Key Sharing Protocol Using Exchange by Integers over Public Noiseless Channels Between Users that Provides Security without Cryptographic Assumptions

Viktor Yakovlev, Valery Korzhik
The Bonch-Bruevich Saint-Petersburg State University of Telecommunication
Saint-Petersburg, Russia
viyakov4@gmail.com, korzhikvalery11@gmail.com

Milena Akhmetshina, Aleksei Zhuvikin
The Bonch-Bruevich Saint-Petersburg State University of Telecommunication
Saint-Petersburg, Russia
ame16.mil@gmail.com, mail@zhuvikin.com

***Abstract*—Recently, we proposed similar key sharing protocol executed over the public noiseless channels and also without any cryptographic assumptions. Exchange by matrices was believed to have superior performance in that protocol. In the current paper, the matrix exchange was replaced by integer exchange over the same channels. It requires to apply additional subprotocols to make both legitimate and eavesdropper channels worse. A replacing matrices by integers allowed us to derive some probability relations theoretically instead of relying on the simulation results only in our previous paper. Moreover, the use of the integers in place of matrices requires less traffic over channels between legitimate correspondents. The performance evaluation of the proposed protocol in terms of the reliability and security is presented as well.**

I. Introduction

The secret key sharing problem between ordinary individuals involved into Internet activity is still not solved completely. The use of public key cryptography (in particularity Diffie-Hellman protocol [1]) is based on some cryptographic assumptions. Those are, first of all, discrete log and factorization problems. Unfortunately, it was proved by Shor [2] that these problems can be solved by so called quantum computer. Although a practical implementation of quantum computer is still highly conjectural, nevertheless such a threat exists in the future and generated a new trend in cryptography, so called *post-quantum* cryptosystems. For example, McEliece [3] is the one of them. However, post-quantum cryptosystems are too complex for ordinary users.

Another trend of key sharing is to execute so called quantum cryptography [4], but still it is not the way for the Internet community because it requires special devices in quantum communication channels.

A new approach for the key sharing problem is based on the notion of *physical layer security* [5]. This way exploits physical properties of the real communication channels connecting legitimate users sharing a secret key in a presence of eavesdropper. But as a rule, all such methods require some restrictions on eavesdropper channels that cannot be provided in practice. So, it is necessary to provide a lower bound for power of noises in eavesdropper channel or the upper bound for the number of antennas if users apply MIMO technology [6].

In the paper [7] the first such key sharing protocol (KSP) that is not based on some *cryptographic assumptions* and does not require some requirements to eavesdropper channels executing over ordinary Internet has been proposed. This KSP executes exchanging with use of the large size matrices over noiseless channel with feedback. The proof of reliability and security was based on the simulation results because theoretical derivations occur intractable for such matrices. Moreover, such KSP generates large volume of channel traffic.

In the current paper, we propose to replace matrix channel exchanging with ordinary integers (numbers) over the Internet channels which allowed us, firstly, to prove theoretical bounds on reliability and security and, secondly, to reduce the traffic over the channels.

It is worth to note that in paper mentioned above [7] the use of integer exchange (see Table I there) was rejected as an inefficient approach. But in the current paper we overcome such a defect by means of the additional iterative protocols.

In section II a new KSIP with exchange by integers over the public noiseless channels is presented and formulas for the probabilities of different joint events which will be used in subsequent sections were proved. Section III describes PIMC protocol given in [7] and presents additional iterative IPIMC and subprotocol DBC called by a *degradation both legitimate and eavesdropping channels* with a derivation of some probability characteristic. In Section IV the reliability and security of the proposed KSP are proved. Section V summarizes main results and proposes some actual direction for the future investigations.

II. Description of new KSIP for integers and the proof of some probabilities required in the next sections

Let us denote such a protocol by abbreviation KSIP. The scenario of the protocol is presented in Fig.1. Before the

transmission over the channel both participating users Alice (A) and Bob (B) generate random numbers p and q, correspondingly as well as the random numbers related to artificial noises n_A and n_B respectively.

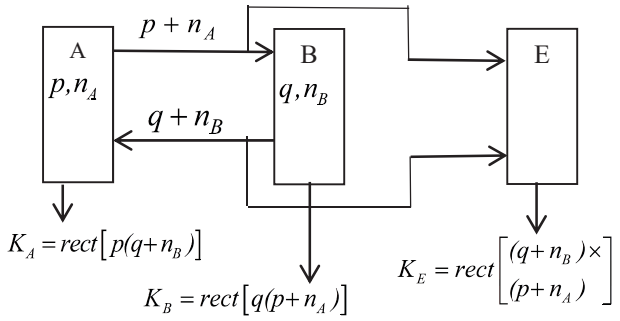

Fig.1. Scenario of the raw key bit formation protocol based on exchange by integers over public noiseless channels with feedback

Next, A transmits the sum of integers $p+n_A$ to B over the channe $A \to B$ land B transmits the sum of integers $q+n_B$ to A over the channel $B \to A$. The Channels $A \to B$ and $B \to A$ can jointly be called by the *main channel.* We believe that random numbers *p, q* belong to *Gaussian distribution* with parameters (0,1) and they are mutually independent as well as both random numbers n_A, n_B are distributed with parameters $(0, \sigma^2)$ and mutually independent too. Then, each of the users A and B calculates product of their integers p and q by corresponding sums $q+n_B$ and $p+n_A$ received over the channels, that gives the random values $p(q+n_B)$ and $q(p+n_A)$, respectively. Finally, both legitimate users calculate bits of the "raw" keys K_A and K_B as follows

$$K_A = rect\left[p(q+n_B)\right],$$

$$K_B = rect\left[q(p+n_A)\right],$$

$$\text{where } rect(x) = \begin{cases} 0, if\ x \ge 0, \\ 1, if\ x < 0 \end{cases}.$$

Eavesdropper E intercepts values $p+n_A, q+n_B$ over the channels $A \to B$, $B \to A$ and calculates their product. Eventually, E calculates her "raw" bits as

$$K_E = rect((p+n_A)(q+n_B)). \quad (1)$$

Derivation of the BERs for the main channel and for eavesdropper (wire-tap) channel.

