Channel Traffic Minimizing Key Sharing Protocol Intended for the Use over the Internet and Secure without any Cryptographic Assumptions

Viktor Yakovlev, Valery Korzhik
viyakov4@gmail.com, korzhikvalery11@ gmail.com

Vladimir Starostin, Alexey Lapshin, Aleksei Zhuvikin
star.vs.47@gmail.com, SCodeC.LA@gmail.com, mail@zhuvikin.com

***Abstract* — A further development of the key sharing protocol presented at previous FRUCT-31 session is proposed in the current paper. In contrast to the previous protocol executing by an exchange of real integers over Internet channel, a new protocol version operates with binary bits and all operations are modulo two additions. Such version significantly decreases channel traffic required for key sharing and a complexity of signal processing. The full description of protocol is given. Optimality of signal processing proposed in the paper for eavesdropper is proved.**

I. INTRODUCTION

We have already published a number of papers devoted to key sharing problem through communication channels [1 - 5]. All mentioned approaches to solving the key sharing problem were based on notion of *physical layer security* where it is exploited some difference between properties of channel connecting legitimate and eavesdropper users. Therefore, in the paper [1] we execute a difference between signal/noise ratio (SNR) in such channels. In the paper [2], the difference in wave propagation for multi ray channels is used, in paper [3] channels with different fading parameters and randomized smart antennas are operated. In the paper [4] communication channel is noiseless, but legitimate users are creating noise artificially. The difference between the paper [4] and [5] is that in the first case we proposed to exchange between users by complex matrices, and the second – the exchange by real numbers. Thus, we considered some tradeoff between complexity of signal processing and channel traffic volume. In fact, a transition to real numbers from matrices requires using additional protocol, which in [5] was called *the degradation of both channels* (DBC).

Eventually, in the current paper we replace real-valued artificial noises to binary artificial noises that even simplifies signal generation of the protocol in comparison with protocol presented in the paper [4].

It is necessary to keep in the mind that we ignore here such well-known key sharing protocol (KSP) based on public key cryptosystems (like Diffie-Hellman protocol [6] and others).

The reason for this is the ability to crack some public key cryptosystems (PKC) using quantum computers in the future. It is worth to note that although at present practical implementation of such computers is questionable (see [7] and others), but such devices can be created perhaps in the future. Thus, this can lead to polynomial complexity of solving problems such as integer factorization, logarithm, and so on.. Moreover, there appears even special term *post quantum cryptosystems* (PQC), which cannot be broken in polynomial time by quantum computers. Into class of PQC can be included, for example, McElise or lattice-based cryptosystems [8]. But the complexity of signal processing for them is usually much greater than the complexity of conventional PCSs. With regard to the KSP presented in the current paper, we prove convincingly that its security does not depend on any cryptographic assumptions. In fact, the security of our KSP is due to the negligible leakage of Shannon's information to the interceptor after the execution of the protocol. Therefore, KSP can be called theoretically secure. As for its complexity in terms of signal processing and the amount of channel traffic required to implement it, it looks like least simple one, since we are dealing with binary sequences.

Section II presents a supernova KSP with binary sequences exchanged over public constant channels (such as the Internet) and proves formulas for raw bit error probabilities in both the legitimate channel (p_m) and the eavesdropping channel (p_m). This section fixes also some of the previous security holes in KSP, namely proving that an eavesdropper cannot improve signal processing to break it. Section III describes *preferently improved of the mail channel* (PIMC) protocol and DBS protocol including their iterative versions.

In section IV the reliability and security of the proposed KSP are proved after application of error correction codes and privacy amplification procedures. Section V discuses some additional problems of KSP. In particular, implementation of random number generator, authentication of users and software realization. Summarization of the main results and some actual problems for further investigation are given in section VI.

