text{OTP}}$ only. Therefore, the key recycling procedure may decrease the demand in secret keys significantly.

Moreover, the key recycling scheme allows using a weaker class of function family, namely, $\varepsilon$-almost XOR universal$_2$.

**Definition 2** ($\varepsilon$-almost XOR universal$_2$ family of hash functions). A family of hash functions

$$\mathcal{H} = \{ h_k : \mathcal{M} \rightarrow \{0,1\}^\tau \}_{k \in \mathcal{K}}$$

is $\varepsilon$-almost XOR universal$_2$ ($\varepsilon$-AXU$_2$), if for all distinct $m, m' \in \mathcal{M}$, uniformly random chosen $k \in \mathcal{K}$ and any $c \in \{0,1\}^\tau$,

$$\Pr_{k \in \mathcal{K}} [h_k(m) \oplus h_k(m') = c] \leq \varepsilon,$$

where $\oplus$ stands for XOR. If $\varepsilon = |T|^{-1}$, then the family of such hash functions is called XOR universal$_2$ (XU$_2$).

Consider the following authentication scheme based on an $\varepsilon$-AXU$_2$ family

$$\mathcal{H} = \{ h_k : \mathcal{M} \rightarrow \{0,1\}^\tau \}_{k \in \mathcal{K}}$$

with the use of the key recycling scheme. One can see that this is equivalent to the construction of a new family of the following form:

$$\mathcal{H}^{\text{ext}} = \{ h_{k,k_{\text{OTP}}} : \mathcal{M} \rightarrow \{0,1\}^\tau \}_{k \in \mathcal{K}, k_{\text{OTP}} \in \{0,1\}^\tau}$$

with

$$h_{k,k_{\text{OTP}}}(m) := h_k(m) \oplus k_{\text{OTP}}.$$

One can see that the resulting family $\mathcal{H}^{\text{ext}}$ is $\varepsilon$-ASU$_2$.

Therefore, the authentication scheme based on the use of an $\varepsilon$-AXU$_2$ family with the key recycling scheme is equivalent to the authentication scheme on the basis of an $\varepsilon$-ASU$_2$ family. Thus, it allows one to achieve both unconditionally secure authentication and reducing the size of the secret key consumed by each tag generation.

It is important to note that in this scheme the use of the key recycling scheme leads to a disclosure of the used hash function after a number of rounds (the number of rounds can be made arbitrarily large). A security proof of the considered scheme in a composable security framework is provided in Ref. [14]. It is shown that schemes with authenticating $n$ messages based on the key recycling scheme and $\varepsilon$-AXU$_2$ family is $n\varepsilon$-secure. Following the notion of the composable security framework [14], it means that an abstract distinguisher with unlimited computational resources, has an advantage at most $n\varepsilon$ in distinguishing the considered scheme from an ideal authentication system. Such a notion of security is particularly important for security analysis of the QKD post-processing procedure.

### C. Combining families of hash functions

A useful technique for construction of ASU$_2$ hash family with desired properties is based on the composition of a given family ASU$_2$ and a family with weaker requirements, such as a universal$_2$ family [27].

**Definition 3** ($\varepsilon$-almost universal$_2$ family of functions). A family of functions

$$\mathcal{H} = \{ h_k : \mathcal{M} \rightarrow T \}_{k \in \mathcal{K}}$$

is called $\varepsilon$-almost universal$_2$ ($\varepsilon$-AU$_2$) if for any distinct messages $m, m' \in \mathcal{M}$ and uniformly chosen $k \in \mathcal{K}$,

$$\Pr_{k \in \mathcal{K}} [h_k(m) = h_k(m')] \leq \varepsilon$$

If $\varepsilon = |T|^{-1}$, then $\mathcal{H}$ is called universal$_2$ (U$_2$).

A composition of $\varepsilon_1$-AU$_2$ and $\varepsilon_2$-ASU$_2$ families results in $(\varepsilon_1 + \varepsilon_2)$-ASU$_2$ family [16]. Therefore, having two hash families, one has a useful tool for obtaining a resulting $\varepsilon$-ASU$_2$ family with desired properties.

The similar strategy is applicable in principle for constructing an $\varepsilon$-AXU$_2$ family using the composition $\varepsilon$-AU$_2$ and $\varepsilon$-AXU$_2$ families.