First, due to symmetric distributions of the random numbers *p, q,* n_A and n_B we get the following relations:

$$\mathrm{P}(K_A = 0) = \mathrm{Pr}(K_A = 1) = 1/2,$$

$$\mathrm{P}(K_B = 0) = \mathrm{Pr}(K_B = 1) = 1/2,$$

$$\mathrm{P}(K_E = 0) = \mathrm{Pr}(K_E = 1) = 1/2.$$

Let us find the probabilities of coincidence for raw bits of A and B, that is $\mathrm{P}(K_A = K_B)$. The event of such coincidence corresponds to the following inequalities fulfillment

$p(q+n_B) > 0$, and $q(p+n_A) > 0$ if K_A=K_B=*0* or (2)

$p(q+n_B) < 0$, and $q(p+n_A) < 0$ if K_A=K_B=1. (3)

In the case (2) the fulfillment of the following equalities is sufficient,

$$p > 0, (q+n_B) > 0 \quad q > 0, (p+n_A) > 0, \quad (4)$$

$$p < 0, (q+n_B) < 0 \quad q < 0, (p+n_A) < 0, \quad (5)$$

$$p > 0, (q+n_B) > 0 \quad q < 0, (p+n_A) < 0, \quad (6)$$

$$p < 0, (q+n_B) < 0 \quad q > 0, (p+n_A) > 0 \quad (7)$$

The condition (4) can be transformed to the following one

$$p > 0, n_A > -p \quad or \quad q > 0, n_B > -q$$

Similar transformation can be obtained for the conditions (5)-(7), while the probabilities of the conditions (4)-(7) can be presented as follows

$$P1 = \int_0^{\infty} w(p) \int_{-p}^{\infty} w(n_A) dn_A dp \cdot \int_0^{\infty} w(q) \int_{-q}^{\infty} w(n_B) dn_B dq \cdot \quad (8)$$

$$P2 = \int_{-\infty}^{0} w(p) \int_{-\infty}^{-p} w(n_A) dn_A dp \cdot \int_{-\infty}^{0} w(q) \int_{-\infty}^{-q} w(n_B) dn_B dq \quad (9)$$

$$P3 = \int_0^{\infty} w(p) \int_{-\infty}^{-p} w(n_A) dn_A dp \cdot \int_{-\infty}^{0} w(q) \int_{-q}^{\infty} w(n_B) dn_B dq \quad (10)$$

$$P4 = \int_{-\infty}^{0} w(p) \int_{-p}^{\infty} w(n_A) dn_A dp \cdot \int_0^{\infty} w(q) \int_{-\infty}^{-q} w(n_B) dn_B dq \quad (11)$$

where $w(p), w(q), w(n_A), w(n_B)$ are the probability densities of the random values *p, q,* n_A, and n_B respectively.

We can write inequalities for the case when K_A=K_B=1 (see (3)) in a similar way to (8)-(11) and derive their probabilities denoted by *P5, P6, P7, P8.*

It easy to see that this group of inequalities has the same probability as the first one due to the symmetry of Gaussian distribution. Then we finally get

$$\mathrm{P}(K_A = K_B) = 2(P1 + P2 + P3 + P4)..$$

Disagreement between the key bits of A and B is equivalent to the following conditions

$p(q+n_B) > 0$, and $q(p+n_A) < 0$ or

$p(q+n_B) < 0$, and $q(p+n_A) > 0$.

Following to the same approach as for deriving of the probability $\mathrm{P}(K_A = K_B)$ and considering the symmetry of Gaussian distribution after the simple but tedious transforms, we get

$$\mathrm{P}(K_A \ne K_B) = 2(Q1 + Q2 + Q3 + Q4) \ , \quad (12)$$

where

$$Q(1)=\int_0^{\infty} w(p)\int_{-\infty}^{-p} w(n_A)dn_A dp\cdot\int_0^{\infty} w(q)\int_{-q}^{\infty} w(n_B)dn_B dq\cdot \quad (13)$$

$$Q(2)=\int_{-\infty}^{0} w(p)\int_{-p}^{\infty} w(n_A)dn_A dp\cdot\int_{-\infty}^{0} w(q)\int_{-\infty}^{-q} w(n_B)dn_B dq\cdot \quad (14)$$

$$Q(3)=\int_0^{\infty} w(p)\int_{-p}^{\infty} w(n_A)dn_A dp\cdot\int_{-\infty}^{0} w(q)\int_{-q}^{\infty} w(n_B)dn_B dq\cdot \quad (15)$$

$$Q(4)=\int_{-\infty}^{0} w(p)\int_{-\infty}^{-p} w(n_A)dn_A dp\cdot\int_{-\infty}^{0} w(q)\int_{-\infty}^{-q} w(n_B)dn_B dq\cdot \quad (16)$$

It is obviously that $\mathrm{P}(K_A \neq K_B)=1-\mathrm{P}(K_A = K_B)$ but it is required the derivation of relations (13)-(16) in order to prove the probabilities of different joint-events later.

In Table I all variants of possible events with corresponding to them probabilities *(Pi,Qi)* are presented.