II. DESCRIPTION OF SUPERNOVA KSP EXECUTING EXCHANGE WITH BINARY SEQUENCES OVER NOISELESS CHANNELS

Scenario corresponding to supernova KSP is presented in Fig. 1. (By the way, we called our KSP by “supernova” in order to strike its difference with our previous KSP presented recently

in [5].). We can see from that Figure that both legitimate users A and B initially generate and store binary random sequences $\delta_A, \gamma_A, \delta_B, \gamma_B$ with the following properties:

$$\begin{aligned} &P\{\delta_A = 0\} = P\{\delta_A = 1\} = 1/2, \\ &P\{\delta_B = 0\} = P\{\delta_B = 1\} = 1/2, \\ &P\{\gamma_A = 0\} = P\{\gamma_B = 0\} = 1-p, \\ &P\{\gamma_A = 1\} = P\{\gamma_B = 1\} = p. \end{aligned} \quad (1)$$

Bits are i.i.d. in any sequence and all sequences $\delta_A, \gamma_A, \delta_B, \gamma_B$ of the bits are mutual independent, $\oplus$ is bitwise modulo 2 addition.

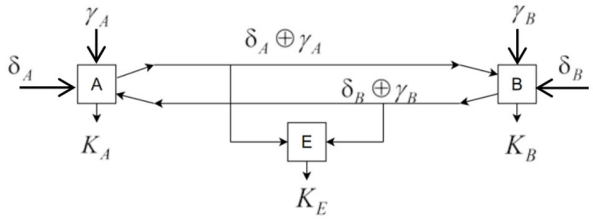

Fig.1. Scenario of the raw bit formation protocol based on exchange by binary sequences $\delta_A \oplus \gamma_A$, $\delta_B \oplus \gamma_B$ over public noiseless channel with feedback

Next A transmits bitwise sum of binary sequences $\delta_A \oplus \gamma_A$ to B over channel A→B and B transmits bitwise sum of binary sequences $\delta_A \oplus \gamma_B$ to A over the channel B→A. Eavesdropper E is able to intercept these sequences over noiseless channels. Legitimate users A and B compute raw key bit sequences following to the rule:

$$K_A = \delta_A \oplus \delta_B \oplus \gamma_B \quad (2)$$

$$K_B = \delta_B \oplus \delta_A \oplus \gamma_A \quad (3)$$

(We call the sequences "raw" key bit ones because they must be further converted into other binary sequences that can be called “valid” keys as long as they satisfy the conditions of reliability and security. (see further section III and IV).

We believe that eavesdropper E estimates the raw key bit sequence as follows:

$$K_E = \delta_A \oplus \delta_B \oplus \gamma_A \oplus \gamma_B \quad (4)$$

Using relations (1) - (4) it is easy to find the probabilities of error in raw key sequences both for legitimate users, that we call the *BER* in the *main channel* (p_m), and the error probabilities between user A (say the “main” user) and eavesdropper E, that we call *BER* in the *eavesdropper channel* (p_e):

$$\begin{aligned} p_m &= P\{K_B \neq K_A\} = P\{\gamma_B \oplus \gamma_A = 1\} = P\{\gamma_B = 0\} \cdot P\{\gamma_A = 1\} + \\ &+ P\{\gamma_B = 1\} \cdot P\{\gamma_A = 0\} = (1-p)p + p(1-p) = 2p(1-p), \end{aligned} \quad (5)$$

$$p_e = P\{K_E \neq K_A\} = P\{\gamma_A = 1\} = p. \quad (6)$$

Example. Let us take p=0.1, then we get from (5), (6) $p_m = 0.18$, $p_e = 0.1$. This simple example shows that after a completion of KSP we can get even eavesdropper channel some better than the main one. Therefore the goal of the next sequence processing is divers p_m and p_e in such a way to provide opposite inequality ($p_m < p_e$) but ($p_e \geq p_0$), where p_0 is some given value. Such protocols: *iterative protocol preferently improved of the main channel (IPIMC*) and degradation of both channel *(DBC*) will be described in the Section III. The remainder of Section II will be devoted to design of special table describing interconnection between the probabilities of different random values, which will later be used to theoretical prove of formulas for p_m and p_e, obtained after applying of the protocols mentioned above. (It is worth to note that such theoretical proof becomes possible only after a replacing of primary matrix exchange protocol (see [4]) to real value exchanging ([5]) or binary sequence exchanging in the current paper.)

In Table I are presented the probabilities of vector random value $x = (\delta_A, \delta_B, \gamma_A, \gamma_B)$.