**Theorem 1.** Let

$$\mathcal{H}^{(1)} = \{ h_k^{(1)} : \mathcal{M} \rightarrow T_1 \}_{k \in \mathcal{K}_1}$$

be an $\varepsilon_1$-AU$_2$ family, and

$$\mathcal{H}^{(2)} = \{ h_k^{(2)} : T_1 \rightarrow \{0,1\}^\tau \}_{k \in \mathcal{K}_2}$$

be an $\varepsilon_2$-AXU$_2$ family. Then the following family

$$\mathcal{H} = \{ h_{k_1,k_2} = h_k^{(2)} \circ h_k^{(1)} : \mathcal{M} \rightarrow \{0,1\}^\tau \}_{k_1 \in \mathcal{K}_1, k_2 \in \mathcal{K}_2}$$

is an $(\varepsilon_1 + \varepsilon_2)$-AXU$_2$ family. Here $\circ$ holds for the function composition.
Abort QKD process

See Appendix A 1 for the proof. This theorem allows us to obtain an \((\varepsilon_1 + \varepsilon_2)\)-ASU_2 hash family as the composition of \(\varepsilon_1\)-AU_2 family with \(\varepsilon_2\)-AXU_2 family and the OTP encryption. We note that this construction allows obtaining a possibility of recycling a key required in \(\varepsilon_1\)-U_2 family with \(\varepsilon_2\)-AXU_2 family.

III. LIGHTWEIGHT AUTHENTICATION PROTOCOL FOR QKD

In this section, we consider an authentication task in the framework of QKD and introduce our lightweight authentication protocol. The protocol is based on the key recycling scheme and the universal families combination approach (described above) as well as on a concept of the ping-pong delayed authentication.

A. Ping-pong delayed authentication

The workflow of a QKD device can be split into two main stages. The first “quantum” stage is related to the preparing, transmitting and measuring quantum signals (usually, attenuated laser pulses) through an untrusted quantum channel (optical fiber or free space). As the result of the first stage, the parties, Alice and Bob, obtain a set of records regarding preparing and measuring events. These records are usually referred to raw keys. The second stage is the classical post-processing procedure. It is aimed on the extraction (or distillation) of secret keys from raw keys, or coming to a conclusion that such an extraction is impossible due to eavesdropper’s (Eve’s) activities. The criteria here is that the quantum bit error rate (QBER) value exceeds a certain critical threshold [2]. In the latter case, the parties just abort the QKD session.

The post-processing stage typically consists of sifting, parameter estimation, information reconciliation, and privacy amplification procedures [28]. During these procedures, the parties communicate with each other via a classical channel, which has to be authenticated in an unconditionally secure way. It means that any tampering with classical communication by Eve in the classical channel should result in aborting QKD protocol in the same way how it is aborted in the case of its interception in the quantum channel. If the QBER value is below the threshold and no tampering in classical communication is detected, Alice and Bob obtain a pair of provably secret keys. Then the post-processing procedure can be repeated again with a new pair of raw keys. We refer to such sequences of stage repeating as rounds.

The main point of concern is that the unconditionally secure authentication scheme itself requires a symmetric key. That is why the process of QKD can be considered as a key-growing scheme (or secret-growing scheme): Alice and Bob have to share some pre-distributed keys in order to provide authentication in the first QKD round and they use a portion of quantum-generated keys for authentication in the following rounds. That is why reducing the portion of quantum-generated secret keys, that is consumed during the authentication procedure, is of significant importance for improving the performance of QKD systems.

It turns out that it is reasonable to check the authenticity of the classical channel in the very end of the post-processing round, rather than to add an authentication tag to each classical message separately. The idea is that if Eve has tampered with a classical channel, then the legitimate parties are able to detect this event by authenticating all the traffic in the classical channel once. If the authentication check procedure fails, then secret keys will not be generated. Meanwhile, in the favourable case, if Eve does not interfere with classical communication, the authentication key consumption will be small.

However, an important point is that, actually, there are two classical channels which authenticity should be checked: from Alice to Bob and from Bob to Alice. In order to obtain secure symmetric keys, each of the legitimate parties has to be sure that all the sent and received messages are not modified by Eve.

To address this issue we propose the following solution for authenticity check, that it is applied through pairs of rounds rather than a single round. The scheme of the designed solution is presented in Fig. 1. During each

![FIG. 1. Scheme of the QKD post-processing procedure with delayed authentication. Each of legitimate parties generate and check authentication tags once for the whole traffic during the post-processing procedure.](image-url)
of the post-processing rounds the unconditionally secure authentication tag is transmitted from one party to another in a ‘ping-pong’ manner: If in \( N \)th round a tag is sent from Alice to Bob, then in \((N+1)\)th round a tag is sent back from Bob to Alice, and so on. The tag is computed after a final privacy amplification step and is based on a string composed of all the income and outcome classical messages transmitted within the current post-processing round. After receiving the tag, the legitimate party checks its validity by computing a verification tag from his (her) own versions of sent and received classical messages. If the check passed, then the party becomes sure that (up to the fixed error probability) the classical communication was not modified by Eve, so the obtained keys are provably secure. Otherwise, the party terminates the whole QKD process, tries to reach the partner on another channel, and compare the state of secret pools.

In the case of a successful check, the party adds the obtained key into the pool of secret keys. Moreover, the party also adds to the distilled key from the previous round (if the number of the current round is greater than one). This is because at the current round the party who receives the tag becomes sure that in the previous round, where he (she) was a tag sender, the authentication check was also passed. Otherwise, the protocol should be already terminated on the other side and no valid tag should come at the current round.