Let us find the probability of the key bit coincidence between A and E.

$K_A = K_E$, if $rect[p(q+n_B)] = rect[(p+n_A)(q+n_B)]$,

These events occur if and only if the following inequalities hold:

$$p>0,\ \ q+n_B>0\ \ \ p+n_A>0,$$

$$p<0,\ \ q+n_B<0\ \ \ p+n_A<0,$$

$$p>0,\ \ q+n_B<0\ \ \ p+n_A>0,$$

$$p<0,\ \ q+n_B>0\ \ \ p+n_A<0.$$

They have the probabilities presented in Table I and denoted by *P1, P2, P7, P8, Q3, Q4, Q5, Q6.* Thus, we can write

$$P(K_E = K_A)=P1+P2+P7+P8+Q3+Q4+Q5+Q6.$$

The probability of the key bit disagreement between E and A can be expressed as follows

$$P(K_E \neq K_A)=1-P(K_E = K_A). \quad (17)$$

It is worth to note that all p**o**ssible combinations of the random events with values *p, q,* $p+n_A$ and $q+n_B$ are listed in the Table I. In particular case, we can derive the probabilities as $P(K_E,K_B,K_A)$, $P(K_E,K_A)$, $P(K_B,K_A)$ or conditional ones $P(K_E,K_B/K_A)$, $P(K_B/K_A)$ and others from the Table I which will be needed in a sequel.

The probabilities $p_m = P(K_A \neq K_B)$ and $p_e = P(K_A \neq K_E)$ calculated by formulas (12), (17) for the different variations of noises are presented in Table II, where p_m denotes the BER in the main (legitimate) channel, and p_e denotes the BER in the wire-tape (eavesdropper) channel.

The last two columns of Table II contain the values p_m, p_e obtained by simulation of KSIP protocol with transmission of $3\cdot10^5$ random numbers *p, q,* n_A, and n_B. We can see that simulation results are in a very good coincidence with the theoretical ones.

TABLE II. THE PROBABILITIES $p_m = P(K_A \neq K_B)$ AND $p_e = P(K_A \neq K_E)$ AGAINST NOISE VARIANCE σ^2

σ^2	p_m	p_e	$P(11)$	p_m (simul)	p_e (simul)
0.1	$1.760\cdot10^{-1}$	$9.749\cdot10^{-2}$	$9.749\cdot10^{-2}$	$1.759\cdot10^{-1}$	$9.751\cdot10^{-2}$
0.2	$2.319\cdot10^{-1}$	$1.339\cdot10^{-1}$	$1.159\cdot10^{-1}$	$2.318\cdot10^{-1}$	$1.338\cdot10^{-1}$
0.3	$2.681\cdot10^{-1}$	$1.595\cdot10^{-1}$	$1.341\cdot10^{-1}$	$2.682\cdot10^{-1}$	$1.594\cdot10^{-1}$
0.4	$2.946\cdot10^{-1}$	1.79510^{-1}	1.47310^{-1}	$2.947\cdot10^{-1}$	$1.797\cdot10^{-1}$
0.5	$3.151\cdot10^{-1}$	$1.959\cdot10^{-1}$	$1.575\cdot10^{-1}$	$3.147\cdot10^{-1}$	$1.959\cdot10^{-1}$
0.6	$3.316\cdot10^{-1}$	$2.098\cdot10^{-1}$	$1.658\cdot10^{-1}$	$3.316\cdot10^{-1}$	$2.097\cdot10^{-1}$
0.7	$3.452\cdot10^{-1}$	$2.218\cdot10^{-1}$	$1.726\cdot10^{-1}$	$3.453\cdot10^{-1}$	$2.218\cdot10^{-1}$
0.8	$3.567\cdot10^{-1}$	$2.323\cdot10^{-1}$	$1.783\cdot10^{-1}$	$3.567\cdot10^{-1}$	$2.324\cdot10^{-1}$
0.9	$3.665\cdot10^{-1}$	$2.416\cdot10^{-1}$	$1.832\cdot10^{-1}$	$3.664\cdot10^{-1}$	$2.416\cdot10^{-1}$
1.0	$3.750\cdot10^{-1}$	$2.5\cdot10^{-1}$	$1.875\cdot10^{-1}$	$3.750\cdot10^{-1}$	$2.500\cdot10^{-1}$

TABLE I. ALL VARIANTS OF POSSIBLE EVENTS WITH CORRESPONDING PROBABILITIES ASSIGNED TO THEM

Probability of events	A			B			Decision	E		Decision
	sign p	sign $q+n_B$	bit K_A	sign q	sign $p+n_A$	bit K_B	$K_A = K_B$?	sign $(q+n_B)\times(p+n_A)$	bit K_E	$K_A = K_E$?
P1	+	+	**0**	+	+	**0**	yes	+	**0**	yes
P2	-	-	**0**	-	-	**0**	yes	+	**0**	yes
P3	+	+	**0**	-	-	**0**	yes	-	**1**	no
P4	-	-	**0**	+	+	**0**	yes	-	**1**	no
P5	+	-	**1**	+	-	**1**	yes	+	**0**	no
P6	-	+	**1**	-	+	**1**	yes	+	**0**	no
P7	+	-	**1**	-	+	**1**	yes	-	**1**	yes
P8	-	+	**1**	+	-	**1**	yes	-	**1**	yes
Q1	+	+	**0**	+	-	**1**	no	-	**1**	no
Q2	-	-	**0**	-	+	**1**	no	-	**1**	no
Q3	+	+	**0**	-	+	**1**	no	+	**0**	yes
Q4	-	-	**0**	+	-	**1**	no	+	**0**	yes
Q5	+	-	**1**	+	+	**0**	no	-	**1**	yes
Q6	-	+	**1**	-	-	**0**	no	-	**1**	yes
Q7	-	+	**1**	+	+	**0**	no	+	**0**	no
Q8	+	-	**1**	-	-	**0**	no	+	**0**	no

The fact that for all values σ^2 the following inequality holds $p_e < p_m$ can also be seen from this Table. It requires to apply further additional subprotocols in order to get the opposite inequality $p_e > p_m$ for execution of the privacy amplification procedure.

