Quantum Electronics **50** (5) 426–439 (2020)

https://doi.org/10.1070/QEL17283

On the operational meaning and practical aspects of using the security parameter in quantum key distribution

A.S. Trushechkin

Abstract. **We discuss the operational meaning of a commonly accepted security parameter in quantum key distribution, which is based on the trace distance. We separately consider the cases of using a key in a one-time pad and in a computationally secure cipher. Some practical aspects of using the security parameter are also elucidated, which are usually not paid enough attention in theoretical studies and which therefore may cause difficulties for experimentalists and engineers. It is shown that a one-time pad requires not only a higher key generation rate than computationally secure ciphers, but also a significantly stronger condition on the key security parameter.**

Keywords: *quantum cryptography, quantum key distributions, security parameter, trace distance.*

1. Introduction

Quantum key distribution (QKD) protocols allow two parties (we will call them legitimate) to generate a common private binary string (key) [1]. Unlike key distribution protocols based on public key cryptography, quantum cryptography provides theoretical security, i.e., security against even unlimited computing power of the adversary. If the security of the QKD protocol is proved mathematically, then the only way to break it is to attack the hardware. QKD has one more property that is distinctive: the key generated with its help cannot be broken after the protocol is completed. In contrast, ciphertexts generated using computationally secure ciphers (i.e., based on the assumption of limited computing capabilities of the adversary) can be decrypted later with the advent of new cryptanalysis algorithms or due to progress in computer technology. If, for example, a new type of attack on equipment is detected in the QKD systems, this does not help to find out the keys distributed up to this point.

The theoretical degree of security of the key generated in the QKD protocols is expressed in terms of the trace distance between the real classical-quantum state (in which the classical subsystem corresponds to the key, and the quantum one belongs to the adversary) and the corresponding ideal state, characterised by uniform distribution of the key and the absence of correlation between the key and the adversary's quantum subsystem. If the trace distance does not exceed ε (usually, ε is very small, e.g., about 10^{-10} or 10^{-11} [2–4]), then the key is called ε-secure. A key corresponding to an ideal classical-quantum state is called ideal, or perfectly secure.

The reasons why this particular security measure is most suitable are described, e.g., in Ref. [5], which is a review summarising the advent and development of the theory of *universally composable security* [6–11] and its adaptation to quantum cryptography [12–17]. This measure of security is universally composable, i.e., applicable in any context. Imperfect security of the key increases the probability of breaking a cryptographic protocol in which the key is used (application protocol), e.g., an encryption system. However, the application protocol itself may be also imperfect. Other measures of key security (for example, based on the Shannon amount of information) can lead to a situation when the combination of two small imperfections (key and application) can lead to a full compromise of the application protocol [18]. The use of a universally composable measure of security ensures that if the key is close to ideal, then whatever the application protocol, the degree of imperfection (also expressed in terms of the distance to the ideal implementation) increases as a result of using an imperfect key only by a small amount.

However, despite the theoretical attractiveness of the concept of universally composable security, the distance to the ideal state is a very unusual measure of security from a practical point of view. Therefore, it is advisable to give this security parameter generally accepted in the QKD some more practical operational meaning. In Ref. [5], this parameter is considered as the probability of failure in the quantum cryptography protocol, but the authors themselves acknowledge the conventionality of such interpretation. This interpretation was criticised by other authors [19–22]. Although this security criterion is well studied and a large number of its properties help understand its intuitive and operational meaning, in our opinion, these issues deserve further discussion, which is a subject of the present paper.

We believe that in order to give the security parameter a more practical operational meaning, it is useful to consider separately the combination of QKD with a one-time pad, i.e., with a theoretically secure cipher, and with computationally secure symmetric ciphers (Magma, Kuznyechik, AES, etc.).

A.S. Trushechkin Steklov Mathematical Institute, Russian Academy of Sciences, ul. Gubkina 8, 119991 Moscow, Russia; Russian Quantum Center, Skolkovskoe sh. 45, 121353 Moscow, Russia; National University of Science and Technology MISiS, Leninsky prosp. 4, 119049 Moscow, Russia; e-mail: trushechkin@mi-ras.ru

Received 17 February 2020
Kvantovaya Elektronika **50** (5) 426–439 (2020)
Translated by V.L. Derbov

In the first case, we base on the reduction of a set of possible plaintexts that occurs when replacing a completely secure key with an ε-secure one. In the second case, we base on the interpretation given in the footnote to the polemic note [23], in which the security parameter is related to an increase in the probability of breaking a cipher (or any other cryptographic application that uses a key) due to the key imperfection. In both cases, ε can be associated with the probability of breaking the QKD protocol, which confirms the existing intuitive meaning of this parameter.

In [24], the parameter ε is associated with the difficulty of enumeration for breaking a computationally secure symmetric cipher. In his fundamental work [25], Shannon introduced the concept of work characteristic for a computationally secure cipher as the average amount of computation required to find out the key. Assuming that there is no algorithm more efficient for breaking the used cipher than key enumeration (brute force attack), in Ref. [24] it is estimated how the average amount of computation on enumerating keys is reduced due to the use of an ε-secure key instead of a perfectly secure one. In this paper, we generalise this result to the case of the existence of more efficient methods of cryptanalysis of a cipher than simple enumeration of keys.

We will also consider the issue of an ε-secure key security in the case, when part of the key becomes known to the adversary. For example, this may occur in partially known plaintext attacks. In this situation the use of trace distance has been criticised by some authors [19–22]. We will elucidate what is exactly ensured and what is not ensured by ε-security in the sense of trace distance.

Section 3 is devoted to all these problems, i.e., the operational meaning of the security parameter. In Section 4, we consider some practical issues that directly follow from the properties of the trace distance, but which, in our opinion, are not paid enough attention in theoretical publications and therefore can cause difficulties for experimentalists and engineers. First, factors to be taken into account in the security parameter are considered. Then we discuss the generalisation of the security parameter for implementations of the QKD protocols, in which the key length is not specified in advance, but is calculated in the course of the protocol execution. Next, we present the rules for calculating the key security parameters obtained by splitting the key generated in the QKD session into parts, and vice versa, obtained by merging the keys obtained in different QKD sessions.

In the end of Section 4, examples of calculating the 'rapidity' of breaking the QKD protocol with realistic parameters are given. These examples also allow better navigation in choosing the value of ε for a particular practical situation. Moreover, it follows from them that a one-time pad requires not only a higher key generation rate than computationally secure ciphers, but also a significantly lower value of ε.

We begin with Section 2, which gives a definition of the security parameter in terms of a trace distance.

2. Definition of the security parameter

Recall the main stages of any QKD protocol [1, 3, 26].

1. Transmission of quantum states between legitimate parties and their measurement. As a result, the legitimate parties form binary strings called *raw keys*.

2. Announcement of information provided by the protocol (position numbers at which the signal was registered, bases, etc.). Elimination of positions stipulated by the protocol (in which registration did not occur, bases did not match, etc.). The binary strings after the end of this stage are the *sifted keys*.

3. Error correction. Binary strings after the end of this stage are *corrected keys*. Often this stage ends with verification that the corrected keys of the legitimate parties are likely to match. For verification, special hash function families are used. From this moment, we can talk about one common key of legitimate parties (with a low probability of noncoincidence).

4. Estimation of the degree of adversary intervention and the decision to generate a key or abandon it (protocol abortion) based on the observed data.

5. Privacy amplification. At this stage, a hash function is applied to the corrected key, which converts it into a key of shorter length, about which the adversary has only negligible information with high probability. This is expressed precisely through the concept of ε-security, the definition of which is given below. The result of this stage and the entire protocol is the *final* or *secret key*.

6. Authentication of all communication through the classical channel. To simplify, we can say that this is a protection against the man-in-the-middle attack, when the adversary communicates through the classical channel and distributes a key to each of the legitimate parties on behalf of the other legitimate party, i.e., this is protection against adversary intervention in the classical communication channel. In quantum cryptography, theoretically secure message authentication codes are usually used, i.e., the codes whose security does not depend on the computing capabilities of the adversary. They are based on the properties of special families of hash functions and imply that before the start of the session, the legitimate parties have a common short secret key (e.g., from the previous session). If intervention is detected in classical communication channel, the protocol is considered aborted and the generated key is destroyed.

In various implementations of the QKD protocols, the order of the steps described may vary, but this is not essential. To avoid confusion, we note that the protocol uses three families of hash functions: for verification, for privacy amplification, and for authentication. These stages differ in their goals; therefore, families of hash functions are generally different.

Consider the stage of making a decision about generating a key. There are two options for its implementation. In the first option, a decision is made about the possibility of generating a key of a predetermined length. If this is impossible, the protocol is aborted. In the second option, the maximum possible key length is calculated from the observed data. Failure to generate a key in this case corresponds to a key of zero length. We will consider this option in Section 4. As will be shown below, in a certain sense it reduces to the case of a fixed key length. Prior to this, we will consider the first option, i.e., generating a key of a predetermined length L.

In this case, as a result of executing the QKD protocol and the adversary's attack, a state (positive kernel operator with a unit trace) of the form

$$\rho_{KE} = p_\perp |\perp\rangle_K\langle\perp| \otimes \rho_E^\perp + (1-p_\perp) \sum_{k\in\{0,1\}^L} p_k |k\rangle_K\langle k| \otimes \rho_E^k \qquad (1)$$

arises (see [5]), where $0 \leqslant p_\perp \leqslant 1$ (probability of protocol abortion); $\sum_{k\in\{0,1\}^L} p_k = 1$ (total probability of any possible key value); K is the register used to record k or the sign $\perp$, which means aborting the protocol and refusing to generate the key (the Hilbert space of the register $\mathbb{C}^{2^L+1}$); and ρ_E^k are positive operators in the Hilbert space of the adversary $\mathscr{H}_E$ with a dimension unknown in advance. Note that here we neglect the probability that the keys of the legitimate parties may be not coincident in order to focus on the issue of the degree of the key security; therefore, we indicate only one register K with the key k. The probability of a mismatch between the keys of legitimate parties will be discussed in Section 3.4. Abortion of the protocol can occur when excessive intervention in the quantum channel is detected in stage 4 or when intervention in the classical communication is detected in stage 6. Since the adversary's attack is unknown to legitimate parties, they also do not know exactly what state of the form (1) takes place. On the contrary, the adversary knows what kind of attack he carried out, and therefore we believe that he knows state (1). However, this does not mean that he knows exactly which value of the key k was implemented.