Thus, the key consumption is defined by computing a single tag of all the classical messages used in the round. The exception is a final round where additional authentication check was also passed. Otherwise, the protocol is sent back from Bob to Alice, and so on. The tag is computed after a final privacy amplification step and is sent from Alice to Bob, then in \((N+1)\)th round an authentic acknowledgment message from a party who performed a check is required. The generation of this message could be considered as a fictitious post-processing round where no secret key is produced.

### B. Instantiation of universal families

Here we consider the problem of minimizing a key consumption for computing a tag with a given error probability threshold. For this purpose, we employ the key recycling approach based on using AXU2 family with the OTP encryption (see Subsection II B). We construct an AXU2 family by combining AU2 and XU2 families.

Let us consider a value \( \varepsilon_{\text{auth}} \) that describe an error probability during the authentication procedure. On the one hand, with a given fixed \( \varepsilon_{\text{auth}} \), it is preferable to have a length of the final tag to be as small as possible to minimize the OTP key consumption. This is due to the fact that the length of the tag is equal to the one of the OTP key during each round of the authentication procedure. Therefore, it is optimal to use an XU2 family with the minimal possible hash tag length \( \tau = \lceil \log_2 \varepsilon_{\text{auth}} \rceil \) (hereinafter \( \lceil \cdot \rceil \) and \( \lfloor \cdot \rfloor \) stand for the standard ceil and floor rounding operations).

On the other hand, it is also important to minimize the length of the recycled key as well. It immediately follows from the Stinson bound [29], that a key size defining an element from XU2 family \(|{\mathcal{K}}\)| is at least as large as a length of input message \(|{\mathcal{M}}|\), which is quite expensive for use in the QKD. In order to decrease the length of the required key, we employ a preceding XU2 hashing with using a function from AU2 family for decreasing the length of the input string. Such a pre-compression comes together with an additional collision probability \( \varepsilon_{1} \) corresponding to the employed \( \varepsilon_{1}\text{-AU2} \) family. However, we can choose \( \varepsilon_{1} \ll 2^{-\tau} \) such that the total error probability \( \varepsilon_{1}+2^{-\tau} \) becomes almost equal to the optimal value \( 2^{-\tau} \).

In the result, we obtain a scheme which has an optimal OTP key consumption with a recycled key length of the order of \( \log \log |{\mathcal{M}}| \). Let us then consider choice of the particular \( \varepsilon\text{-AU2} \) and XU2 families.

For an ASU2 family we choose a modification of the well-known method of polynomial hashing [30]. This approach works as follows. It starts from calculating a polynomial over a finite field with coefficients given by the input and the calculation point given by the random key. Consider a prime number \( p \) given in the following form:

\[
p = 2^w + \delta_w,
\]

where integers \( w \) and \( \delta_w \) be a smallest as possible (see Appendix B for a list of such primes). Let \( \mu \) be an upper bound on a length of an authenticated message, that is a maximal total length of all the messages used in a single round of the QKD post-processing. Consider a family

\[
{\mathcal{H}}^{(1)} = \{h^{(1)}_k : \{0,1\}^\mu \rightarrow \{0,1\}^{w+1}\}_{k \in \{0,1\}^w}
\]

with

\[
h^{(1)}_k(m) = \text{str}\left(\sum_{i=1}^l \text{int}(m_i)\text{int}(k)^{i-1} \pmod p\right).
\]

Here \( \{0,1\}^{\mu} \) stands for a set of all bit-strings of length less or equal to \( \mu \), \( l := \lceil \mu + 1/w \rceil \), \( \text{str} \) and \( \text{int} \) are standard functions providing a transition between bit-string and integer number representation, and \( \{m_i\}_{i=1}^l \) are \( w\)-bit chunks of \( m \), obtained first by concatenation of \( m \) with 1 followed by a block of zeros in order to achieve an extended string of length \( lw \) and then by splitting the resulting string into \( l \) of \( w\)-bit pieces.

A precise statement is then as follows:

**Theorem 2.** \( {\mathcal{H}}^{(1)} \) is an \( \varepsilon_{1}\text{-AU2} \) family with

\[
\varepsilon_{1} = \lceil \mu/w \rceil 2^{-w}.
\]

See Appendix A 2 for the proof.

From the practical point of view, it is also useful to consider a generalized family of the following form:

\[
{\mathcal{H}}^{(1)}_\lambda = \{h_{k_1,\ldots,k_\lambda}^{(1)}() = h^{(1)}_{k_1}()\cdots h^{(1)}_{k_{\lambda}}():
\{0,1\}^{\mu} \rightarrow \{0,1\}^{\lambda(w+1)}\}_{(k_1,\ldots,k_{\lambda}) \in \{0,1\}^\lambda}
\]

This family is obtained by concatenation of \( \lambda \geq 1 \) independent instances of functions from family \( {\mathcal{H}} \). The following statement holds true:
Theorem 3. $\mathcal{H}^{(1)}_\lambda$ is an $\varepsilon_1^\lambda$-AU$_2$ family with $\varepsilon_1$ of the following form:

$$\varepsilon_1 = \lfloor \mu/w \rfloor 2^{-\alpha}.$$  \hfill (20)

See Appendix A 3 for the proof.