The calculation of the joint probabilities $P(11) = P(K_A \neq K_B, K_A \neq K_E) = Q1 + Q2 + Q7 + Q8$ against the different values σ^2 is presented in Table II as well. It is worth to note, that events $K_A \neq K_B$ and $K_A \neq K_E$ occur dependently because $P(K_A \neq K_B, K_A \neq K_E) \neq P(K_A \neq K_B) \cdot P(K_A \neq K_E)$ as far as we can see from Table II.

This fact prompts us that additional subprotocol should be arranged as more sophisticated than simple protocol PIMC considered in the paper [7]. Thus, let us introduce such a protocol in the next section that we called by *Iterative protocol preferently improved of the main channel* (IPIMC), But before we present key bit generation scheme based on hardware generator.

In order to provide a good statistics of the key bits, we suggest using special *hardware random number generator* since it would be impossible to use program-oriented generator (like MT 199937) owing its vulnerable to sequence prediction attack otherwise. We propose to select generators like "Crypton USB-DRN" manufactured by company "Ankad" [8]. Photo of this device is shown in Fig.2. We tested this device over the standard NIST tests [9] and conducted that it passes all of them except for the last two among the list of whole 15 tests.

Fig.2. View of random number generator "Crypton USB-DRN"

III. ITERATIVE IPIMC PROTOCOL AND SUBPROTOCOL (DMEC) FOR DEGRADATION OF BOTH THE MAIN AND EAVESDROPPER CHANNELS

The scheme of the truly random sequence (gamma –γ) transmission elaborated by the hardware generator is shown in Fig.3. As far as we can see user A generates truly random gamma that is XOR-ed with A's raw bit string K_A and the sum is transmitted over public noiseless channel to user B who adds the received string to the raw bit string K_B in order to get

$$U = K_A \oplus \gamma \oplus K_B = K_A \oplus \gamma \oplus K_A \oplus \varepsilon_{AB} = \gamma \oplus \varepsilon_{AB}, \quad (18)$$

where ε_{AB} is the noise string between raw strings K_A and K_B ($\varepsilon_{AB} = K_A \oplus K_B$) and $\oplus$ denotes operation of bitwise modulo two addition.

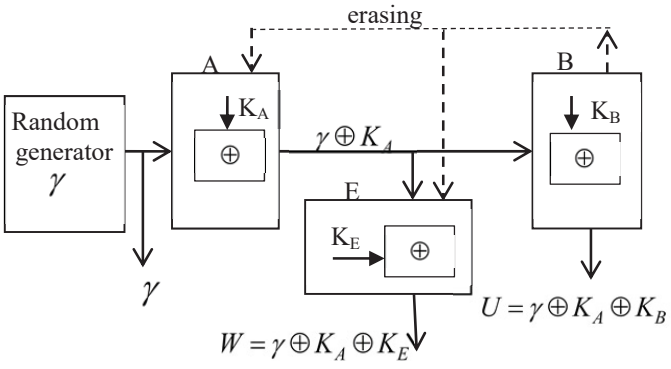

Fig.3. Scheme gamma transmission from use A to use B under the presence of eavesdropper

From now on a string γ will be considered as a final key bit string between users A and B. At the same time, E receives $\gamma \oplus K_A$ over public noiseless channel and is able to add her intercepted bit string possessing her raw string K_E (see Fig 3) as follows

$$W = K_A \oplus \gamma \oplus K_E = K_A \oplus \gamma \oplus K_A \oplus \varepsilon_{AE} = \gamma \oplus \varepsilon_{AE}, \quad (19)$$

where ε_{AE} is the noise string between raw strings K_A and K_E ($\varepsilon_{AE} = K_A \oplus K_E$).

It is worth to note that modulo two bitwise addition operation in relations (18) and (19) is obviously an optimal processing of the received strings due to a minimization of her errors.

We can see from Table 2 the difference between probabilities $P(K_A \neq K_B) = p_m$ and $P(K_A \neq K_E) = p_e$ is not sufficient enough. In order to solve this problem, we propose to apply special protocol that we called *Iterative protocol preferently improved of the mail channel* (IPIMC).

This protocol is performed as follows. User A repeats s times every bit of gamma and transmits s-block to user B over the public noiseless channel. User B is processing the string by (18) and accepts s-block if and only if they consist of equal s bits (zeros or ones). User B erases corresponding block and inform user A about inadequate s-blocks overwise. Moreover, the case when the new probabilities $\tilde{p}_l, \tilde{p}_e$ are not sufficiently diverse from each other may occur. Then PIMC protocol can be repeated assuming that new probabilities are equal to output of PIMC protocol probabilities.

At the same time, eavesdropper E also receives s-blocks with BER p_e and controls public channel where user B informs user A about acceptance or rejection of some blocks that have been announced by user B as the rejected ones. As far as s-blocks are accepted by user B, then eavesdropper is enforced to use a majority rule. This means that E receives bit equal to one if s-block contains more ones than zeros in s-block and bit equal to zero in the opposite case. It is easy to prove that such rule is the optimal one for E.