The ideal state corresponding to the real state (1) has the form

$$\rho_{KE}^{\text{ideal}} = p_\perp |\perp\rangle_K\langle\perp| \otimes \rho_E^\perp + 2^{-L}(1-p_\perp)\Big(\sum_{k\in\{0,1\}^L} |k\rangle_K\langle k|\Big)\otimes\rho_E^\top, \tag{2}$$

where

$$\rho_E^\top = \sum_{k\in\{0,1\}^L} p_k \rho_E^k. \tag{3}$$

State (2) is characterised by the fact that if a decision is made to generate a key, then the key is distributed evenly and does not correlate with the state of the adversary. In this sense, it is ideal. Correspondence to state (1) consists in the coincidence of the protocol abortion probabilities and the reduced states of the adversary subsystem: $\text{Tr}_K \rho_{KE}^{\text{ideal}} = \text{Tr}_K \rho_{KE}$, where Tr is the trace, Tr_K is the partial trace in the space of the register K.

The QKD protocol and, accordingly, the key generated by this protocol is called ε-secure if, for any adversary attack (from the considered class of attacks), the following condition is satisfied:

$$D(\rho_{KE}, \rho_{KE}^{\text{ideal}}) \leqslant \varepsilon, \tag{4}$$

where

$$D(\rho,\sigma) = \frac{1}{2}||\rho-\sigma||_1 = \frac{1}{2}\text{Tr}\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)} = \frac{1}{2}\sum_i |\lambda_i|$$

is the trace distance between arbitrary states ρ and σ; λ_i are the eigenvalues of the operator $\rho-\sigma$ (with the multiplicities taken into account). If the operators ρ and σ are simultaneously diagonalisable, then the trace distance reduces to the distance in variation between the two probability distributions:

$$D(\rho,\sigma) = \frac{1}{2}\sum_i |p_i - q_i|, \tag{5}$$

where p_i and q_i are the eigenvalues of the operators ρ and σ, respectively.

Remark 1. 'More ideal' would be the state

$$\rho_{KE}^{\top,\text{ideal}} = 2^{-L}\sum_{k\in\{0,1\}^L} |k\rangle_K\langle k| \otimes \rho_E^\top, \tag{6}$$

which is characterised by the fact that the legitimate parties generate the key with certainty, i.e., $p_\perp = 0$. However, in this case inequality (4) when replacing ρ_{KE}^{ideal} with $\rho_{KE}^{\top,\text{ideal}}$ in the general case will not be fulfilled: the adversary can always make legitimate parties refuse to generate a key. To this end, he can, e.g., carry out intense eavesdropping of the quantum channel, so that the key generation becomes impossible, or simply block all the transmitted quantum states. Therefore, we cannot ensure the proximity of state (1) to (6) and are forced to restrict ourselves to ensuring proximity to state (2).

Another important point is associated with the probability $p_\perp$ of refusal to generate the key. Consider the state

$$\rho_{KE}^\top = \sum_{k\in\{0,1\}^L} p_k |k\rangle_K\langle k| \otimes \rho_E^k. \tag{7}$$

Then

$$D(\rho_{KE}, \rho_{KE}^{\text{ideal}}) = (1-p_\perp) D(\rho_{KE}^\top, \rho_{KE}^{\top,\text{ideal}}),$$

and inequality (4) can be rewritten in the form

$$(1-p_\perp) D(\rho_{KE}^\top, \rho_{KE}^{\top,\text{ideal}}) \leqslant \varepsilon, \tag{8}$$

or

$$D(\rho_{KE}^\top, \rho_{KE}^{\top,\text{ideal}}) \leqslant \varepsilon' = \varepsilon/(1-p_\perp). \tag{9}$$

The states $\rho_{KE}^\top$ and $\rho_{KE}^{\top,\text{ideal}}$ are conditional quantum states if the protocol was not aborted and the key was generated. If the key is generated, legitimate users are interested in the distance between these states, i.e., $D(\rho_{KE}^\top, \rho_{KE}^{\top,\text{ideal}})$, and not between states that include the unrealised probability of protocol abortion. However, as follows from (9), the distance $D(\rho_{KE}^\top, \rho_{KE}^{\top,\text{ideal}})$, generally speaking, does not have to be small. Intensive eavesdropping of the quantum channel will lead to a large correlation of the key and the adversary subsystem, so that the distance between these two states will be large. For example, if we consider the QKD BB84 protocol, then the adversary can carry out the simplest intercept–resend attack and guess all bases with nonzero probability. Thus, he will receive complete information about the key, without introducing any error. However, in this case, with a high probability, the legitimate parties will detect eavesdropping (thus, in our example, the probability of guessing all the bases is very small) and will conclude that it is impossible to generate a private key, so that the probability $1-p_\perp$ will be low, thus ensuring smallness of (8).

Inequality (8) ensures that either the adversary has practically no information about the key [the value of $D(\rho_{KE}^\top, \rho_{KE}^{\top,\text{ideal}})$ is small], or an extremely unlikely event of intense eavesdropping nondetection occurred. Parameter ε', characterising the distance between conditional quantum states under the con-

dition of key generation, will be also often used. We will return to the problem of taking into account the unknown probability $1 - p_\perp$ in Section 3.4.

3. Operational meanings of the security parameter

3.1. Indistinguishability, increased probability of application protocol breaking, universal composability

The main operational sense of the security parameter definition in terms of a trace distance consists in the following: the probability of successfully distinguishing any quantum states ρ and σ (provided that their *a priori* probabilities are 1/2) cannot exceed $(1 + D(\rho, \sigma))/2$, i.e., it exceeds the probability 1/2 of simple guessing by a small amount if $D(\rho, \sigma)$ is small [5]. By distinguishing we mean the measurement of the observable, which is given by two positive operators, M_ρ and $M_\sigma = I - M_\rho$ (I is the identity operator). The outcome corresponding to the first (second) operator is interpreted as supplying the state ρ (σ), respectively, to the input of the distinguishing device. Then the probability of distinguishing is the average probability that the input state is determined correctly:

$$P_{\text{dist}} = \frac{1}{2}[\text{Tr}(\rho M_\rho) + \text{Tr}(\sigma M_\sigma)].$$

The estimate

$$P_{\text{dist}} \leqslant \frac{1 + D(\rho,\sigma)}{2} \tag{10}$$

is exact, i.e., there is always a dimension that turns this inequality into equality. The value $2P_{\text{dist}} - 1$ is called *distinguishing advantage*. Thus, the trace distance between two quantum states is equal to the maximum gain in their distinguishability.

This property has important implications for QKD. To formalise the concept of distinguishability in the theory of universally composable security, the role of *a distinguisher* is introduced, which selects the input data for the protocol, performs all the actions of the adversary, receives the protocol output (in our case, the key), and then performs any actions with this output. A detailed description of the theory can be found, for example, in [5, 11] and in earlier works, referenced above in the beginning of the article. To simplify, as applied to the QKD protocol, we can say that the ddistinguisher first plays the role of an adversary and implements an arbitrary attack, after which it is given with 1/2 probabilities either the real state ρ_{KE} resulting from the execution of the protocol and attack on it, or the corresponding ideal state ρ_{KE}^{ideal}. The task of a ddistinguisher is to guess which of these two states is given to it, i.e., to distinguish between these states. For this purpose, it can perform any action with the provided classical-quantum state. For example, it can execute an arbitrary protocol for encrypting a private message using a key recorded in the register K and simulate an adversary's attack on the cryptographic system, taking into account partial knowledge of the key by the adversary. For example, if a cipher is broken, while with an ideal key the breaking has a very low probability or is not possible at all, then the ddistinguisher will conclude that it was given a real state.

The arbitrary use of the key in any cryptographic application, together with the attack of this application by the adversary, taking into account partial knowledge of the key and the subsequent decision about which of the two states took place, is a specific, complexly organised measurement that distinguishes between these states. Therefore, estimates (4) and (10) indicate that it is impossible in any way to distinguish a real state from an ideal one with a probability exceeding the probability of a simple guess 1/2 by more than $\varepsilon/2$.

Let us draw another important conclusion from these considerations. Suppose that the probability of breaking an application protocol using an ideal key using a specific attack is p. This probability can be affected by both the characteristics of the application protocol and the *a priori* information of the adversary. For example, if the application protocol is a symmetric encryption system, then the probability p may decrease if the adversary becomes aware of a portion of the plaintext. If this cipher system is secure only in the computational sense, then the p can be defined as the probability of breaking at the expense of a certain amount of computation. For example, if a more efficient attack on a cipher than a complete enumeration of all keys (brute force attack) is not known, then enumerating a fraction of p of the entire set of keys will yield a probability of breaking p (the probability that one of the keys will work). If the key is not perfect, then this probability increases, because the adversary can begin the enumeration from the most probable versions (for more details see [24]).

An application protocol can be launched only when the key is generated, i.e., the conditional states $\rho_{KE}^{\top}$ and $\rho_{KE}^{\top,\text{ideal}}$ should be considered. According to Eqn (9), the trace distance between them is $\varepsilon' = \varepsilon/(1 - p_\perp)$.

Proposition 1. *Let p be the probability of breaking some protocol using the private key as a result of some attack, provided that the ideal key specified by state (6) is used. Then, when using an ε-secure key that corresponds to state (7), this probability does not* exceed $p + \varepsilon/(1 - p_\perp)$.

This proposition is illustrated in Fig. 1 for enumeration of keys as an example.