We remind here that the idea behind this generalization is that one can decrease the collision probability down to the desired level without an increase in the employed ring modulus defined by the value of $w$. Thus, it is possible to set $w = 31$ or $w = 63$ in order to perform all the calculation with 32 and 64-bit integers, correspondingly.

As the AU$_2$ family for our protocol, we use Toeplitz hashing [24] given by multiplying a message with the Toeplitz matrix in the form:

$$T_k := \begin{bmatrix} k_\beta & k_{\beta+1} & k_{\beta+2} & \ldots & k_{\beta+\alpha-1} \\ k_{\beta-1} & k_\beta & k_{\beta+1} & \ldots & k_{\beta+\alpha-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_1 & k_2 & k_3 & \ldots & k_\alpha \end{bmatrix},$$  \hfill (21)

which is defined by a binary string of the following form:

$$k = (k_1, \ldots, k_\alpha, k_{\alpha+\beta-1}).$$  \hfill (22)

Consider a family

$$\mathcal{H}^{(2)} = \{h_k^{(2)} : \{0,1\}^\alpha \rightarrow \{0,1\}^\beta\}_{k \in \{0,1\}^{\alpha+\beta-1}},$$  \hfill (23)

with

$$h_k^{(2)}(x) = T_k \cdot x \pmod{2},$$  \hfill (24)

where $x$ is treated as the column vector, and $\cdot$ stands for the dot-product. A precise statement is then as follows:

Theorem 4. $\mathcal{H}^{(2)}$ is an AU$_2$ family.

See Appendix A 4 for the proof.

In order to combine Toeplitz hashing with polynomial hashing, we have to set $\alpha = \lambda (w + 1)$, and $\beta = \tau$, where $\tau$ is the length of the final authentication tag. Consequently, we obtain a scheme illustrated in Fig. 2.

FIG. 2. Scheme for the the authentication tag calculation procedure in the proposed lightweight authentication protocol for QKD.

| $w$ | $\mu$, Mbits | $L_{\text{rec}}$, bits | $L_{\text{OTP}}$, bits |
|-----|--------------|------------------------|------------------------|
| 31  | 135          | 40                     | 167                    |
| 63  | 199          | 40                     | 167                    |
| 128 | 231          | 40                     | 231                    |
| 256 | 263          | 40                     | 231                    |

TABLE I. Quantum key consumption on recycled key $L_{\text{rec}}$ and OTP key $L_{\text{OTP}}$ for different bound on total length of classical messages transmitted during a single QKD round $\mu$. The security parameter is fixed at the level $\varepsilon_{\text{auth}} = 10^{-12}$.

The length of the recycled key combined from keys for defining $\mathcal{H}^{(1)}_\lambda$ and $\mathcal{H}^{(2)}$ elements is given by the following expression:

$$L_{\text{rec}} = 2\lambda w + \lambda + \tau - 1,$$  \hfill (25)

while the length of OTP key is given by $L_{\text{OTP}} = \tau$. The final security parameter is as follows:

$$\varepsilon = 2^{-\tau} + \left\lfloor \mu/w \right\rfloor \lambda 2^{-\lambda w}.$$  \hfill (26)

It grows polynomially with the total size of messages in the classical channel during the post-processing round $\mu$.

Finally, we consider an optimal way of choosing parameters for the proposed authentication scheme. We start with a given upper bound on tolerable authentication error probability $\varepsilon_{\text{auth}}$ and the maximal total length of messages $\mu$. We also restrict ourselves to the consideration only two cases of $w = 31$ and $w = 63$.

In order to have an smallest possible OTP key consumption, it is practical to set the following value:

$$\tau := \lfloor \varepsilon_{\text{auth}} \rfloor + 1.$$  \hfill (27)

The remaining part of $\varepsilon_{\text{auth}}$ could be used for AU$_2$ family, so we have to find a minimal possible integer $\lambda$ so that the following condition is fulfilled:

$$\left\lfloor \mu/w \right\rfloor \lambda 2^{-\lambda w} \leq \varepsilon_{\text{auth}} - 2^{-\tau}.$$  \hfill (28)

The performance of the suggested scheme for different values of $\mu$ and $w \in \{31, 63\}$ and the fixed value of the security parameter $\varepsilon_{\text{auth}} = 10^{-12}$ is presented in Table I. For optimal size of the OTP key, the length of the recycled key is low, and it seems to be easy to accumulate the required size during the very first QKD round.