Let us prove the relations for BERs after a completion one iteration of PIMC protocol for both legitimate users $p_m(s)$ and for eavesdropper one $p_e(s)$.

It is easy to see that BER for legitimate users is

$$p_m(s) = \frac{(p_m)^s}{P_{accept}}, \quad (20)$$

where $P_{accept} = (1-p_m)^s + (p_m)^s$ is the probability of s-block acceptance.

Considering probability $p_e(s)$, it is necessary to take into account that every s-block accepted by E was already accepted by user B. That means, the events are dependent one each other. The BER probability for eavesdropper E after one iteration and even *s*, can be calculated as follows

$$p_e(s) = \frac{1}{P_{accept}} \left[\begin{array}{l} \sum\limits_{i:t(\bar{w}_i)>s/2} p(\bar{w}_i, \bar{u} = 0^s) + 1/2 \sum\limits_{i:t(\bar{w}_i)=s/2} p(\bar{w}_i, \bar{u} = 0^s) + \\ + \sum\limits_{i:t(\bar{w}_i)<s/2} p(\bar{w}_i, \bar{u} = 1^s) + 1/2 \sum\limits_{i:t(\bar{w}_i)=s/2} p(\bar{w}_i, \bar{u} = 1^s) \end{array} \right], \quad (21)$$

where $\bar{u}, \bar{w}$ s-blocks received by B and E, respectively, 0^s - s-block of all zeros, 1^s - s-block of all ones, $t(\bar{w}_i)$ - Hamming weight of vector $\bar{w}_i$.

The join probabilities $p(\bar{w}_i, \bar{u}_j) = \prod_{k=0}^{s-1} p(w_{ik}, u_{jk})$,

where $p(w_{ik}, u_{jk})$ are the joint probabilities which can be found in Table I.

In Table III the results of the BER $p_m(s)$, $p_e(s)$ after the performance evaluation of IPIMC protocol for parameter s=4 and two iterations against different values of noise variance $\sigma^2_{n_A} = \sigma^2_{n_B} = \sigma^2$ calculated by (20), (21) are presented.

TABLE III. THE RESULTS OF CALCULATION $p_m(s), p_e(s)$ BY (20),(21) AGAINST DIFFERENT NOISE VARIANCES σ^2

σ^2	PIMC 1-st iteration		PIMC 2-nd iteration	
	$p_m(s=4)$	$p_e(s=4)$	$p_m(s=4)$	$p_e(s=4)$
0.1	$2.075 \cdot 10^{-3}$	$1.433 \cdot 10^{-3}$	$1.656 \cdot 10^{-11}$	$4.704 \cdot 10^{-7}$
0.2	$8.237 \cdot 10^{-3}$	$5.713 \cdot 10^{-3}$	$4.604 \cdot 10^{-9}$	$7.773 \cdot 10^{-6}$
0.3	$1.769 \cdot 10^{-2}$	$1.233 \cdot 10^{-2}$	$9.803 \cdot 10^{-8}$	$3.758 \cdot 10^{-5}$
0.4	$2.951 \cdot 10^{-2}$	$2.064 \cdot 10^{-2}$	$7.581 \cdot 10^{-7}$	$1.105 \cdot 10^{-4}$
0.5	$4.285 \cdot 10^{-2}$	$3.011 \cdot 10^{-2}$	$3.372 \cdot 10^{-6}$	$2.472 \cdot 10^{-4}$
0.6	$5.707 \cdot 10^{-2}$	$4.026 \cdot 10^{-2}$	$1.061 \cdot 10^{-5}$	$4.666 \cdot 10^{-4}$
0.7	$7.167 \cdot 10^{-2}$	$5.076 \cdot 10^{-2}$	$2.639 \cdot 10^{-5}$	$7.846 \cdot 10^{-4}$
0.8	$8.630 \cdot 10^{-2}$	$6.135 \cdot 10^{-2}$	$5.547 \cdot 10^{-5}$	$1.214 \cdot 10^{-3}$
0.9	$1.007 \cdot 10^{-1}$	$7.176 \cdot 10^{-2}$	$1.029 \cdot 10^{-4}$	$1.766 \cdot 10^{-3}$
1	$1.147 \cdot 10^{-1}$	$8.216 \cdot 10^{-2}$	$1.734 \cdot 10^{-4}$	$2.448 \cdot 10^{-3}$

Table IV shows the same values of $p_m(s), p_e(s)$ as they were in table 3 but obtained by simulation of the PIMS protocol with a transmission of $8 \cdot 10^9$ random numbers *p*, *q*, n_A and n_B according to the KSPI protocol instead (see Fig.1).

Comparing the content of Tables III and IV we can conclude that they are in a good coincidence. After the second iteration, the BER values $p_m(s), p_e(s)$ are decreased significantly but the probability $p_e(s)$ is decreased much less than probability $p_m(s)$.