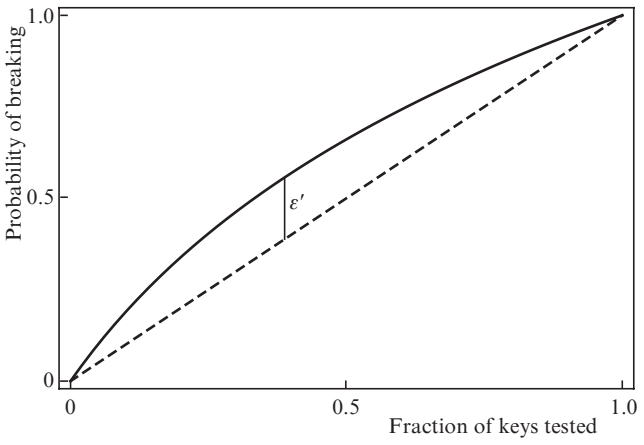

Figure 1. Probability of breaking a computationally secure symmetric cryptosystem by enumeration of keys (brute force attack) versus the fraction of keys tested. The dashed line is the case of an ideal key, i.e., completely private and evenly distributed, described by state (6); the solid curve is the case of an ε-secure key described by state (7), which satisfies inequality (9). The maximum displacement of the solid line from the dashed along the ordinate axis does not exceed $\varepsilon' = \varepsilon/(1 - p_\perp)$. The figure illustrates Proposition 1; in the case of other attacks, the dependence for the ideal key is not necessarily linear.

Proof of Proposition 1. We prove it from the contrary. Let the probability of breaking the cipher system increase by $\varepsilon_1/(1-p_\perp)$, due to replacing an ideal key with an ε-secure one, where $\varepsilon_1 > \varepsilon$. Let us show that then we can distinguish between the states ρ_{KE} and ρ_{KE}^{ideal} with a probability exceeding $(1+\varepsilon)/2$, which contradicts (10). The algorithm for the ddistinguisher is as follows:

1. The ddistinguisher performs a binary measurement specified by the projector $|\perp\rangle\langle\perp|\otimes I_E$ (I_E is the identity operator in the space $\mathscr{H}_E$), i.e., a measurement of whether the key is generated. If the key has not been generated, which happens with probability $p_\perp$, the ddistinguisher is unable to determine what state is given to him, and is forced to resort to simple guessing. Therefore, in this case, the probability of a correct answer is $P_{\text{dist}}^{\perp} = 1/2$.

2. If the key is generated, which happens with a probability $1-p_\perp$, then the ddistinguisher, playing the role of legitimate parties, measures the register K, obtains the key as a result, and launches the application protocol with this key. Then, as an adversary, he attacks the application protocol. If the breaking was successful, then the ddistinguisher decides that it was given the state ρ_{KE}, otherwise, the state ρ_{KE}^{ideal}. Then, since the probabilities of breaking the application protocol with ideal and real key are p and $p+\varepsilon_1/(1-p_\perp)$, respectively, the probability of a correct answer is

$$P_{\text{dist}}^{\top} = \frac{1}{2}\left[\left(p+\frac{\varepsilon_1}{1-p_\perp}\right)+(1-p)\right] = \frac{1}{2}\left(1+\frac{\varepsilon_1}{1-p_\perp}\right).$$

The probability of a correct answer, averaged over the outcomes of algorithm 1 and 2, is

$$P_{\text{dist}} = p_\perp P_{\text{dist}}^{\perp} + (1-p_\perp)P_{\text{dist}}^{\top} = \frac{1+\varepsilon_1}{2} > \frac{1+\varepsilon}{2},$$

which contradicts (10).

This interpretation of the security parameter is given only in the footnote in the polemic note [23], but, in our opinion, is one of the most important interpretations: Proposition 1 allows us to interpret $\varepsilon' = \varepsilon/(1-p_\perp)$ [and not Eqn (20) below, as is often believed] as the probability of key breaking. Indeed, let suppose that the application protocol is practically unbreakable. For example, when attacking a cipher with realistic amounts of computation, the probability of breaking p is practically zero [formally $p \ll \varepsilon/(1-p_\perp)$]. Then the probability of breaking the application protocol using the ε-secure key will be $p+\varepsilon/(1-p_\perp) \approx \varepsilon/(1-p_\perp)$. In a sense, this value can then be considered the probability of breaking the key, since it is precisely the key nonideality that allows breaking the application protocol.

We mentioned the application of Proposition 1 to the case of using a key in a computationally secure cipher. This proposition can be applied, e.g., to the case of using the key in theoretically secure message authentication codes, which imply a common private key possessed by the legitimate parties. If p is the probability of collision of hash functions used in the code with a perfectly secure key, then the probability of collision with an ε-secure key will not exceed $p+\varepsilon'$.

To simplify, we can say that Proposition 1 is exactly the one containing the universal composability of the security parameter: if the application protocol is also nonideal, the degree of its nonideality due to the use of a nonideal key increases by a small amount that can be estimated. This property allows exploring the security of the application protocol with an ideal key and the security of the key distribution protocol separately. Using other measures of the degree of the key nonideality instead of the trace distance, e.g., those based on the Shannon amount of information, can result in complete compromise of the entire system due to combination of two nonidealities (of the key and application protocol) [18].

3.2. Relation of the security parameter with the performance characteristic of a computationally secure cryptosystem

We continue considering the case when the generated key is used in a computationally secure symmetric cryptosystem. In the fundamental work of Shannon [25], the concept of the performance characteristic of such a cryptosystem as the average amount of computation required to determine the key is introduced. In Ref. [24], a formula was obtained that relates the security parameter to the average amount of operations on enumerating keys (that is, for brute force attack). If, with an ideal key, the average number of keys that must be considered to decrypt the message is $(2^L+1)/2$, then in the case of an ε-secure key this value in our notation will be $[(2^L(1-2\varepsilon')+1]/2$, where $\varepsilon' = \varepsilon/(1-p_\perp)$.

Now let us generalise the results to the case of existence of more efficient attacks than the brute force one. Let $p(T)$ be the probability of breaking the cipher with expenditure of computing time T provided that the legitimate parties use the ideal key. With simple enumeration of the keys, if T is measured by the number of keys being tested, $p(T) = T/2^L$. Then from Proposition 1 it follows that when using an -secure key this probability does not exceed $p(T)+\varepsilon'$. Here, for simplicity, we consider only the computing time, but we can also include other characteristics that make breaking difficult.

Consider the inverse function $T(p)$, i.e., the amount of computation necessary to ensure the probability of breaking

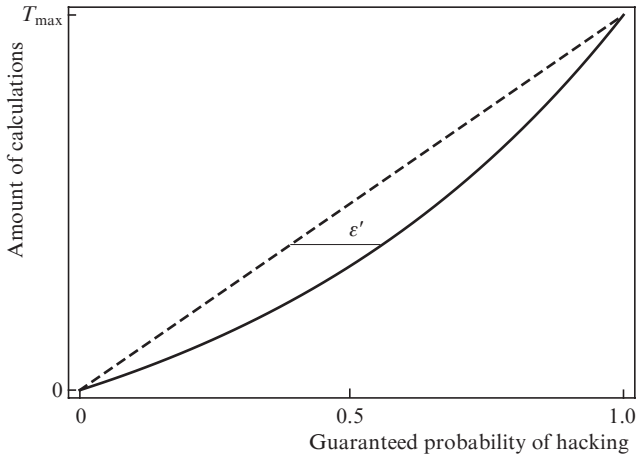

Figure 2. Amount of computations needed to ensure a given probability of breaking a computationally secure symmetric cryptosystem by key enumeration (brute force attack) – the functions that are inverse (up to a factor) to those shown in Fig. 1; $T_{\max}$ is the amount of computations required to break the cipher for sure. If it is measured in the number of keys examined, then $T_{\max} = 2^L$. As in Fig. 1, the dashed line is the case of an ideal key, the solid curve is the case of an ε-secure key, $\varepsilon' = \varepsilon/(1-p_\perp)$. The figure illustrates Eqn (11). In the case of other attacks, the dependence corresponding to the ideal key is not necessarily linear.

p when using the ideal key. Denote by $T(p)$ the corresponding amount of computations when using the ε-secure key. Then the last statement of the previous paragraph can be rewritten as follows:

$$T_\varepsilon(p) \geqslant T(p - \varepsilon'). \tag{11}$$

Formula (11) is illustrated in Fig. 2. This is a general formula relating the complexity of breaking and the security of the key in terms of trace distance: when legitimate parties use the ε-secure key, the laboriousness of breaking can be estimated from below via the laboriousness in the case of an ideal key, but it provides smaller by ε' probability of breaking.

3.3. Using the QKD with a one-time pad

We examined the operational meanings of the security parameter when combining the QKD with computationally secure ciphers. Currently, this often takes place in practice, since the QKD rate is still not high enough. However, the original idea of quantum cryptography is to provide a theoretically secure private communication, i.e., to combine a theoretically secure QKD with a theoretically secure cipher, i.e. a one-time pad. Since this cipher is not breakable when using an ideal key, the previous operational interpretations of the trace distance associated with the probability of breaking are not suitable. In our opinion, in this case the following interpretation may be useful.

Let $\mathcal{M} \subset \{0, 1\}^L$ be the set of possible plaintexts known to the adversary. This can be a set of all kinds of binary strings of length L, a set of all meaningful texts (in binary representation), as well as some subset of the set of meaningful texts, which may reflect the a priori knowledge of the adversary about which messages could be encrypted. For example, if the adversary knows that one of the two possible messages is encrypted, then $\mathcal{M}$ contains two elements. We denote the power of this set by M. When encrypted with a one-time pad with an ideal key, this set does not change, because the cipher is perfectly secure. However, if an ε-secure key is used, then this set can be reduced. For example, for some keys k it is possible that $p_k = 0$, so that, having read the ciphertext, the adversary will exclude from $\mathcal{M}$ those plaintexts that correspond to keys with zero probabilities. We denote by $M' < M$ the number of possible plaintexts after the adversary has read the ciphertext in state (1). The adversary has no way to find out which of these plaintexts was actually sent. Note that M' is a random variable, since it depends on the ciphertext and, therefore, on the key generated by legitimate users (in accordance with the distribution $\{p_k\}$), as well as on the random result of the adversary measuring his subsystem. We also denote the number of excluded plaintexts by $M'' = M - M'$.