We also provide a comparison of our approach performance with experimental results on the realization of a fast and versatile QKD system reported in Ref. [31]. In the considered QKD setup, each round of the post-processing procedure was executed for sifted key blocks of length $L_{\text{sift}} = 995,328$ bits. For the authentication purposes, the key recycling technique with ASU$_2$ family from Ref. [26] was used. In the realized authentication scheme a 127-bit tag was generated for every 2$^{20}$ bits of
classical communication providing the authentication security parameter \( \varepsilon_{\text{auth}} = 10^{-33} \). As the main figure of merit for the authentication efficiency, we consider a relative authentication cost \( c \) defined as a fraction of secret key consumed for authentication in the following rounds. It has the following form:

\[
c = \frac{L_{\text{OTP}}}{L_{\text{sec}}} = \frac{\lceil \varepsilon_{\text{auth}} \rceil + 1}{L_{\text{sift}} \cdot \eta_{\text{pa}}}. \tag{29}
\]

Here \( L_{\text{sec}} \) is a length of a secret key produced after privacy amplification procedure, and \( \eta_{\text{pa}} \) is the privacy amplification compression coefficient \( (L_{\text{sec}} = L_{\text{sift}} \cdot \eta_{\text{pa}}) \).

We present a comparison of the relative authentication costs calculated from the experimental data presented in Table 1 of Ref. [31] with corresponding values for proposed lightweight authentication protocol given by Eq. (29). One can see that the lightweight authentication protocol reduces the relative authentication costs by \( \approx 28 \) times (see Fig. 3).

## IV. SECURITY ANALYSIS OF KEY GROWING

Here we consider the security of the proposed authentication protocol in the framework of the full key-growing workflow. The scheme is presented in Fig. 4. Let us recall that before the first round of the QKD process Alice and Bob have to pre-distribute a pair of secret keys in order to provide the required classical authenticated channel. However, the ping-pong principle of the employed delayed authentication scheme requires the pre-distributed keys in the second round as well, since the parties come to the secret key pool only in the second round. We work here in the framework of the universally composable security [18].

In order to provide authentication during first and second rounds we use the considered AU \( U_2 + Xu_2 + \text{OTP} \) scheme (see Sec. III). The length of the pre-distributed keys is as follows:

\[
L_{\text{pred}} = L_{\text{rec}} + 2L_{\text{OTP}}. \tag{30}
\]

The crucial point is that the pre-distributed keys have to be secure only up to the moment of the second authentication check: In the case of successful completion of the first two QKD rounds the parties can discard the pre-distributed keys (or even announce them publicly), and switch to the use of quantum-generated keys.

We use \( \varepsilon_{\text{pred}}(t_1) \) as a security parameter for the pre-distributed keys at the moment of the authentication check of the first QKD round happening at time \( t_1 \). However, a quite nontrivial question is quantifying \( \varepsilon_{\text{pred}}(t_1) \), since in most practical scenarios the pre-distribution is based on some computationally secure algorithms (currently used public-key algorithms or post-quantum algorithms), or employs low-entropy “passwords”, known by Alice and Bob. Here for our purposes, we define \( \varepsilon_{\text{pred}}(t_1) \) as an advantage which can have an eavesdropper in distinguishing pre-distributed keys and ideal keys in an assumption that she has access to technological capabilities accumulated by mankind by the time \( t_1 \).

The resulting security parameter of keys generated in the first QKD round is given by the following expression:

\[
\varepsilon_1 = \varepsilon_{\text{pred}}(t_1) + \varepsilon_{\text{auth}} + \varepsilon_{\text{QKD}}, \tag{31}
\]

where \( \varepsilon_{\text{QKD}} \) is the security parameter of the used classical post-processing procedure including parameter estimation, information reconciliation, and privacy amplification stages. We note that in contrast to the security of the pre-distributed keys, the value of \( \varepsilon_{\text{QKD}} \) is calculated exactly based on the parameters of the employed post-processing algorithms. For modern QKD setups its typical range is from \( 10^{-3} \) up to \( 10^{-12} \) [28]. The key, that is obtained from the first round, has to be split into three parts: (i) the recycled authentication keys of length \( L_{\text{rec}} \), (ii) OTP authentication keys of length \( L_{\text{OTP}} \), and (iii) the rest part can be used for any external applications.

For authentication purposes during the second QKD round at the moment \( t_2 \), we use the recycled key with the second OTP block from the distribution (see Fig. 4). The security of the pre-shared key is \( \varepsilon_{\text{store}}(t_1, t_2) = \varepsilon_{\text{store}}(t_1) + \varepsilon_{\text{store}}(t_2) \), where \( \varepsilon_{\text{store}}(t_1, t_2) \) is an additional advantage that the adversary obtain in distinguishing pre-distributed keys from ideal keys using all achievable technological capabilities during storage time from \( t_1 \) up to \( t_2 \). According to the paradigm of composable security, the resulting security parameter of the generated keys is given by the following expression:

\[
\varepsilon_2 = \varepsilon_{\text{pred}}(t_1) + \varepsilon_{\text{store}}(t_1, t_2) + 2(\varepsilon_{\text{auth}} + \varepsilon_{\text{QKD}}). \tag{32}
\]

The resulting key have to be split into two pieces: the first OTP part of length \( L_{\text{OTP}} \) and the rest part, which could be used in external applications.