TABLE IV. THE PROBABILITIES $p_m(s), p_e(s)$ OBTAINED BY SIMULATION OF THE PIMC PROTOCOL WITH S=4 AGAINST DIFFERENT NOISE VARIANCES σ^2

σ^2	PIMC 1-st iteration simulation		PIMC 2-nd iteration simulation	
	$p_m(s=4)$	$p_e(s=4)$	$p_m(s=4)$	$p_e(s=4)$
0.1	$2.14 \cdot 10^{-3}$	$1.45 \cdot 10^{-3}$		$4.36 \cdot 10^{-7}$
0.2	$8.33 \cdot 10^{-3}$	$5.77 \cdot 10^{-3}$		$7.24 \cdot 10^{-6}$
0.3	$1.76 \cdot 10^{-2}$	$1.22 \cdot 10^{-2}$		$3.88 \cdot 10^{-5}$
0.4	$2.94 \cdot 10^{-2}$	$2.05 \cdot 10^{-2}$	$1.06 \cdot 10^{-6}$	$1.11 \cdot 10^{-4}$
0.5	$4.24 \cdot 10^{-2}$	$2.99 \cdot 10^{-2}$	$6.90 \cdot 10^{-6}$	$2.34 \cdot 10^{-4}$
0.6	$5.724 \cdot 10^{-2}$	$4.028 \cdot 10^{-2}$	$2.389 \cdot 10^{-5}$	$4.30 \cdot 10^{-4}$
0.7	$7.213 \cdot 10^{-2}$	$5.078 \cdot 10^{-2}$	$4.542 \cdot 10^{-5}$	$7.54 \cdot 10^{-4}$
0.8	$8.665 \cdot 10^{-2}$	$6.172 \cdot 10^{-2}$	$7.157 \cdot 10^{-5}$	$1.283 \cdot 10^{-3}$
0.9	$1.005 \cdot 10^{-1}$	$7.167 \cdot 10^{-2}$	$7.954 \cdot 10^{-5}$	$1.914 \cdot 10^{-3}$
1	$1.154 \cdot 10^{-1}$	$8.287 \cdot 10^{-2}$	$3.66 \cdot 10^{-4}$	$2.494 \cdot 10^{-3}$

This means that the third iteration is not needed for application at all. However, an advantage of this result can only be reached under the condition that both these probabilities occur not very small. In order to make the difference between probabilities large enough, it is necessary to perform protocol *degradation of both channel* (main and eavesdropper) after a completion of IPIMC protocol in sequel.

Let us denote such a protocol by abbreviation DBC. It can be realized by many ways, but the simplest method is to add modulo 2 adjacent bits of outputs sequences after a completion of IPIMC protocol. Then we get

$$\tilde{\gamma}_i = \gamma_{2i} \oplus \gamma_{2i+1}, \quad \tilde{u}_i = \tilde{u}_{2i} \oplus \tilde{u}_{2i+1}, \quad \tilde{w}_i = \tilde{w}_{2i} \oplus \tilde{w}_{2i+1},$$

where $\tilde{\gamma}_i, \tilde{u}_i, \tilde{w}_i$ i = 1, 2, ... l are output bits after a completion of PIMC protocol by A, B, and E respectively. Protocol DBC can be repeated iteratively l times and denoted then as IDBC with a notation of last output sequences by $\tilde{\gamma}_i^{\nu}, \tilde{u}_i^{\nu}, \tilde{w}_i^{\nu}$.. Let us $\tilde{p}_m^{(\nu)}, \tilde{p}_e^{(\nu)}$ be the BER after application of IDBC protocol by B and E, because A in line with our assumption has in his disposition error free key sequence γ. It is easy to conclude that such BERs are:

$$\tilde{p}_m^{(\nu)} = 2\tilde{p}_m^{(\nu-1)}(s)(1-\tilde{p}_m^{(\nu-1)}(s)),$$

$$\tilde{p}_e^{(\nu)} = 2\tilde{p}_e^{(\nu-1)}(s)(1-\tilde{p}_e^{(\nu-1)}(s)), \ \nu = 1,2,\ldots$$

In Fig.4 the BER probabilities for the main and wiretap channels given different number of iterations for IDBC protocol and two iteration of IPIMC protocol against square root of variances for additive artificial noise σ are presented.

IV. ESTIMATIONS OF RELIABILITY AND SECURITY FOR THE PROPOSED KSP

We can see from Fig.4 that for some σ-interval 0.26 - 0.53 the acceptable diverse between BER in the main and the wire-tap channels can be obtained. At the boundary points of this interval we have : $\tilde{p}_m^{16} = 5 \cdot 10^{-8}$, $\tilde{p}_e^{16} = 6.13 \cdot 10^{-3}$ for $\sigma = 0.26$, $\nu = 16$ and $\tilde{p}_m^{16} = 4.2 \cdot 10^{-3}$ $\tilde{p}_e^{16} = 0.49$ for $\sigma = 0.53$, $\nu = 16$.

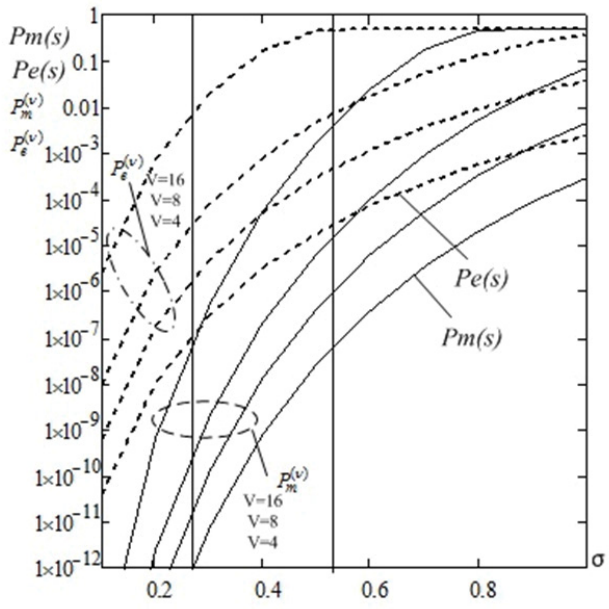

Fig. 4. The BER probabilities for the main and wiretap channels for 4,8,16 iterations of IDBC protocol and two iteration of IPIMC protocol against square roots of additive noise variances

The following question arises. Is it really necessary to apply some error correction code in order to improve the reliability of the key distribution between legitimate users while taking into account that a transmission of the check symbols of this code results in additional information leakage to eavesdropper's key bits? It is obviously that a decision depends on a selection of parameters such as noise variance σ^2.