We note that $P_1 = P_5 = P_9 = P_{13}$, $P_2 = P_6 = P_{10} = P_{14}$, $P_3 = P_7 = P_{11} = P_{15}$, $P_4 = P_8 = P_{12} = P_{16}$.

Hence, let us use in the future only notations P_1, P_2, P_3, P_4, taking into account that

$$P_1 + P_2 + P_3 + P_4 = 1/4, \quad \sum_{i=1}^{16} P_i = 1. \quad (7)$$

Taking the data from Table 1 it is easy to find the probability distribution for all raw keys (K_E, K_B, K_A) and also the probabilities for key disagreements. They are presented in Table II.

TABLE II. THE PROBABILITIES OF RAW KEY BIT COMBINATIONS

	K_E	K_B	K_A	*Combination probabilities* (K_E, K_B, K_A,)	*Key disagreements*	
					$K_B \oplus K_A$	$K_E \oplus K_A$
1	0	0	0	$P1 + P13 = \frac{1}{2}(1-p)^2$	0	0
2	0	0	1	$P7 + P11 = \frac{1}{2}(1-p)p$	1	1
3	0	1	0	$P6 + P10 = \frac{1}{2}(1-p)p$	1	0
4	0	1	1	$P4 + P16 = \frac{1}{2}p^2$	0	1
5	1	0	0	$P8 + P12 = \frac{1}{2}p^2$	0	1
6	1	0	1	$P2 + P14 = \frac{1}{2}(1-p)p$	1	0
7	1	1	0	$P3 + P15 = \frac{1}{2}(1-p)p$	1	1
8	1	1	1	$P5 + P9 = \frac{1}{2}(1-p)^2$	0	0
				Sum of probabilities $(1-p)^2 + p^2 + 2(1-p)p = 1$		

Using the data taken from Table II it is easy to find joint probability distribution $P(K_B, K_A)$:

$$\begin{aligned} &P(K_B, K_A) = P_{BA}(00) = \frac{1}{2}((1-p)^2 + p^2), \\ &P(K_B, K_A) = P_{BA}(01) = (1-p)p, \\ &P(K_B, K_A) = P_{BA}(10) = (1-p)p, \\ &P(K_B, K_A) = P_{BA}(11) = \frac{1}{2}((1-p)^2 + p^2) \end{aligned} \quad (8)$$

TABLE I. THE PROBABILITIES OF VECTOR RANDOM VALUE X

	$x=(\delta_A,\delta_B,\gamma_A,\gamma_A)$				The probabilities of random values $(\delta_A,\delta_B,\gamma_A,\gamma_A)$	*Bits of raw key*		
№	δ_A	δ_B	γ_A	γ_B		$K_E=\delta_A\oplus\gamma_A\oplus\delta_B\oplus\gamma_B$	$K_B=\delta_A\oplus\gamma_A\oplus\delta_B$	$K_A=\delta_A\oplus\delta_B\oplus\gamma_B$
1	0	0	0	0	$P1=\frac{1}{4}\cdot(1-p)^2$	0	0	0
2	0	0	0	1	$P2=\frac{1}{4}\cdot(1-p)p$	1	0	1
3	0	0	1	0	$P3=\frac{1}{4}\cdot(1-p)p$	1	1	0
4	0	0	1	1	$P4=\frac{1}{4}\cdot p^2$	0	1	1
5	0	1	0	0	$P5=\frac{1}{4}\cdot(1-p)^2$	1	1	1
6	0	1	0	1	$P6=\frac{1}{4}\cdot(1-p)p$	0	1	0
7	0	1	1	0	$P7=\frac{1}{4}\cdot(1-p)p$	0	0	1
8	0	1	1	1	$P8=\frac{1}{4}\cdot p^2$	1	0	0
9	1	0	0	0	$P9=\frac{1}{4}\cdot(1-p)^2$	1	1	1
10	1	0	0	1	$P10=\frac{1}{4}\cdot(1-p)p$	0	1	0
11	1	0	1	0	$P11=\frac{1}{4}\cdot(1-p)p$	0	0	1
12	1	0	1	1	$P12=\frac{1}{4}\cdot p^2$	1	0	0
13	1	1	0	0	$P13=\frac{1}{4}\cdot(1-p)^2$	0	0	0
14	1	1	0	1	$P14=\frac{1}{4}\cdot(1-p)p$	1	0	1
15	1	1	1	0	$P15=\frac{1}{4}\cdot(1-p)p$	1	1	0
16	1	1	1	1	$P16=\frac{1}{4}\cdot p^2$	0	1	1

and joint probability distributions of error for E and B (See Table III).