Now we proceed to a question of how much M' is less than M. Since $\log_2 M$ and $\log_2 M'$ are the Hartley entropies of the plaintext, the decrease

$$I = \log_2 M - \log_2 M' = \log_2 \frac{M}{M'} \tag{12}$$

can be interpreted as the amount of Hartley information received the adversary as a result of reading the ciphertext. We will show that with high probability this value is approximately equal to ε'.

Before formulating the general proposition, we consider two examples. Let $\rho_E^k = \rho_E$, i.e., the adversary subsystem does not correlate with the key, but the adversary attack or equipment imperfection leads to an uneven distribution p_k. If the key is ε-secure, then

$$\sum_{k \in \{0,1\}^L} |p_k - 2^{-L}| \leqslant \varepsilon' = \varepsilon/(1 - p_\perp);$$

therefore, $p_k = 0$ can be fulfilled for a maximum of $2^L\varepsilon'$ keys (up to rounding down). For realistic values of L and ε, e.g., $L = 10^6$, $\varepsilon = 10^{-11}$, this value is greater than unity. Let $\mathcal{M}$ be the entire set of binary strings of length L, i.e., $M = 2^L$. Then $2^L\varepsilon'$ messages are reliably excluded from the set $\mathcal{M}$, i.e., $M' = 2^L - 2^L\varepsilon' = 2^L(1 - \varepsilon')$. This means that $M''/M = \varepsilon'$, i.e., the percentage of excluded open messages is very small. In terms of the amount of Hartley information we have

$$I = \log_2(1 - \varepsilon')^{-1} \approx \varepsilon'/\ln 2.$$

Now let us consider another extreme case: $M = 2$, i.e., when one of two possible messages is encrypted. If the key corresponding to the second message belongs to the excluded set with power $2^L\varepsilon'$, then $I = 1$ and the adversary decrypts the message. However, the probability of a fixed key belonging to the set $2^L\varepsilon'$ is equal to ε'. This means that the security criterion based on the trace distance provides security only in a probabilistic sense. This also manifests itself, e.g., in Eqn (8): the generated key is unsecure if a low-probability event of an undetected intense attack occurs.

We now prove the general proposition.

Proposition 2. *Let the ε-secure key corresponding to state (1) be used for encryption with a one-time pad. For any plaintext from the set $\mathcal{M} \subset \{0,1\}^L$ of power M for the mean value $\overline{M''}$ of a random variable M'', the number of plaintexts excluded after reading the ciphertext by the adversary has, the estimate*

$$\overline{M''} \leqslant 2\varepsilon' M, \tag{13}$$

is valid, where $\varepsilon' = \varepsilon/(1 - p_\perp)$.

Markov's inequality allows proceeding from estimating the average number of excluded plaintexts to probabilistic assessment with individual outcomes (the key and the outcome of the adversary's measurement of his subsystem).

Consequence. *Under the conditions of Proposition 2,*

$$\Pr\left[\frac{M''}{M} > \delta\right] \leqslant \frac{2\varepsilon'}{\delta}. \tag{14}$$

holds.

Thus, the ratioM'/M can significantly differ from unity [respectively, Hartley information (12) can differ from zero] only with a probability of the order of '. If a significant difference of Hartley information from zero is considered key breaking (the adversary obtained substantially nonzero information about the message due to a nonideal key), then ε' can be considered the probability of breaking the key in order of magnitude. However, the value of $2\varepsilon'/\delta$ for a specific and minimal δ acceptable for users characterises the probability of breaking with more accuracy.

Proof of Proposition 2. As in the proof of Proposition 1, we reduce the problem to distinguishing between the states

$\rho_{KE}^{\top}$ and $\rho_{KE}^{\top,\mathrm{ideal}}$, i.e. to Eqn (2). Consider the following algorithm. The ddistinguisher imitates both the actions of legitimate users and the actions of an adversary in order to encrypt and decrypt a private message. Acting as legitimate users, the ddistinguisher measures the register K, obtains the key as a result, and then encrypts the specified plaintext with it. Then, playing the role of the adversary, he measures his subsystem taking into account the known ciphertext and writes out excluded plaintexts. Since we are considering theoretical security, the ddistinguisher has unlimited computing power. Then it (already as a ddistinguisher as such) finds out whether the encrypted open message belongs to the set of excluded open messages. If the answer is yes, he concludes that he was given the ideal state $\rho_{KE}^{\mathrm{ideal}}$, and, otherwise, the real state ρ_{KE}.

Thus, the algorithm always identifies the real state as real, i.e., in the case of a real state, the probability of the correct answer is unity. For ideal state, the probability of a correct answer is equal to the probability that the key is one of many excluded. Since in the ideal state the distribution of keys is uniform, this probability is M''/M. M'' is a random variable, so we need to take its average value. However, averaging is performed not over the real state, but over the ideal one; we denote this mean value by $\overline{M''}^{\mathrm{ideal}}$. We emphasise that the excluded keys are written out based on the measurement result as if the state were real, because the ddistinguisher does not know what state it was given. But since we are considering the case when the ideal state is given, averaging is performed over the ideal state. Strictly speaking, we consider another random variable, but for simplicity we also denote it by M''. Then the probability of distinguishing is

$$P_{\mathrm{dist}}^{\top} = \frac{1}{2}\left(1 + \frac{\overline{M''}^{\mathrm{ideal}}}{M}\right) \leqslant \frac{1+\varepsilon'}{2},$$

where the last inequality holds due to inequality (10). Hence

$$\overline{M''}^{\mathrm{ideal}} \leqslant \varepsilon' M. \tag{15}$$

We now estimate $\overline{M''}/M \leqslant \varepsilon'$, where averaging in the numerator occurs already over the real state. The random variable M'' is a result of the measurement of the state, $\rho_{KE}^{\top,\mathrm{ideal}}$ or $\rho_{KE}^{\top}$, the trace distance between which does not exceed ε'. The variation distance between the distributions obtained by measuring the same observable over these states also cannot exceed ε' [5]. Since $0 \leqslant M'' \leqslant M$,

$$\overline{M''} \leqslant \overline{M''}^{\mathrm{ideal}} + \varepsilon' M.$$

From this and Eqn (15) the desired inequality (13) follows.

Remark 2. In the previous subsections, we considered arbitrary probability distributions on the set of keys from the point of view of the adversary after the attack performed by him. Here we took into account only the fact that some keys can have zero probability for the adversary, which leads to the exclusion of some plaintexts. If we take into account here that probable keys generally have different probabilities, then plaintexts will also have different probabilities, i.e., from the point of view of the adversary, one plaintext after an attack will be more probable than another. In addition, initially (before the attack) one plaintext may be more probable for the adversary than another due to various reasons.

We did not address such a level of consideration, since the meaning of the probabilities of plaintexts depends on the further actions of the adversary, which goes beyond cryptography. Suppose that as a result of an attack the adversary learns that the probability is 0.6 for one plaintext and 0.4 for the other. He can bet that the plaintext that has the greatest probability has been sent and not take into account that another text could have been sent. Then probability 0.6 acquires an objective statistical meaning: in the course of multiple repetitions of this situation (i.e., a session of the QKD and the corresponding attack) in 6% of cases, the adversary will behave basing on a correctly decrypted text. However, the adversary may keep in mind both these options rather than discard plaintext having lower probability, remembering, however, that one option is more likely, and plan his activity based on the subjective perception of these probabilities. In this case, it is already difficult to assign an objective statistical meaning to these probabilities.

However, we can generalise our reasoning to the case when the adversary discards keys not only with zero, but also with sufficiently low probability $p_k < p_{\mathrm{crit}}$, where p_{crit} is the given critical value. Then Proposition 2 and the consequence from it remain valid.

3.4. Accounting for the probability of key generation

In Propositions 1 and 2 and in formulae (11) and (14), the distance between the conditional quantum states $\rho_{KE}^{\top}$ and $\rho_{KE}^{\top,\mathrm{ideal}}$ corresponding to the already generated key is characterised by the quantity $\varepsilon' = \varepsilon/(1-p_{\perp})$ rather than by ε. The probability $p_{\perp}$ is unknown to us, however, due to the probabilistic interpretations of the quantity ε' that we obtained in Subsections 3.1 and 3.3, this does not create a problem for us. In the notation of Proposition 1, let us consider instead of the value $p + \varepsilon'$ (recall that p is the probability of breaking the application protocol, for example, the cipher, with a certain attack and, possibly, by cost of certain computing resources), the value

$$(1-p_{\perp})(p+\varepsilon') = p(1-p_{\perp}) + \varepsilon \leqslant p + \varepsilon.$$

In the left-hand side there is a joint probability that the protocol will not be aborted, and the cipher will be broken, the right-hand side of the inequality is an estimate of this probability from above. Similarly,

$$(1-p_{\perp})2\varepsilon'/\delta = 2\varepsilon/\delta$$

is the joint probability that the protocol will not be aborted and the adversary will be able to exclude from the set of possible open texts a fraction exceeding δ (e.g., δ can be the maximum allowable fraction). Both events correspond to the breaking of QKD protocol: nondetection of eavesdropping and generation of a key, the use of which led to breaking the cipher. The probabilities ε in the first case (for $p \ll \varepsilon$) and $2\varepsilon/\delta$ in the second case will be referred by us to as the probabilities of breaking the QKD protocol, in contrast to the probabilities of breaking the key, where ε' was used instead of ε.