For the authentication in the \( N^{\text{th}} \) round \( (N \geq 3) \) the parties employ a recycled key from the first round and
v. Conclusion and outlook

In this work, we have developed the novel lightweight authentication protocol for QKD systems. We have described the practical unconditionally secure authentication scheme for QKD systems, which is based on combining two basic ideas: (i) the proposed ping-pong scheme, which allows generating only a single authentication tag during a QKD post-processing round, and (ii) unconditionally secure tag generation based AXU family with OTP encryption, which allows further reducing key consumption by using key recycling. We also have demonstrated how to construct suitable AXU families using the combination of polynomial AU and Toeplitz XU families. Finally, we have obtained the scheme which minimizes the OTP key consumption for given authentication error bound $\varepsilon_{\text{auth}}$.

The proposed scheme is promising for industrial QKD setups due to the possible enhancement of effective secret key generation rate. Another interesting area, where the considered approach could be implemented, is QKD networks. The idea is that in order to generate unconditionally secure secret keys between two parties in a QKD network using a trusted node scheme, all classical communication should be authenticated in an unconditionally secure manner. For that purposes, a considered implementation of the key recycling approach could be employed.

ACKNOWLEDGEMENTS

The work was supported by the Russian Foundation for Basic Research (18-37-20033).

Appendix A: Proofs of the theorems

1. Proof of the Theorem 1

Proof. Consider two distinct element $m$ and $m'$ from $\mathcal{M}$. We need to prove that for any $c \in \{0,1\}^r$, the following relation holds:

$$\Pr_{k_1 \overset{\$}{\leftarrow} \mathcal{K}_1, k_2 \overset{\$}{\leftarrow} \mathcal{K}_2} [h_{k_1,k_2}(m) \oplus h_{k_1,k_2}(m') = c] \leq \varepsilon_1 + \varepsilon_2.$$  \hspace{1cm} (A1)

First, we set $c = (0, \ldots, 0)$, then the condition

$$h_{k_1,k_2}(m) \oplus h_{k_1,k_2}(m') = c$$  \hspace{1cm} (A2)

reduces to

$$h_{k_1,k_2}(m) = h_{k_1,k_2}(m').$$  \hspace{1cm} (A3)

The LHS of Eq. (A1) can be written as follows:

$$\Pr_{k_1 \overset{\$}{\leftarrow} \mathcal{K}_1} [h_{k_1}^{(1)}(m) = h_{k_1}^{(1)}(m')] + \Pr_{k_2 \overset{\$}{\leftarrow} \mathcal{K}_2} [h_{k_2}^{(2)}(h_{k_1}^{(1)}(m)) = h_{k_2}^{(2)}(h_{k_1}^{(1)}(m'))] \Pr_{k_1 \overset{\$}{\leftarrow} \mathcal{K}_1, k_2 \overset{\$}{\leftarrow} \mathcal{K}_2} [h_{k_1}^{(1)}(m) \neq h_{k_1}^{(1)}(m')] \times \Pr_{k_1 \overset{\$}{\leftarrow} \mathcal{K}_1} [h_{k_1}^{(1)}(m) \neq h_{k_1}^{(1)}(m')].$$  \hspace{1cm} (A4)

The first term in Eq. (A1) is bounded by $\varepsilon_1$ according to the definition of an $\varepsilon_1$-AU family, while the second term can be bounded by $\varepsilon_2$ according to the definition of an $\varepsilon_2$-XU family. Thus, the final upper bound is given by $\varepsilon_1 + \varepsilon_2$.

Then we set $c \neq (0, \ldots, 0)$. One can see that in order to have

$$h_{k_1,k_2}(m) \oplus h_{k_1,k_2}(m') = c$$  \hspace{1cm} (A5)
it is necessary to have
\[ h_{k_1}^{(1)}(m) \neq h_{k_1}^{(1)}(m'). \] (A6)
Then we can rewrite LHS of Eq. (A1) in the following form:
\[
\Pr_{k_2 \in \mathcal{K}_2} \left[ h_{k_2}^{(2)}(h_{k_1}^{(1)}(m)) \oplus h_{k_2}^{(2)}(h_{k_1}^{(1)}(m')) = c \right] \\
h_{k_1}^{(1)}(m) \neq h_{k_1}^{(1)}(m') \Pr_{k_1 \in \mathcal{K}_1} \left[ h_{k_1}^{(1)}(m) \neq h_{k_1}^{(1)}(m') \right].
\] (A7)
One can see that the value of Eq. (A7) is bounded by the value of \( \varepsilon_2 \) according to the definition of a \( \varepsilon_2 \)-XU\(_2\) family, that ends the proof.