We can see from Table III that the errors in the main channels and in eavesdropper channel are dependent. Example: $P_{em}(11)=(1-p)p\neq p_e\cdot p_m=2p(1-p)$). It would be interesting to find the conditional probability

$$P(K_E,K_B/K_A)=\frac{P(K_E,K_B,K_A)}{P(K_A)}=2P(K_E,K_B,K_A).$$

TABLE III. JOINT PROBABILITY OF ERRORS FOR E AND B

$K_E\oplus K_A$	$K_B\oplus K_A$	*Probabilities of coincidence-mismatch bits of the key* $p_{em}=P(K_E\oplus K_A,K_B\oplus K_A)$
0	0	$p_{em}(00)=(1-p)^2$
1	0	$p_{em}(10)=p^2$
0	1	$p_{em}(01)=(1-p)p$
1	1	$p_{em}(11)=(1-p)p$

Using Table II, let us find the above probability and write them in Table IV.

TABLE IV. THE CONDITIONAL PROBABILITIES $P(K_E,K_B/K_A)$

№	*Conditional probabilities* $P(K_E,K_B/K_A)$	*The formulas for calculating of probabilities*
1	$P_{EBA}(00/0)$	$(1-p)^2$
2	$P_{EBA}(01/0)$	$(1-p)p$
3	$P_{EBA}(10/0)$	p^2
4	$P_{EBA}(11/0)$	$(1-p)p$
5	$P_{EBA}(00/1)$	$(1-p)p$
6	$P_{EBA}(01/1)$	p^2
7	$P_{EBA}(10/1)$	$(1-p)p$
8	$P_{EBA}(11/1)$	$(1-p)^2$

III. DESCRIPTION OF IPIMC AND DBC PROTOCOLS

We believe that truly random binary sequence (say "gamma" (γ)) is generated by the hardware generator at the user A side, is XOR-ed with A's raw bit string K_A and the sum is transmitted over public noiseless channel to user B as it is showed in Fig 2.

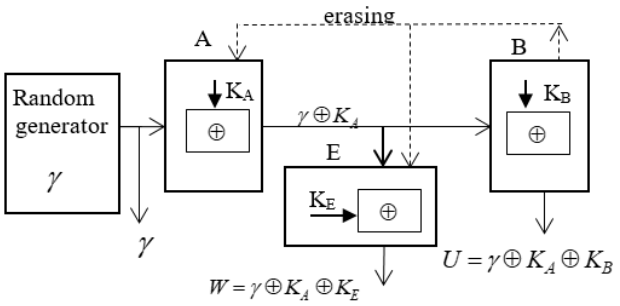

Fig.2. Scheme of gamma transmission from A to B under interception by eavesdropper E

User B adds the received string by modulo 2 to his raw bit string K_B in order to get

$$U = K_A \oplus \gamma \oplus K_B = K_A \oplus \gamma \oplus K_A \oplus \varepsilon_{AB} = \gamma \oplus \varepsilon_{AB}, \quad (9)$$

where ε_{AB} is the noise string between raw string K_A and K_B .

From now on, a string γ is considered as a final key between users A and B. At the same time E receives $\gamma \oplus K_A$ over public noiseless channel adds her intercepted bit string with her processing raw string K_E to get

$$W = K_A \oplus \gamma \oplus K_E = K_A \oplus \gamma \oplus K_A \oplus \varepsilon_{AE} = \gamma \oplus \varepsilon_{AE}, \quad (10)$$

where ε_{AE} is the noise string between raw string K_A and K_E .

Let us to emphasize, that γ is the message that everyone wants to know.