Let us explain the meaning of the probability of breaking the protocol of the QKD by an example. Let this probability be, for example, 10^{-11} and let the QKD protocol sequentially run an unlimited number of times. Suppose, in some cases, the protocol will be aborted due to actual eavesdropping,

noise fluctuations, malfunctions, or loss of hardware settings. In these sessions, the security of the key (if it were generated) does not interest us. In some cases, the protocol will not be aborted and a key will be generated, but the adversary will not be able to break the cipher. In some cases, the protocol will not be aborted, but the adversary will be able to break the cipher. This is exactly what corresponds to breaking the protocol of the QKD. The probability of breaking the QKD protocol ε allows evaluating how often this will happen. The probability $\varepsilon = 10^{-11}$ means that such an event will occur on average once in 10^{11} key distribution sessions. Of course, due to protocol abortions, fewer keys than 10^{11} will be generated during this time, i.e., interpretation of ε in terms of the average frequency of sessions leading to breaking the cipher does not depend on $1 - p_\perp$. The probability $1 - p_\perp$ (in practice, in the absence of interception, it should be close to unity) only affects how many keys we will generate in the interval before the next break. We will return to calculations with realistic parameter values in Section 4.4.

At the same time, in Eqn (11), which relates the security of the key to the performance characteristic of the cipher, the quantity ε' is retained. This can be understood as follows: in this case we are considering the difficulty of breaking a cipher with one fixed key rather than a large number of sessions and keys; hence, it is just the degree of security of a particular key ε' that enters the formula, and therefore it is necessary to estimate $1 - p_\perp$ from below. A small value of $1 - p_\perp$ means that with high probability the protocol should have been aborted. Therefore, if in practice the protocol is aborted rather rarely, this value cannot be small simultaneously in all sessions. However, its smallness cannot be ruled out in a single selected session. Then the legitimate parties can determine the threshold probability α (an analogue of the significance level in mathematical statistics) and assume that the event that happened could not have too little probability (less than α). Then we can put $1 - p_\perp \geqslant \alpha$ if the protocol was not aborted. Probably, the need for such an estimate suggests that the interpretation of the security parameter in terms of the breaking difficulty is not very convenient; the interpretation in terms of the breaking probability is preferable due to the possibility of multiplying the probabilities and interpreting them statistically.

It is important to note that the problem of key degradation does not impede this statistical interpretation. This problem is as follows: the theoretically secure authentication codes for classic messages used in the QKD protocol suggest that the legitimate parties have an initial short private key. When it is exhausted, part of the key distributed in one of the preceding sessions, i.e., an imperfect key, is used for authentication, which reduces the security of the new key. This new key is used to authenticate the next sessions, resulting in the repeated decrease in security of new keys, etc.

From the position of universally composable security, the problem of key degradation was first considered in [12]. This paper considers a chain of n sessions of the QKD protocol, in which part of the key generated in one session is used for authentication in the next session, and the other (larger) part is used for useful applications, for example, encryption. It is shown that if the QKD protocol with a perfectly secure key for authentication is ε-secure, then the whole chain is $n\varepsilon$-secure. The value of $n\varepsilon$ is also determined in terms of the trace distance between the real classical-quantum state (the result of the work of all sessions) and ideal state. This quantity can be given a probabilistic meaning, too. If a session is aborted, then the subsequent sessions of the chain will not be started since there is no authentication key. However, it is important that the sessions in which the keys were generated would not be broken. As before, breaking a session of the QKD protocol can be understood as the realisation of the probability of Proposition 1 or formula (14). Then $n\varepsilon$ is the probability of breaking at least one session that was launched and ended with the generation of a key. In other words, if we start a chain many times, then this undesirable event will occur on average once per $(n\varepsilon)^{-1}$ starts of the chain. If every n sessions a 'fresh' (perfectly secure) authentication key is taken, i.e., the chain is really launched sequentially a large number of times, then this proposition is equivalent to the previous one: on average, one of $1/\varepsilon$ sessions is broken.

From these considerations, an important practical conclusion follows: *when you abort a QKD session, the next session must be started with a fresh (perfectly secure) key for authentication*, since the chain security has been proved precisely in this setting. This conclusion seems counterintuitive: if the adversary intensively eavesdropped a given session of the QKD, which led to its abortion, this does not make the keys generated in previous sessions less secure. However, we emphasise once again that the security parameter ε actually characterises the degree of security of the QKD session rather than the key security; this parameter includes the unknown probability of protocol abortion as a factor. For example, the adversary could carry out more intensive eavesdropping than usual, not only in the last session, but also in the penultimate one, but then this would not lead to the abortion of the protocol. Since probability acquires an objective statistical meaning in multiple repetitions of tests, any judgment regarding the interpretation of the degree of security ε should be based on rigorous probabilistic and statistical reasoning associated with a large number of QKD sessions. These considerations make it necessary to take a fresh key after some session of the QKD was aborted.

Finally, we note that the probability of protocol abortion is also involved in assessing the probability of noncoincidence of keys of the legitimate parties. With this probability the second parameter (in addition to ε), characterising the quality of the key pair [2, 5, 27], is associated. Typically, the assessment of the key mismatch probability is based on the properties of the hash functions used in the verification procedure (see the description of the stages of the QKD protocol). Let K_A and K_B be the corrected keys of the legitimate parties. To make sure that they coincide, the same hash function F is applied to them, randomly selected from a special family of hash functions, which ensures that the probability of hash functions coincidence under the condition of key mismatch is small:

$$\Pr[F(K_A) = F(K_B) \,|\, K_A \neq K_B] \leqslant \varepsilon_{\text{cor}} \tag{16}$$

for a given small ε_{cor}. The subscript cor (correctness) means the requirement that the keys of legitimate parties are coincident with high probability. However, in practice, with the keys generated, we are interested in another probability,

$$\Pr[K_A \neq K_B \,|\, F(K_A) = F(K_B)],$$

i.e., the probability of key mismatch provided that the hash functions coincide. If the hash functions do not coincide, the protocol is aborted. However, in order to evaluate this probability, it is necessary to know the probability

$\Pr[F(K_A)\neq F(K_B)]= p_\perp^{\text{ver}}$, i.e., the probability of protocol abortion due to mismatch of verification hash functions:

$$\begin{aligned}\Pr[K_A \neq K_B \,|\, F(K_A) = F(K_B)] &= \Pr[K_A \neq K_B \wedge F(K_A) \\ &= F(K_B)]/(1 - p_\perp^{\text{ver}}) = \Pr[F(K_A) = F(K_B) \,|\, K_A \neq K_B] \\ &\times \Pr[K_A \neq K_B]/(1 - p_\perp^{\text{ver}}) \leqslant \Pr[F(K_A) \\ &= F(K_B) \,|\, K_A \neq K_B]/(1 - p_\perp^{\text{ver}}) \leqslant \varepsilon_{\text{cor}}/(1 - p_\perp^{\text{ver}}).\end{aligned}$$

Just these calculations are presented in the proof of the protocol correctness in Ref. [27], where the criterion for the correctness of the protocol is formulated as

$$\Pr[K_A \neq K_B \wedge F(K_A) = F(K_B)] \leqslant \varepsilon_{\text{cor}}, \tag{17}$$

i.e., also as the joint probability that the test (in this case, verification) is passed, but the keys are not as expected (in this case, they do not coincide).

Estimation of the joint probability of type (17), i.e., the joint probability that the protocol will not be aborted, but the key will be unreliable, is also present in the authentication of communication using the classical channel, which is taken into account in parameter ε. We will consider this subject in Section 4.1.

3.5. Security in the case when a part of the key leaks to the adversary

A series of papers [19–22] argue that a key that is secure in the sense of estimate (4) may in fact be unsecure when part of it becomes known to the adversary. For example, this may occur in the case of an attack based on a partially known plaintext. Let the plaintext be encrypted with a one-time pad with an ε-secure key and part of the plaintext is known to the adversary. Then he becomes aware of the corresponding part of the key. In fact, this issue has already been considered in Section 3.3: the set $\mathcal{M}$ of plaintexts known to the adversary reflects the adversary's partial knowledge of plaintext. For example, if the adversary knows the first word of the plaintext, then $\mathcal{M}$ contains only plaintexts starting with this word. Proposition 2 and formula (14) indicate that it is very likely that a reduction in the number of plaintexts because of the adversary reading a ciphertext with an ε-secure key in a percentage ratio is small. But with some low probability, a significant reduction of this set, up to a single element, i.e., a complete decryption of the message, is also possible.

The following explicit example illustrating this situation is presented in Ref. [23]. Consider the state ρ_{KE} (1), in which for simplicity we set $p_\perp = 0$ and $\rho_E^k = \rho_E$, i.e., the adversary subsystem does not correlate with the key, but the key is distributed non-uniformly. This can happen due to both adversary attacks and imperfect hardware. Namely, let $p_k = 2^{-L}$ for all $k = 0\ldots0 \equiv \mathbf{0}$ and $k = 1\ldots1 \equiv \mathbf{1}$, the probabilities of which are $p_{\mathbf{0}} = 0$ and $p_{\mathbf{1}} = 2^{-L+1}$. Such a key is ε-secure for $\varepsilon \geqslant 2^{-L}$. As we mentioned above, with realistic values of L and ε, this inequality holds.

Suppose that the adversary becomes aware that all bits of the key except the last one are zeros. Then, since $p_{\mathbf{0}} = 0$, the adversary learns that the last bit of the key is one. This allows him to decrypt the last bit of the private message unknown to him (in the example with encryption with a one-time pad and partially known plain text). According to researchers who criticise the use of trace distance as a measure of security, this violates Shannon's definition of cryptosystem security, which postulates that whatever the adversary's prior knowledge, he should learn nothing more.

At the same time, it is worth noting that the probability that all bits of the key, except the last one, are equal to zero is very small and amounts to 2^{-L}. In other words, with a probability of $1 - 2^{-L}$ there will be ones among these bits, and then the knowledge of these bits by the adversary will not allow him to determine with certainty the last bit of the key. Moreover, with a probability of $1 - 2^{-L+2}$, there will be zeros and ones among these bits; in this case, there will be complete uncertainty regarding the last bit. This example once again illustrates formula (14) and the fact that the security criterion based on the trace distance guarantees security only in a probabilistic sense.