Let us consider two scenarios – the one without error correcting code and one with it. For the first scenario assuming $\sigma = 0.26$ and two iterations of IPIMC protocol with s=4 we get (see Fig.4) $p_m(s=4) = 7.63 \cdot 10^{-13}$, $p_e(s=4) = 9.4 \cdot 10^{-8}$. After applying of 16 iterations in IDMC protocol, we obtain $\tilde{p}_m^{16} = 5 \cdot 10^{-8}$ $\tilde{p}_e^{16} = 6.13 \cdot 10^{-3}$. Then, the probability P_{ed} of any error in the shared 256 bits key is $P_{ed} = (1-(1-\tilde{p}_m^{(16)})^{256})$ = 1.2*10^{-5}.

We believe that such P_{ed} guarantes sufficiently acceptable reliability of key distribution.

Let us suppose that the value of information leakage to eavesdropper can be upper bounded by $I = 10^{-10}$ bit. (we note that a connection of this value I with the probability of error for an optimal decoding by eavesdropper is based on Fano inequality [10] presented in [11]. In order to ensure the value of I=10^{-10} bits it is necessary to execute so called *privacy amplification procedure* determined in the paper [12] by U. Maurer. In [12] he has proven the theorem which states that if one applies string hashing to the initial key based on the universal$_2$ class of random hash functions, then the amount of information at the output of such hash function is upper bounded as follows

$$I \le \frac{2^{-(k-l-t_c)}}{\ln(2)}, \qquad (22)$$

where k is input length of key bit sequence, l is the final length of key bit sequence,

$$t_c = k + k\,log((\tilde{p}_e^{16})^2 + (1-(\tilde{p}_e^{16}))^2) \qquad (23)$$

is the Renyi information.

If we let the final key length to be 256 bits which is sufficient enough for cryptographic standards like AES, GOST, etc. [3], then the initial length of string k can be found from (22), (23) and given known value of $\tilde{p}_e^{16}$, it will be equal to 16400.

In order to estimate KSP efficiency it is necessary to calculate the key rate. The last value can be found for the general case when error correction code is used as

$$R = \frac{l}{k/(R_q \cdot R_m^i(s) \cdot R_{DMC}) + r}. \qquad (24)$$

If the error correction code is not used, then (24) can be simplified to

$$R = \frac{l \cdot R_q \cdot R_m^i(s) \cdot R_{DMC}}{k}, \qquad (25)$$

where R_q is the code rate for a transformation of real values $p + n_A, q + n_B$ into binary strings for transmission over the main channel later. In partcular case, $R_q = 1/a$, where a is the amount of binary digits for $p + n_A, q + n_B$ used during the transformation. Hence,

$R_m^{(i)} = \frac{1}{s^t} P_{ac}(1) \cdot P_{ac}(2) \cdots P_{ac}(t)$, where $P_{ac}(i)$ is the probability of the s-block acceptance by B in iteration number i, t is the total amount of iterations performed in IPIMC protocol and s is the block length of IPIMC protocol, R_{DMC} is the rate of DMC protocol.

It follows from definition of that value that $R_{DMC} = 1/2^v$.

By substituting of all parameters for codeless scenario into (25) we get $R = 9.73 \cdot 10^{-10}$. This means that in order to share 256 key bits it is necessary to form $1.03 \cdot 10^9$ raw key bits transmitting over the ordinary Internet. Taking 10-100 Mbit/s as the widespread transmission rate over the Internet channel the execution of KSP takes about 10-100 sec.

If it is necessary to shorten the time of KSP execution the error correcting code implementation can be used. But in such

a scenario we can take into account that eavesdropper is able to intercept all check sybols transmitting over the public channel and, hence, to get some additional information about the shared key string. In fact, we can apply privacy amplification procedure as well but the calculating of the information leakage to eavesdropper should be slightly different than in (22). We have to use so called *modified privacy amplificated procedure* (MPAP) discribed in [13]. It implies applying the hash functions taken from the unuversal$_2$ class and consequently in a "puncturing" of some bits (see [13] for details). After evaluation of the MPAP performance we get from (22) the following bound

$$I \le \frac{2^{-(k-l-t-r)}}{\ln(2)\cdot 0.42}\ , \qquad (26)$$

where all items in (26) are the same as in (22), exept for r that is the number of check bits of the error correcting code and constant factor 0.42. In the coding-based scenario it is more effective to select the parameter $\sigma = 0.53$ and then we calculate that $\tilde{p}_m^{16}(s=4) = 4.2\cdot 10^{-3}$ $\tilde{p}_e^{16}(s=4) = 0.49$.

In order to decrease the probability of the incorrect block decoding probability P_{ed} the linear error correcting *(n,k,d)*-code is considered where n is a block length, k=256 is the amount of information symbols equal to final key length and with minimal code distance d. Then we obtain

$$P_{ed} = 1 - \sum_{i=0}^{d/2} \binom{n}{i} \left(p_m^{(16)}(s)\right)^i \left(1-\left(p_m^{(16)}(s)\right)\right)^{n-i}. \qquad (27)$$

Estimation of d for such a code can be found by Varshamov-Gilbert bound [14]

$$R_c \ge 1 - g(\frac{d}{n}), \qquad (28)$$

where $R_c = k/n$ is the code rate and $g(x) = -x\log x - (1-x)\log(1-x)$ is the entropy function.