Lemma 1.

Suppose that $0 \le p < 0.5$.

Then, the decision rule W (10), chosen by eavesdropper, on γ bits is optimal because it results in the smallest bit error given the binary intercepted bits $X = \delta_A \oplus \gamma_A$, $Y = \delta_B \oplus \gamma_B$, $Z = \gamma \oplus K_A = \gamma \oplus \delta_A \oplus \delta_B \oplus \gamma_B = \gamma \oplus \delta_A \oplus Y$ are known.

Proof.

Eve gets immediately $\gamma = Z \oplus \delta_A \oplus Y$. The last step is the following: eavesdropper has to evaluate δ_A as function on the only measurable quantity available for this $X = \delta_A \oplus \gamma_A$: $\delta_A \approx f(X)$. Thus, it is necessary to choose among four Boolean functions of one variable: *f(X)*= $\{0, 1, X, X \oplus 1\}$.

Selecting $\delta_A = 0$, one easily gets *BER* in the *eavesdropper channel:* $p_e = 0.5$. The same holds for $\delta_A = 1$. If $\delta_A = X$, then we find, that error probability is $p_e = P(\gamma_A = 1) = p$ and $p_e = P(\gamma_A = 0) = 1 - p$, if $\delta_A = X \oplus 1$. Comparing error probabilities $0.5, p, 1-p$, we conclude, that the choice $\delta_A = X$ is the best, if $\begin{cases} 0 \le p < 0.5 \\ p < 1 - p \end{cases}$ or if $0 \le p \le 0.5$. Thus, the best approximation of truly random binary sequence "gamma" is

$$\gamma \approx Z \oplus X \oplus Y = \gamma \oplus K_A \oplus X \oplus Y = \gamma \oplus K_A \oplus K_E = W$$

Otherwise, if $0.5 < p \le 1$, Eva decides, that the bit string of interest must be approximated as follows: $\gamma = W \oplus 1$. ■

We can see from (5) and (6) and an example below that raw BER p_m is not small enough for reliable key exchange whereas BER between A and E p_e is too much to provide the desired key string security. In order to overcome such problems we propose to apply special protocols, that we call IPIMC and IDBC.

Subprotocol IPMC

Let us consider initially one iteration of this protocol. This protocol is performed as follows. User A repeats *s* times every bit of gamma (γ) and transmits such *s*-blocks to user B over the public noiseless channel. User B is processing the string by (9) and accepts s-block if and only if it consists of equal s bits (all zeroes or all ones). User B erases rejected blocks and inform user A on accepting or rejecting the blocks. Moreover the case when the new probabilities p_m, p_e are not sufficiently different from each other, the protocol can be repeated provided that the new input probabilities are equal to the output probabilities at the first iteration of the protocol.

It is easy to see that BER for legitimate users at the first iteration of protocol is:

$$p_m^{(1)}(s) = \frac{p_m{}^s}{P_{accept}}, \quad (11)$$

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

At the second iteration of IPIMC protocol we get

$$p_m^{(2)} = \frac{\left(p_m^{(1)}(s)\right)^s}{P_{accept}^{(2)}}, \quad (12)$$

where $p_{accept}^{(2)} = \left(1 - p_m^{(1)}\right)^s + \left(p_m^{(1)}\right)^s$.

In a similar manner, it is possible to find the output probabilities $p_m^{(l)}(s)$ after *l* iterations.

As for eavesdropper E, she knows which s-blocks legitimate users accept, but for her they do not always consist of all zeros or ones. Moreover, the BER for E is different from p_e because of knowing that B has already accepted that s-block. Optimal decision rule for eavesdropper is presented by the following Lemma:

Lemma 2.

If legitimate users perform protocol IPIMC, then optimal decision rule for eavesdropper E given user B has accepted that block, is *majority rule*: for odd numbers, take a decision that bit equal to one if the number of symbols ones in the s-block more the number of zeroes and equal to zero otherwise. If s is even and the number of zeros is equal to the number of ones, then take a decision about bit equal to one or zero randomly with the probability ½.

Proof.