We now proceed from an example to a proof of the general fact: when part of an ε-secure key leaks to an adversary, the remaining part of the key is still ε-secure. This fact was proved in Ref. [5] in terms of the general theory of universally composable security, but we will prove this by using direct transformations of quantum states. Let the adversary know the last $1 \leqslant L_2 < L$ bits of the key ($L_1 = L - L_2$). The key $k \in \{0, 1\}^L$ can be written as $k = k_1 k_2$, $k_i \in \{0, 1\}^{Li}$, $i = 1, 2$. Then the state ρ_{KE} (1) can be rewritten in the form:

$$\begin{aligned}\rho_{KE} &= p_\perp |\perp\rangle\langle\perp| \otimes \rho_E^\perp + (1 - p_\perp) \\ &\times \sum_{k_1 \in \{0,1\}^{L_1}} \sum_{k_2 \in \{0,1\}^{L_2}} p_{k_1 k_2} |k_1 k_2\rangle\langle k_1 k_2| \otimes \rho_E^{k_1 k_2} \\ &= p_\perp |\perp\rangle\langle\perp| \otimes \rho_E^\perp + (1 - p_\perp) \\ &\times \sum_{k_1 \in \{0,1\}^{L_1}} p_{k_1} |k_1\rangle\langle k_1| \otimes \sum_{k_2 \in \{0,1\}^{L_2}} \frac{p_{k_1 k_2}}{p_{k_1}} |k_2\rangle\langle k_2| \otimes \rho_E^{k_1 k_2} \\ &\equiv p_\perp |\perp\rangle\langle\perp| \otimes \rho_E^\perp + (1 - p_\perp) \sum_{k_1 \in \{0,1\}^{L_1}} p_{k_1} |k_1\rangle\langle k_1| \otimes \tilde{\rho}_E^{k_1},\end{aligned} \tag{18}$$

where

$$p_{k_1} = \sum_{k_2 \in \{0,1\}^{L_2}} p_{k_1 k_2}.$$

Thus we rewrote the k_2 subsystem as part of the adversary subsystem, which corresponds to a leakage of a part of the key and considering only the rest of the key left for legitimate participants. For ideal state (2) we have $p_{k_1 k_2} = 2^{-L}$, $p_{k_1} = 2^{-L_1}$, and

$$\begin{aligned}\tilde{\rho}_E^{k_1} &= \sum_{k_2 \in \{0,1\}^{L_2}} \frac{p_{k_1 k_2}}{p_{k_1}} |k_2\rangle\langle k_2| \otimes \rho_E \\ &= \left(2^{-L_2} \sum_{k_2 \in \{0,1\}^{L_2}} |k_2\rangle\langle k_2|\right) \otimes \rho_E,\end{aligned}$$

i.e., it does not depend on k_1. Therefore, the ideal state for the transfer of the subsystem k_2 to the adversary also turns into ideal state for a key of length L_1. Since in fact, we are dealing with the same states ρ_{KE} and ρ_{KE}^{ideal}, simply written in a differ-

ent way, then the trace distance between them remains the same. Therefore, for leakage of a key part to the adversary the state corresponding to the rest of the key is at a distance ε from the appropriate ideal state. It means that the remaining part of the key is ε-secure and all interpretations mentioned above are applicable to it. We will return to discussing these interpretations in Section 3.6.

3.6. Other interpretations of the security parameter

We gave interpretations of the security parameter, having the most practical operational sense. For completeness, based on Ref. [5] we consider two other commonly referred interpretations that allow better understanding of the meaning of this parameter.

The first one is an estimate of the probability of guessing the key by the adversary. The adversary wants to guess the key by measuring his subsystem. Let the measurement be given by a probabilistic operator-valued measure $\{M_k\}_{k\in\{0,1\}^L}$, $M_k \geqslant 0$, $\sum_k M_k = I_E$, all operators act in the space $\mathscr{H}_E$. The adversary interprets the outcome k as that the legitimate parties generated key k. The guessing probability is defined as

$$P_{\text{guess}} = \sum_{k\in\{0,1\}^L} \text{Tr}[\rho_{KE}(|k\rangle\langle k|\otimes M_k)] = \sum_{k\in\{0,1\}^L} p_k \text{Tr}(\rho_E^k M_k).$$

For the ideal state (2), obviously, $P_{\text{guess}} = 2^{-L}$, i.e., the adversary's subsystem does not correlate with a key that is evenly distributed, so the adversary can only merely guess the value of a uniformly distributed random variable. For the real state (1), we have

$$P_{\text{guess}} \leqslant 2^{-L} + D(\rho_{KE}^{\top}, \rho_{KE}^{\top,\text{ideal}}). \tag{19}$$

This estimate can be obtained from formula (10) for the probability of distinguishing between the states ρ_{KE} and ρ_{KE}^{ideal}, the distinguishing dimension is $\{\tilde{M}, I-\tilde{M}\}$, where the operator

$$\tilde{M} = \sum_{k\in\{0,1\}^L} |k\rangle\langle k|\otimes M_k$$

corresponds to the correct guessing, and $I-\tilde{M}$ corresponds to the error. If the key is guessed correctly, then the ddistinguisher interprets this result as input of the real state ρ_{KE} (since this probability is higher for it); in the case of an error, it reacts as if that the ideal state ρ_{KE}^{ideal} was input. The estimate (19) can also be considered a particular case of Proposition 1, since substituting the most probable key is one of the methods of attacking the application protocol. The estimate (19) suggests that the use of an ε-secure key increases this probability by no more than $D(\rho_{KE}^{\top}, \rho_{KE}^{\top,\text{ideal}})$.

The second interpretation is related to the probability of coincidence with a hypothetical ideal key. Namely, let us consider the measurement of an arbitrary observable, which is carried out simultaneously with the real state of ρ_{KE} and the ideal state of ρ_{KE}^{ideal}. The measurement result is a random variable, so we get two random variables, Z and Z^{ideal}, with probability distributions $P_Z(z)$ and $P_{Z^{\text{ideal}}}(z)$, where z runs a finite set of measurement outcomes. Then there exists a joint distribution $P_{Z,Z^{\text{ideal}}}(z,z')$ for which $P_Z(z)$ and $P_{Z^{\text{ideal}}}(z')$ are partial distributions and for which the following inequality holds:

$$\Pr[Z \neq Z^{\text{ideal}}] \equiv \sum_z P_{Z,Z^{\text{ideal}}}(z,z) \leqslant \varepsilon, \tag{20}$$

if the trace distance between ρ_{KE} and ρ_{KE}^{ideal} does not exceed ε. We recall that Pr denotes the probability of an event.

The estimate (20) is sometimes interpreted as the fact that ε is the upper estimate of the probability of the difference between the measurement result of any observable in the state ρ_{KE} and the measurement result in the ρ_{KE}^{ideal} state, i.e., the probability that the QKD protocol will manifest itself in a different way than hypothetical ideal key distribution protocol. In other words, the QKD protocol behaves similar to an ideal one, up to a low probability.

However, this interpretation is criticised [19–22] because neither the ideal protocol nor the ideal state ρ_{KE}^{ideal} nor, therefore, the random variable Z^{ideal} and the joint distribution $P_{Z,Z^{\text{ideal}}}(z,z')$ actually exist, so that inequality (20) does not correspond to any real fact. For example, it cannot be verified experimentally. Even if we construct a model of an ideal key distribution protocol (for example, by connecting legitimate parties with an additional channel, completely protected from the adversary's eavesdropping), then with independently working protocols $P_{Z,Z^{\text{ideal}}}(z,z') = P_Z(z)P_{Z^{\text{ideal}}}(z')$, so that inequality (20) is not satisfied. It is also agreed in Ref. [5] that estimate (20) serves only for better intuitive understanding of the trace distance and the choice of the value of ε. In our opinion, the interpretations given in Subsections 3.1 and 3.3 reflect the meaning of $\varepsilon/(1-p_\perp)$ in a more practical way as the probability of breaking.

4. Practical aspects of using the security parameter

4.1. Components of the security parameter

Typically, the security parameter is composed of the following components [3, 4, 26]:

$$\varepsilon = \varepsilon_{\text{pa}} + \varepsilon_{\text{pe}} + \varepsilon_{\text{auth}}, \tag{21}$$

where ε_{pa} is the parameter of privacy amplification (pa); ε_{pe} is the probability that one of the statistical estimates performed during the protocol will be incorrect (pe – parameter estimation); $\varepsilon_{\text{auth}}$ is the parameter of authentication of communication via the classic channel. We briefly explain each term.

Privacy amplification implies application of a hash function to a corrected key that maps it into a key of smaller length. The adversary's information about the key is reduced, as a result, not to zero, but to the given parameter epa, which is an estimate of the trace distance. The stronger the key is compressed, i.e., the greater the difference between the lengths of the corrected and final keys, the smaller it is. Conceptually, this parameter is the main one in Eqn (21). However, to ensure the security of the key, two more conditions must be met.

First, to calculate how much you need to compress the key to ensure a given ε_{pa}, you need to evaluate the adversary's information about the sifted key. For this purpose, interval statistical estimates of certain parameters are used, according to which the degree of adversary intervention is revealed. In

the simplest case, this is only the noise level (QBER – quantum bit error rate), but frequently the QKD protocols include estimates of other parameters, too. Any interval statistical estimate has a confidence probability. Let the statistical estimation of m parameters $X_1, \ldots, X_m$ be performed in the QKD protocol; all estimates have the form

$$A_j \leqslant X_j \leqslant B_j, \tag{22}$$

where one of the boundaries, A_j or B_j, can be infinite, i.e., the estimate can be one-sided (which most often takes place). The boundaries A_j and B_j, as a rule, depend on the observed data (for example, empirical mean values), i.e. formally they are also random variables. Let $1 - \varepsilon_{\text{pe}}^{(j)}$, $j = 1, \ldots, m$ be the confidence probabilities of the estimates, i.e.,

$$\Pr[\text{observed data} \,|\, X_j \notin [A_j, B_j]] \leqslant \varepsilon_{\text{pe}}^{(j)},$$

and, therefore,

$$\Pr[\text{observed data} \wedge X_j \notin [A_j, B_j]] \leqslant \varepsilon_{\text{pe}}^{(j)}$$

is also true. Then

$$\varepsilon_{\text{pe}} = \sum_{j=1}^{m} \varepsilon_{\text{pe}}^{(j)}, \tag{23}$$

i.e., ε_{pe} is the upper estimate of the probability that at least one interval estimate (22) is not valid.