\[ \square \]

2. Proof of the Theorem 2

**Proof.** Consider two distinct input bit strings \( m \) and \( m' \) of length not larger than \( \mu \). The condition on the collision event is as follows:
\[
\sum_{i=1}^{l} \left( \text{int}(m_i) - \text{int}(m'_i) \right) \text{int}(k)^{i-1} = 0 \pmod{p}, \] (A8)
where \( \{m_i\}_{i=1}^{l} \) and \( \{m'_i\}_{i=1}^{l} \) are \( w \)-bit chunks of \( m \) and \( m' \) correspondingly and \( l := \lceil (\mu + 1)/w \rceil \). Since \( m \neq m' \), there is at least one \( m_i \neq m'_i \), so the LHS of Eq. (A8) can be considered as a non-zero polynomial of the degree that is not larger than \( l - 1 \). The collision corresponds to the case, where the value random \( k \) taken from \( \{0,1\}^w \) turns out to be a root of this polynomial. According to the fundamental theorem of algebra for finite fields, there are no more than \( l - 1 \) roots, and the collision probability is upper bounded by the following value:
\[
\frac{l - 1}{|\mathcal{K}|} \leq \frac{\mu}{w} 2^{-w}, \] (A9)
since \( \lfloor (\mu + 1)/w \rfloor - 1 \leq \lfloor \mu/w \rfloor \). \[ \square \]

3. Proof of the Theorem 3

**Proof.** Consider two distinct input strings \( m \) and \( m' \) of length not larger than \( \mu \). The condition of the identity of
\[
h_{k_1, \ldots, k_\lambda}(m) = h_{k_1, \ldots, k_\lambda}(m')
\] (A10)
is equivalent to a set of identities \( h_{k_i}^{(1)}(m) = h_{k_i}^{(1)}(m') \) for all \( i = 1, \ldots, \lambda \). Since all \( k_i \) are independent, we obtain
\[
\Pr_{(k_1, \ldots, k_\lambda) \in \{0,1\}^w^\lambda} \left[ h_{k_1, \ldots, k_\lambda}(m) = h_{k_1, \ldots, k_\lambda}(m') \right] = \\
\prod_{i=1}^{\lambda} \Pr_{k_i \in \{0,1\}^w} \left[ h_{k_i}(m) = h_{k_i}(m') \right] \leq \varepsilon_2^{\lambda} \] (A11)
due to results of the Theorem 2. \[ \square \]

4. Proof of the Theorem 4

**Proof.** Consider two distinct binary vectors \( x \) and \( x' \) from \( \{0,1\}^\alpha \) and some binary vector \( c \) from \( \{0,1\}^\beta \). Let \( \Delta x = x \oplus x' \). Denote by \( j \) the index of the first nonzero element of \( \Delta x \) (we note that \( j \) always exists since \( x \neq x' \)). The condition
\[
h_k(x) \oplus h_k(x') = c
\] (A12)
can be rewritten in the following form:
\[
T_k \cdot \Delta x = c.
\] (A13)
We can rewrite Eq. (A13) as follows:
\[
k_{\beta+1-i} + \sum_{\gamma=j+1}^{\alpha} k_{\beta+j-i+\gamma} \Delta x_{\gamma} = c_i \text{ for } i = 1, \ldots, \beta
\] (A14)
Since all the elements of \( k \) are independent random bits, it easy to see that for each \( i = 1, \ldots, \beta \)
\[
\Pr_{k_{\beta+i} \in \mathcal{K}} \left[ k_{\beta+j-i+\gamma} \Delta x_{\gamma} = c_i \right] = \frac{1}{2}
\] (A15)
independently from each other. Thus, the total probability \( \Pr_{k_{\beta+i} \in \mathcal{K}} [T_k \cdot \Delta x = c] \) is given by \( 2^{-\beta} \). \[ \square \]

\[
\begin{array}{cccccc}
2^{15} + 3 & 2^{25} + 35 & 2^{35} + 53 & 2^{45} + 59 & 2^{55} + 3 \\
2^{16} + 1 & 2^{26} + 15 & 2^{36} + 31 & 2^{46} + 15 & 2^{56} + 81 \\
2^{17} + 9 & 2^{27} + 29 & 2^{37} + 9 & 2^{47} + 5 & 2^{57} + 9 \\
2^{18} + 23 & 2^{28} + 17 & 2^{38} + 23 & 2^{48} + 21 & 2^{58} + 69 \\
2^{19} + 21 & 2^{29} + 11 & 2^{39} + 15 & 2^{49} + 69 & 2^{59} + 131 \\
2^{20} + 7 & 2^{30} + 3 & 2^{40} + 15 & 2^{50} + 55 & 2^{60} + 33 \\
2^{21} + 15 & 2^{31} + 11 & 2^{41} + 15 & 2^{51} + 21 & 2^{61} + 15 \\
2^{22} + 15 & 2^{32} + 15 & 2^{42} + 15 & 2^{52} + 21 & 2^{62} + 135 \\
2^{23} + 9 & 2^{33} + 17 & 2^{43} + 23 & 2^{53} + 5 & 2^{63} + 29 \\
2^{24} + 43 & 2^{34} + 25 & 2^{44} + 7 & 2^{54} + 159 & 2^{64} + 13
\end{array}
\]