Considering formulas (27), (28) we can select a code with parameters n=613, k=482, r=161 and d=27 which provides $P_{ed} = 3.5\cdot 10^{-6}$. The bound (26) implies information leakage to eavesdropper $I_0 = 1.2\cdot 10^{-10}$.

The key rate estimated by (24) is

$$R_{key} = \frac{256}{482 / \left(\frac{1}{8}\right)\left(\frac{0.301}{4}\right)\left(\frac{0.981}{4}\right)\left(\frac{1}{2^{16}}\right) + 161} = 1.87\cdot 10^{-8}\ .$$

It means that in order to share 256 bit key it is nessary to form and transmit over the ordinary Internet channel $5.45\cdot 10^7$ raw key bits. Thus, if the transmittion rate is 10-100 Mbit/s it takes about 0.55 - 5.5 sec. which it is much better than time spent on codeless scenario.

We can remark that in reality the use of an error correcting code with constructive encoding/decoding algorithm is necessary. It was already demonstrated before in [11] how *lower density parity check codes (LDPC*) are useful for that case.

For proposed KSP it is required to perform authentication procedure as for any such protocol. The adversaries could impersonate legitimate users during communication over the public channels and eventually share with them a common key otherwise. It is also possible to use different authentication methods, for example, short keys, Needham-Schroder protocol [15], pairing procedure "face to face" device matching etc. [16].

V. Conclusion

The current paper is generally completes a series of our papers [7], [11], [17] devoted to the problem of the keyless cryptography, namely to *key sharing protocol* intended for *public communication channels*. But in contrast to our previous publications and the results of the other authors we have proposed there more effective protocol under the conditions of eavesdropping. We have refused from the communication using matrices over the channels in favor of integer communication which allowed us to decrease the channel traffic and simplify implementation. It is suggested to execute our protocol over the ordinary public noiseless channels with constant parameters like Internet. Such a protocol scenario offers the wide opportunities to provide confidentiality for ordinary Internet users having in disposition strong encryption/decryption standards, like AES or GOST while having nothing means for a distant key distribution if they did not exchanged by the nature based key in advance. We believe to investigate the following actual problems in the future:

- KSP users authentication;

- the KSP software implemented for the use by not qualified, ordinary users.

References

[1] W. Diffie, M. Hellman, "New Directions in Cryptography", *IEEE Trans. Inf. Theory*, vol. 22, no. 6, 1976, pp. 644-654.

[2] P. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". *SIAM Journal on Scienticand Statistical Computing.* 1997;5(26), pp.1484–1509.

[3] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, ISBN 0-8493-8523-7, The CRC Press series on discrete mathematics and its applications,USA: CRC Press, 2000 N.W. Corporate Blvd., Boca Raton, FL 33431-9868, 1997.

[4] C. H. Bennett,et al., "Experimental quantum cryptography", *Journal of. Cryptol*., vol. 5, N1 (1992), pp. 3-28.

[5] A. Mukherjee, et al., "Principles of Physical Layer Security in Multiuser Wireless Network": A Survey, 2014, arXiv:1011.3754.3 [cs. IP].

[6] J.M.Wallace and D.K.Sharma. "Automatic-Secret Keys from Reciprocal MIMO Wireless Channel Measurements and Analysis", *IEEE Transactions on Information Forensics and Security*, 5:3 (2010), pp. 381-392.

[7] V. Korzhik, V. Starostin, M Kabardov, A. Gerasimovich, V. Yakovlev, A. Zhuvikin. "Protocol of key distribution over public noiseless channels executing without cryptographic assumption" , *International Journal of Computer Science and Application*, 2020, vol.17, no 01, pp.1-14.

[8] KBDZ.469435.052 RE. Podorozhnyi I. V. "Obzor apparatnykh generatorov sluchainykh chisel", *Molodoy uchenyi*. URL: http://moluch.ru/archive/105/24688/ (date of access: 24.06.2020) (In Rus).

[9] L. Bassham et al. ”Review of statistical tests. Suite for random and pseudorandom number generators for cryptographic applications”. SP 800-22, 2010.

[10] R. Fano. *Transmission of Information*. *A statistical theory of communication*, Willy Bullisher, 1961.

[11] V. Korzhik, V.Starostin, V. Yakovlev, M. Kabardov, A. Gerasimovich, .A. Zhuvik. “Information Theoretically Secure Key Sharing Protocol Executing with Constant Noiseless Public Channels”. *Mathematical problems of cryptography*, 2021, T.12, N 3 pp. 31-47.

[12] U. Maurer. “Secret key agreement by public discussion from common information”, *IEEE Transactions on Information Theory*, 39:3 (1993), pp. 733-742.

[13] V. Korjik, G. Morales-Luna, and V. Balakirsky. “Privacy amplification theorem for noisy main channel”, *Lecture Notes in Computer Science*, 2200 (2001), pp. 18-26.

[14] M. Williams, N.S. Sloane. *The theory of Error Correcting codes codes.* Bell lab. 1977.

[15] R.M. Needham and M.D. Schroeder. “Using Encryption for authentication in Large Network of computers” ACM, 21 (1978), pp. 993-999.

[16] R Jin. et al., “MagPairing: Pairing Smartphones in close proximity using magnetometer”, *IEEE Trans. of Information Forensics and Security*, 6 (2016), pp. 1304-1319.

[17] .V.Yakovlev, V. Korzhik, G. Morales-Luna. “Key distribution protocols based on noisy channels in presence of an active adversary: Conventional and new versions with parameter optimization”, *IEEE Transactions on Information Theory*, 54:6 (2008), pp. 2535-2549.