We note separately that the method of decoy states also includes interval statistical estimates, the confidence probabilities of which, therefore, should be taken into account in (23) (see, e.g., [28]).

Second, to ensure security, it is necessary to authenticate communication using the classical channel. The parameter $\varepsilon_{\text{auth}}$ reflects the degree of authentication security. Theoretically secure message authentication codes use hash functions and assume that the legitimate parties have a short initial private key (possibly from a previous session) before starting a QKD session. Then $\varepsilon_{\text{auth}}$ is the probability of collision of the hash function, i.e., the upper bound for the probability of an expression of type (16). Similarly, it can be reduced to a joint probability of type (17), i.e., to the joint probability of line mismatch and hash function coincidence.

Note that in formula (21) we assume that a perfectly secure key is taken for authentication. If an ε_0-secure key is taken (for example, from one of the previous QKD sessions), then this term should be added to the right-hand side of (21). We do not add it, since we have already discussed this problem in Section 3.4.

Let us briefly explain why ε can be represented as the sum of the conceptually main term ε_{pa} and the probability $q \leqslant \varepsilon_{\text{pe}} + \varepsilon_{\text{auth}}$ that one of the necessary conditions for ensuring this ε_{pa} is not fulfilled. Let ρ_{KE} be represented as

$$\rho_{KE} = (1 - q)\rho_{KE}^{\text{good}} + q\rho_{KE}^{\text{bad}},$$

where $D(\rho_{KE}^{\text{good}}, \rho_{KE}^{\text{ideal}}) \leqslant \varepsilon_{\text{pa}}$; and the operators ρ_{KE}^{good} and ρ_{KE}^{bad} are positive and have a unit trace, that is, they are states. Then from the property of strict convexity of the trace distance [29]

$$D(\rho_{KE}, \rho_{KE}^{\text{ideal}}) \leqslant \varepsilon_{\text{pa}} + q \leqslant \varepsilon.$$

As for the correctness parameter (17) (the probability of key mismatch), which we discussed in Section 3.4, in the security analysis it is possible to consider the total parameter $\varepsilon_{\text{tot}} = \varepsilon + \varepsilon_{\text{cor}}$ (in this case the parameter ε usually has a subscript sec, i.e. secrecy). Then $p_\perp$ is understood as the probability of protocol abortion due to any reason: failure to verify, detection of excessive intervention in the quantum channel or intervention in classical communication. However, these conditions – correctness (17) and secrecy (4) – can be considered separately. It is believed that the problem of key mismatch is less acute than the key privacy problem: in many applications erroneous decryption is detected, therefore, when a low probability of key mismatch is triggered, it is sufficient to resend the data (encrypted with a new key, completely independent of the previous one) [2]. Then ε_{cor} can be set to a higher value than ε_{sec}.

4.2. Variable key length

Until now, the key length distributed in the QKD protocol was considered to be predetermined. Let us now consider the implementation of the protocol when the key length is not specified in advance, but is calculated by legitimate parties during the protocol execution. In this case, as a result of the QKD protocol and the adversary attack, a state arises having the form

$$\rho_{KE} = \sum_{l=0}^{L_{\max}} p^{(l)} \rho_{KE}^{(l)},$$

$$\rho_{KE}^{(l)} = \sum_{k \in \{0,1\}^l} p_k \, |k\rangle_K \langle k| \otimes \rho_E^k,$$

where $\sum_{l=0}^{L_{\max}} p^{(l)} = 1$; $\sum_{k \in \{0,1\}^l} p_k = 1$ for all l; $p^{(l)} \geqslant 0$ is the probability of generating a key of length l. Register K in this case has a variable length from zero (refusal to generate a key) to $L_{\max}$ (maximum possible key length), and therefore the corresponding Hilbert space is $\bigoplus_{l=0}^{L_{\max}} \mathbb{C}^l$. The corresponding ideal state is

$$\rho_{KE}^{\text{ideal}} = \sum_{l=0}^{L_{\max}} p^{(l)} \rho_{KE}^{(l),\text{ideal}},$$

$$\rho_{KE}^{(l),\text{ideal}} = 2^{-l} \sum_{k \in \{0,1\}^l} |k\rangle_K \langle k| \otimes \rho_E^{(l)},$$

where $\rho_E^{(l)}$ are defined by formula (3) with $\top$ replaced with (l).

We again determine the ε-security based on inequality (4). Note that due to the orthogonality of $\rho_{KE}^{(l)}$ for different l (similarly for $\rho_{KE}^{(l),\text{ideal}}$) we obtain

$$D(\rho_{KE}, \rho_{KE}^{\text{ideal}}) = \sum_{l=1}^{L_{\max}} p^{(l)} D(\rho_{KE}^{(l)}, \rho_{KE}^{(l),\text{ideal}}).$$

Suppose that inequality (4) holds and the legitimate parties have generated a key of some length $1 \leqslant L \leqslant L_{\max}$. Then they are only interested in the distance $D(\rho_{KE}^{(L)}, \rho_{KE}^{(L),\text{ideal}})$ and the factor before it. To derive the corresponding inequality, we use the fact that $D(\rho_{KE}^{(l)}, \rho_{KE}^{(l),\text{ideal}})$ increases with growing l. In fact, the stronger the key compression at the stage of privacy

amplification, the less information about the final key remains available to the adversary. Then

$$\varepsilon \geqslant \sum_{l=1}^{L_{\max}} p^{(l)} D(\rho_{KE}^{(l)}, \rho_{KE}^{(l),\text{ideal}}) \geqslant \sum_{l=L}^{L_{\max}} p^{(l)} D(\rho_{KE}^{(l)}, \rho_{KE}^{(l),\text{ideal}})$$

$$\geqslant \left(\sum_{l=L}^{L_{\max}} p^{(l)} \right) D(\rho_{KE}^{(L)}, \rho_{KE}^{(L),\text{ideal}}).$$

If we denote the factor in brackets in the last expression by $1 - p_\perp$, then we again obtain Eqn (8). As in the case of a fixed length, it is the probability that the legitimate parties conclude that it is possible to generate a key of length L (if it is possible to generate a longer key, then all the more it possible to generate a key of length L).

Again we can conclude that the value of $D(\rho_{KE}^{(L)}, \rho_{KE}^{(L),\text{ideal}})$, generally speaking, is not small. It can be great if the probability $1 - p_\perp$ is low.

Note that the weaker inequality

$$p^{(L)} D(\rho_{KE}^{(L)}, \rho_{KE}^{(L),\text{ideal}}) \leqslant \varepsilon$$

is insufficient, because if the probability distribution $\{p^{(l)}\}$ has a large variance (which is possible even in the absence of eavesdropping), then all the probabilities $p^{(l)}$ can be sufficiently small. In other words, if l takes a large number of values, then any value of this random variable will be a low-probability event; therefore, it cannot be concluded that $D(\rho_{KE}^{(L)}, \rho_{KE}^{(L),\text{ideal}})$ is small.

Thus, the case of a variable key length reduces to inequality (8); therefore, considering the case of a fixed key length in the rest of the article of does not limit generality.

4.3. Partitioning and joining keys

A cipher usually requires a key of fixed length, which may not coincide with the length of the key distributed using the QKD protocol, especially if the key length in the latter is variable. In this regard, it may be necessary to split the key distributed in the QKD protocol into parts, or, conversely, combine the keys distributed in different QKD sessions into one. These operations can be required together. Suppose that a cipher requires a 256-bit key. Then, we decompose the key distributed using the QKD into blocks with a length of 256 bits. But if the length of the distributed key is not a multiple of 256 bits, then the last block will have a shorter length, and therefore it is necessary to combine it with part of the key distributed in another session. Therefore, it is necessary to specify the formulas for the key security parameter obtained by these operations.

The trace distance $D(\rho, \sigma)$ does not increase when applying an arbitrary quantum transformation, i.e., a completely positive and trace-preserving map Φ, to both arguments: $D(\Phi(\rho), \Phi(\sigma)) \leqslant (\rho, \sigma)$. Omitting part of a key is a quantum transformation. We define it formally. Let the last $1 \leqslant L_2 \leqslant L$ bits of the key be discarded, $L_1 = L - L_2$. As in the analysis of key security in the case leaking a part of the key to the adversary, we write the key $k \in \{0,1\}^L$ in the form $k = k_1 k_2$, $k_i \in \{0,1\}^{L_i}$, $i = 1, 2$. We have

$$\rho_{KE} \mapsto |\perp\rangle\langle\perp| \rho_{KE} |\perp\rangle\langle\perp| \sum_{k_1 \in \{0,1\}^{L_1}} V_{k_1} \rho_{KE} V_{k_1}^\dagger =$$

$$= p_\perp |\perp\rangle\langle\perp| \otimes \rho_E^\perp + (1 - p_\perp) \sum_{k_1 \in \{0,1\}^{L_1}} p_{k_1} |k_1\rangle\langle k_1|$$

$$\otimes \sum_{k_2 \in \{0,1\}^{L_2}} \frac{p_{k_1 k_2}}{p_{k_1}} \rho_E^{k_1 k_2}, \qquad (24)$$

where

$$p_{k_1} = \sum_{k_2 \in \{0,1\}^{L_2}} p_{k_1 k_2}; V_{k_1} = \sum_{k_2 \in \{0,1\}^{L_2}} |k_1\rangle\langle k_1 k_2| \otimes I_E : \mathbb{C}^{2^L} \otimes \mathscr{H}_E \to$$

$$\mathbb{C}^{2^{L_1}} \otimes \mathscr{H}_E.$$

The difference between transformations (24) and transformations (18) when part of the key is leaked to the adversary is that in this case, the subsystem k_2 is discarded rather than included in the adversary subsystem. The rest of the formulas are similar. The ideal state during transformation (24) also transits into the ideal state (in a new space): $p_{k_1 k_2} = 2^{-L}$, $p_{k_1} = 2^{-L_1}$. Therefore, we can conclude that the key obtained from the ε-secure key by discarding part of it, is also ε-secure.