**TABLE II.** Prime numbers in a form \( 2^w + \delta_w \) for \( w \) from 15 up to 64.

**Appendix B: Prime numbers for polynomial hashing**

In Table A4, we provide a list of prime numbers which can be used in the considered scheme of polynomial hashing.
[1] C.H. Bennet and G. Brassard, in Proceedings of IEEE International Conference on Computers, Systems and Signal Processing Bangalore, India (IEEE, New York, 1984), p. 175.
[2] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, Rev. Mod. Phys. 74, 145 (2002).
[3] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dusek M, N. Lütkenhaus, and M. Peev, Rev. Mod. Phys. 81, 1301 (2009).
[4] H.-K. Lo, M. Curty, and K. Tamaki, Nat. Photonics 8, 595 (2014).
[5] E. Diamanti, H.-K. Lo, and Z. Yuan, npj Quant. Inf. 2, 16025 (2016).
[6] Characteristics of commercial devices from ID Quantique (Switzerland), SeQureNet (France), and the Austrian Institute of Technology (Austria) are available.
[7] C. Pacher, A. Abidin, T. Lorünser, M. Peev, R. Ursin, A. Zeilinger, and J.-Å. Larsson, Quant. Inf. Proc. 15, 327 (2016).
[8] M. Peev, M. Nölle, O. Maurhardt, T. Lorünser, M. Suda, A. Poppe, R. Ursin, A. Fedrizzi, and A. Zeilinger, Int. J. Quantum. Inf. 3, 225 (2004).
[9] J. Cederlöf and J.-Å. Larsson, IEEE Trans. Inf. Theory 54, 1735 (2008).
[10] A. Abidin, C. Pacher, T. Lorünser, J.-Å. Larsson, and M. Peev, Proc. SPIE 8189, 8189 (2011).
[11] S. Kunz-Jacques and P. Jouguet, arXiv:1109.2844.
[12] M. Mosca, D. Stefila, and B. Ustaoglu, Lect. Notes Comp. Sci. 7932, 136 (2013).
[13] A. Abidin and J.-Å. Larsson, Quant. Inf. Proc. 13, 2155 (2014).
[14] C. Portmann, IEEE Trans. Inf. Theory 60, 4383 (2014).
[15] M.N. Wegman and J.L. Carter, J. Comp. Syst. Sci. 22, 265 (1981).
[16] D.R. Stinson, Des., Codes Cryptography 4, 369 (1994).
[17] G.S. Vernam, J. Amer. Inst. Electr. Engineers 45, 109 (1926).
[18] J. Müller-Quade and R. Renner, New J. Phys. 11, 085006 (2009).
[19] O. Maurhart, Lect. Notes Phys. 797, 151 (2010).
[20] M. Peev et al., New J. Phys. 11, 075001 (2009).
[21] B. Schneier, Applied cryptography (John Wiley & Sons, Inc., New York, 1996).
[22] D.R. Stinson, J. Combin. Math. Combin. Comput. 42, 3 (2002).
[23] B.D. Boer, J. Comp. Sec. 2, 65 (1993).
[24] H. Krawczyk, Lect. Notes Comput. Sci. 839, 129 (1994).
[25] D. Lemire and O. Kaser, Comput. J. 57, 1624 (2014).
[26] J. Bierbrauer, T. Johansson, G. Kabatianskii, and B. Smeets, Lect. Notes Comput. Sci. 773, 331 (1994).
[27] M.N. Wegman and J.L. Carter, J. Comp. Syst. Sci. 18, 143 (1979).
[28] E. O. Kiktenko, A. S. Trushechkin, Y. V. Kurochkin, A. K. Fedorov J. Phys. Conf. Ser. 741, 012081 (2016).
[29] D.R. Stinson, Congressus Numerantium 114, 7 (1996).
[30] T. Krovetz and P. Rogaway, Lect. Notes Comput. Sci. 2015, 73 (2000).
[31] N. Walenta, A. Burg, D. Caselunghe, J. Constantin, N. Gisin, O. Guinnard, R. Houlmann, P. Junod, B. Korzh, N. Kulesza, M. Legré, C.C.W. Lim, T. Lunghi, L. Monat, C. Portmann, M. Soucarros, P. Trinkler, G. Trolliet, F. Vannel, and H. Zbinden, New J. Phys. 16 013047 (2014).