We can see from the relation (4) that errors in the symbols of the s-block received by E are mutual independent but under the condition, that B has accepted this s-block, the BER for E can be different then $p_e = p$ given by (6). Hence, the errors on the binary symbols of s-block received by E obey to *Bernoully scheme*. Therefore, the probability for sequence of errors containing in s-block ω received by E can be presented as follows:

$$p(\overline{w}) = \hat{p}_e^t (1-\hat{p}_e)^{s-t}, \tag{13}$$

where $\hat{p}_e$ is the BER for E given B accepted this s-block, t is the number of errors on s-block, $\overline{w}$ - accepted s- block by E.

It is well known [10] that Kotelnikov optimal decision rule is equivalent to a pair of inequalities:

$$\begin{cases} 0 \; if \;\; p(\overline{w}/0^s) \geq p(\overline{w}/1^s) \\ 1 \; if \;\; p(\overline{w}/0^s) < p(\overline{w}/1^s), \end{cases}$$

where 0^s, 1^s the transmitted s-block consisting of s zeros or s ones respectively.

Since the function in the right side of (13) is monotonically decreasing on $i \in (1,s)$ for any $\hat{p}_e$ and s, then

$$p\{\overline{w}/0^s\} > p\{\overline{w}/1^s\}$$

if and only if

$$\#\{0 \in \overline{w}\} > \#\{1 \in \overline{w}\}, \tag{14}$$

where $\#\{0 \in \overline{w}\}$ means the number of zeroes in $\overline{w}$, $\#\{1 \in \overline{w}\}$ is the number of ones in $\overline{w}$.

On the other hand the condition (14) is equivalent to "*majority rule*", that proves lemma 2, because if relation (14) to replace by equality, then we have no other way to take decision as to select it randomly with the probability ½. ■

We can see from relations (2) and (4) that events $K_A = (0,1)$ and $K_B = (0,1)$ are dependent for each bit in the s-block. After simple, but tedious transformations, we get the following relation for the BER $p_e^{(1)}$ of the eavesdropper E after a completion of the first iteration of the IPIMC protocol:

$$p_e^{(1)} = p_{em}^{(1)}(10) + p_{em}^{(1)}(11), \tag{15}$$

where

$$p_{em}^{(1)}(10) = \frac{\sum_{i=s/2+1}^{s} C_s^i p_{em}(10)^i \, p_{em}(00)^{s-i} + 1/2 C_s^{s/2} p_{em}(10)^{s/2} \, p_{em}(00)^{s/2}}{P_{accept}^{(1)}}, \tag{16}$$

$$p_{em}^{(1)}(11) = \frac{\sum_{i=s/2+1}^{s} C_s^i p_{em}(11)^i \, p_{em}(01)^{s-i} + 1/2 C_s^{s/2} p_{em}(11)^{s/2} \, p_{em}(01)^{s/2}}{P_{accept}^{(1)}}. \tag{17}$$

The BER $p_e^{(2)}$ of the eavesdropper E after a completion IPIMC protocol is:

$$p_e^{(2)} = p_{em}^{(2)}(10) + p_{em}^{(2)}(11), \tag{18}$$

where

$$p_{em}^{(1)}(10) = \frac{\sum_{i=s/2+1}^{s} C_s^i p_{em}^{(1)}(10)^i \, p_{em}^{(1)}(00)^{s-i} + 1/2 C_s^{s/2} p_{em}^{(1)}(10)^{s/2} \, p_{em}^{(1)}(00)^{s/2}}{P_{accept}^{(2)}}, \tag{19}$$

$$p_{em}^{(1)}(11) = \frac{\sum_{i=s/2+1}^{s} C_s^i p_{em}^{(1)}(11)^i \, p_{em}^{(1)}(01)^{s-i} + 1/2 C_s^{s/2} p_{em}^{(1)}(11)^{s/2} \, p_{em}^{(1)}(01)^{s/2}}{P_{accept}^{(2)}}. \tag{20}$$

In a similar way, one can find BER's at third and next iterations.

In table V are presented the results of calculations BER's obtained by formulas (16)-(20) for different values of the parameter "*p*" and two iterations.