When dividing a key into several shorter keys, each 'short' key is obtained by discarding part of the positions from the original sequence of bits. Therefore, if the source key is ε-secure, then each of the small keys will also be ε-secure. However, when calculating complex probabilities in Proposition 1 and formula (14), it should be kept in mind that *the securities of these keys are not independent. For example, if during the distribution of the original 'long' key a hardly probable event of non-detection of eavesdropping occurred, then all short keys are unsecure at the same time*. Therefore, Proposition 1 and formula (14) must be applied to the long key and not to each short key individually. Then the quantity p in Proposition 1 can be understood as the probability of reading one of the messages encrypted with one of the short keys. To apply formula (14) when using short keys in a one-time pad mode, you can combine the used short one-time pads into a long one. Many possible plaintexts $\mathscr{M}$ can then be obtained as a result of all kinds of concatenations of short plaintexts.

When merging the keys their security parameters add: if two keys had the security parameters ε_1 and ε_2, then the security parameter of the key obtained as a result of their joining is equal to $\varepsilon_1 + \varepsilon_2$. This stems from the inequality [29]

$$D(\rho_1 \otimes \rho_2, \sigma_1 \otimes \sigma_2) \leqslant D(\rho_1, \sigma_1) + D(\rho_2, \sigma_2).$$

Using these two rules allows calculating the security of the key for an arbitrary combination of partitioning and joining of different keys.

4.4. Examples of calculating the time before the first breaking

In order to sort out roughly what value of ε should be chosen in practice, instead of probabilities, it may be convenient to consider the inverse quantities: the average number of sessions of the QKD at which one case of breaking will occur. Multiplying this value by the average execution time of the QKD protocol yields the same quantity expressed in time units.

For the calculations, we will use the following conditions: let the QKD hardware generate a key of length $L = 10^6$ with a

security parameter $\varepsilon = 10^{-11}$ every ten seconds. Suppose first that the key is used in a one-time pad mode (in Ref. [4] it is emphasised that a rate of 100 kbit s^{-1} is sufficient for encrypting audio messages in a one-time pad mode). We use the formula (14) and evaluate the joint probability that the protocol will not be aborted, but the nonideal key will allow the adversary to reduce the number of possible open texts by 1%, i.e. $\delta = 1/100$. The probability of breaking a protocol for the given δ will be $2\varepsilon/\delta = 2\times10^{-9}$, i.e., the breaking will occur on average in one session out of 500 million, or once every 150 years.

Suppose now that the key is used in a computationally secure cipher with a key length of 256 bits, i.e., a key of 10^6 bits is split into short keys, 256 bits long, each new message being encrypted with a new key. We also assume that the key is expended at the same rate as it is generated, that is, $10^5/256 \approx 390$ messages are encrypted per second. Suppose that in a situation where each message is encrypted with a new key, the cipher cannot be broken otherwise than by brute force. This assumption is justified by the fact that usually attacks on popular symmetric cryptosystems require knowledge of a large amount of data encrypted with the same key. Therefore, the encryption of each message with a new key, completely independent of the previous ones, leaves the possibility of breaking only by brute force. Even if we assume that the adversary has the ability to enumerate $2^{128} \approx 10^{38}$ keys, which is an extremely high number, the probability of a successful breaking (with an ideal key) is $p = 10^{-38} \ll \varepsilon$. Therefore, using Proposition 1, $\varepsilon = 10^{-11}$ can be approximately considered as the probability of breaking the cipher (here we mean the joint probability of generating a key and breaking a cipher with this key). Moreover, for breaking the adversary does not need to enumerate 2^{128} keys for each encrypted message. He can try, for example, only one most probable key. With a perfectly secure key, the probability of success will be $2^{-256} \approx 10^{-77}$; the reciprocal of this number is comparable to the number of atoms in the visible part of the Universe. However, when using an ε-secure key, it is only guaranteed that this probability does not exceed $2^{-256} + \varepsilon \approx \varepsilon$. As we understood from Section 4.3, the probability calculation must be performed for a long (10^6 bit) key. The probability of its breaking is $\varepsilon = 10^{-11}$, that is, breaking will occur on average once every 30 thousand years.

From these calculations it follows that a one-time pad requires not only a higher key generation rate than computationally secure ciphers, but also a significantly lower ε value, especially if we want to reduce the fraction δ of plaintext excluded by the adversary due to the key nonideality. Indeed, according to the formula (14), the probability of breaking the protocol is $2\varepsilon/\delta$ rather than ε (as in Proposition 1 with $p \ll \varepsilon$).

5. Conclusions

When using a key distributed with the help of QKD in a computationally secure symmetric cipher (Magma, Kuznyechik, AES, etc.), the most natural interpretation of the key security parameter ε is given by Proposition 1, which is associated with an increase in the probability of breaking the cipher due to replacing an ideal key with an ε-secure one, as well as formula (11) that relates this parameter to the amount of computation that guarantees a given probability of breaking. If the probability of breaking the cipher in realistic attacks is negligible, then ε can be interpreted as the probability of breaking the QKD protocol. Formula (11) relates the security parameter with the difficulty of breaking a cipher, which, when using an ε-secure key by the legitimate parties, can be estimated from below in terms of breaking difficulty with an ideal key, but providing a lower by ε/α probability of breaking, where α is the lower estimate of the key generation probability.

When using a distributed key in a one-time pad mode, it is proposed to use Proposition 2 and formula (14) for the main operational meaning, which show how the set of possible (from the adversary's point of view) plaintexts can be reduced as a result of using an ε-secure key instead of an ideal one. A significant reduction of this set (up to a single element, i.e., a complete decryption of the message) is possible, but it has a low probability of the order of ε. Therefore, even in this case, ε can be considered, in order of magnitude, as the probability of breaking the QKD protocol. At the same time, we saw that a one-time pad requires not only a higher key generation rate than computationally secure ciphers, but also a significantly lower ε value.

Acknowledgements. The author thanks D.A. Kronberg for useful discussions and comments.

References

1. Gisin N., Ribordy G., Tittel W., Zbinden H. *Rev. Mod. Phys.*, **74**, 145 (2002).
2. Tomamichel M., Lim C.C.W., Gisin N., Renner R. *Nat. Commun.*, **3**, 634 (2012).
3. Kiktenko E.O., Pozhar N.O., Duplinsky A.V., Kanapin A.A., Sokolov A.S., Vorobey S.S., Miller A.V., Ustimchik V.E., Anufriev M.N., Trushechkin A.S., Yunusov R.R., Kurochkin V.L., Kurochkin Yu.V., Fedorov A.K. *Quantum Electron.*, **47**, 798 (2017) [*Kvantovaya Elektron.*, **47**, 798 (2017)].
4. Duplinskiy A.V., Kiktenko E.O., Pozhar N.O., Anufriev M.N., Ermakov R.P., Kotov A.I., Brodskiy A.V., Yunusov R.R., Kurochkin V.L., Fedorov A.K., Kurochkin Y.V. *J. Russ. Laser Res.*, **39**, 113 (2018).
5. Portmann C., Renner R. arXiv:1409.3525 (2014).
6. Pfitzmann B., Waidner M. *Proc. 7th ACM Conf. on Computer and Communications Security* (ACM, 2000) pp 245–254.
7. Pfitzmann B.,Waidner M. *IEEE Symposium on Security and Privacy* (IEEE, 2001) pp 184–200.
8. Backes M., Pfitzmann B., Waidner M. *Lecture Notes in Computer Science* (Springer, 2004) Vol. 2951, pp 336–354.
9. Backes M., Pfitzmann B., Waidner M. *Inform. Comput.*, **205**, 1685 (2007).
10. Canetti R. *Proc. 42nd Symp. Foundations of Computer Science, FOCS'01* (IEEE, 2001) pp 136–145; IACR e-print: 2000/067.
11. Maurer U., Renner R. *Proc. Innovations in Computer Science, ICS 2010* (Tsinghua: University Press, 2011) pp 1–21.
12. Ben-Or M., Mayers D. arXiv:quant-ph/0409062 (2004).
13. Ben-Or M., Horodecki M., Leung D., Mayers D., Oppenheim J. *Lecture Notes in Computer Science* (Springer, 2005) Vol. 3378, pp 386–406; arXiv:quant-ph/0409078.
14. Unruh D. arXiv:quant-ph/0409125 (2004).
15. Unruh D. *Lecture Notes in Computer Science* (Springer, 2010) Vol. 6110, pp 486–505; arXiv:0910.2912.
16. Renner R., König R. *Lecture Notes in Computer Science* (Springer, 2005) Vol. 3378, pp 407–425; arXiv:quant-ph/0403133.
17. Müller-Quade J., Renner R. *New J. Phys.*, **11**, 085006 (2009).
18. König R., Renner R., Bariska A., Maurer U. *Phys. Rev. Lett.*, **98**, 140502 (2007).

19. Yuen H. arXiv:0907.4694 (2009).
20. Yuen H. *Phys. Rev. A*, **82**, 062304 (2010).
21. Hirota O. arXiv:1208.2106 (2012).
22. Hirota O. arXiv:1306.1277 (2013).
23. Renner R. arXiv:1209.2423 (2012).
24. Arbekov I.M., Molotkov S.N. *JETP*, **125** (1), 50 (2017) [*Zh. Eksp. Teor. Fiz*., **152** (1), 62 (2017)].
25. Shannon K., in *Raboty po teorii informatsii i kibernetike* (Papers on the Theory of Information and Cybernetics) (Moscow: IIL, 1963) pp. 333–402.
26. Fung C.-H., Ma X., Chau H.F. *Phys. Rev. A*, **81**, 012318 (2010).
27. Tomamichel M., Leverrier A. *Quantum*, **1**, 14 (2017).
28. Trushechkin A.S., Kiktenko E.O., Fedorov A.K. *Phys. Rev. A*, **96**, 022316 (2017).
29. Wilde M.V. *Quantum Information Theory* (Cambridge University Press, 2017).