TABLE V. THE BER'S OBTAINED BY FORMULAS (16)-(20) GIVEN DIFFERENT PARAMETERS "p" AND TWO ITERATIONS

Primary probabilities			*IPIMC-1*		*IPIMC-2*	
p	p_m	p_e	p_m	p_e	p_m	p_e
0.05	0.095	0.05	$1.214*10^{-4}$	$8.355*10^{-5}$	$2.174*10^{-16}$	$1.566*10^{-9}$
0.1	0.18	0.1	$2.316*10^{-3}$	$1.6*10^{-3}$	$2.906*10^{-11}$	$5.874*10^{-7}$
0.15	0.255	0.15	$1.4*10^{-2}$	$9.415*10^{-3}$	$3.549*10^{-8}$	$2.155*10^{-5}$
0.2	0.32	0.2	$4.7*10^{-2}$	$3.3*10^{-2}$	$5.784*10^{-6}$	$2.993*10^{-4}$
0.25	0.375	0.25	$1.15*10^{-1}$	$8.2*10^{-2}$	$2,82*10^{-4}$	$2.448*10^{-3}$
0.3	0.42	0.3	$2.16*10^{-1}$	$1.59*10^{-1}$	$5.684*10^{-3}$	$1.5*10^{-2}$
0.35	0.455	0.35	$3.27*10^{-1}$	$2.5*10^{-1}$	$5.3*10^{-2}$	0.07
0.4	0.48	0.4	0.421	0.341	0.217	0.21
0.45	0.495	0.45	0.48	0.424	0.421	0.376
0.5	0.5	0.5	0.5	0.5	0.5	0.5

Table VI contains the results of simulation for BER's given different values of the parameters "*p*" and two iterations.

TABLE VI. THE RESULTS OF SIMULATION FOR BER'S GIVEN DIFFERENT PARAMETERS "p" AND TWO ITERATIONS

Primary probabilities			*IPIMC-1*		*IPIMC-2*	
p	p_m	p_e	p_m	p_e	p_m	p_e
0.05	0.095	0.05	$1.221*10^{-4}$	$8.492*10^{-5}$	-	-
0.1	0.18	0.1	$2.32*10^{-3}$	$1.599*10^{-3}$	-	$5*10^{-7}$
0.15	0.255	0.15	$1.354*10^{-2}$	$9.43*10^{-3}$	-	$2.1*10^{-5}$
0.2	0.32	0.2	$4.678*10^{-2}$	$3.291*10^{-2}$	$5.5*10^{-6}$	$2.875*10^{-4}$
0.25	0.375	0.25	$1.148*10^{-1}$	$8.217*10^{-2}$	$2,965*10^{-4}$	$2.45*10^{-3}$
0.3	0.42	0.3	$2.157*10^{-1}$	$1.587*10^{-1}$	$5.652*10^{-3}$	$1.479*10^{-2}$
0.35	0.455	0.35	$3.269*10^{-1}$	$2.501*10^{-1}$	$5.274*10^{-2}$	$6.902*10^{-2}$
0.4	0.48	0.4	0.42	0.341	0.217	0.2102
0.45	0.495	0.45	0.48	0.424	0.421	0.376
0.5	0.5	0.5	0.5	0.5	0.5	0.5

Comparing the data of Table V and VI, we can conclude that they have a good coincidence.

However, the eavesdropper's BER (p_e) should not be very small, even if BER for legitimate user is much smaller, because it results in unacceptable information leakage to eavesdropper. This means that p_e has to be increased artificially. That can be reached by application another protocol, which we previously called by protocol of *degradation of both channels* (main and eavesdropper) (DBC) after a completion of IPIMC protocols.

Subprotocol IDBC

DBC protocol can be realized by many ways, but the simplest method is to add modulo two adjacent bits of the output sequences after a completion of IPIMC protocol. Then we get:

$$\gamma_i = \gamma_{2i} \oplus \gamma_{2i+1},\ u_i = u_{2i} \oplus u_{2i+1},\ w_i = w_{2i} \oplus w_{2i+1}, \quad (21)$$

where γ_i, u_i, w_i, i=0,1,..l are output bits after a completion of IPIMC protocol by A, B and E respectively. Protocol DBC can be repeated iteratively v times as IDBC protocol. It is easy to conclude that BER after IDBC application occur:

$$\begin{aligned} p_m^{(v)} &= 2p_m^{(v-1)}(s)(1-p_m^{(v-1)}(s)) \\ p_e^{(v)} &= 2p_e^{(v-1)}(s)(1-p_e^{(v-1)}(s)) \end{aligned} \quad (22)$$

In Fig. 3. the BER probabilities for the main and eavesdropper channels given different number of iterations (v) for IDBC protocol and two iterations of IPIMC protocol against square root variances for additive artificial noise δ are presented.

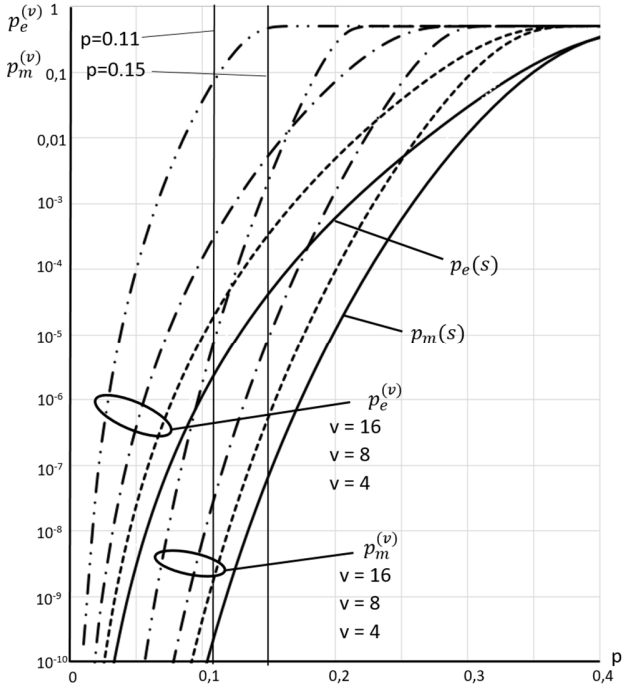

Fig.3. The BER probabilities for the main and the eavesdropper channels for 4, 8, 16 iterations of IDBC protocol and two iterations of IPIMC protocol against probability p

IV. Estimations of both security and reliability of the proposed KSP with application or not of error correcting codes and privacy amplification procedure

As we can see from Fig.3, there is a variety of protocol parameters, which allow us to provide acceptable divers of BER in the main and in the eavesdropper channels. Exact optimization of their choice is a problem for further investigations. But let us present some decision that provides a guaranteed acceptable level of both security and reliability.

Let us consider two scenarios: without error correcting code and with it. For the first scenario assuming $p = 0.11$ and two iterations of IPIMC protocol with s=4.

Thus, we get (see Fig.3) $p_m(s=4) = 1.52 \cdot 10^{-10}$, $p_e(s=4) = 1.36 \cdot 10^{-6}$. After applying eight iterations in IDMC protocol, we get $p_m^{(8)} = 3.9 \cdot 10^{-8}$ $p_e^{(8)} = 3.47 \cdot 10^{-4}$. Then, the probability P_{ed} of any error in the shared 256 key bits is $P_{ed} = 1-(1-p_m^{(8)}(s))^{256} = 9.98*10^{-6}$.

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

Let us suppose that the amount of information leakage to eavesdropper should be limited above by $I = 10^{-10}$ bit. (Note that the relationship between quantity I and the error probability for the optimal decoding by eavesdropper is determined by Fano inequality [11], presented in [4]). In order to ensure the value of I=10^{-10} bits it is necessary to execute so called *privacy amplification procedure* described in the paper [12] by U. Maurer. In [12] he has proven the theorem, which states that if it is applied string hashing to the final key string based on the universal$_2$ class$_2$ of random hash functions, then the amount of information at the output of such hash function is bounded from above as follows

$$I \le \frac{2^{-(k-l-t_c)}}{\ln(2)}, \quad (23)$$

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

$$t_c = k + k\,log((p_e^{(8)})^2 + (1-(p_e^{(8)}))^2) \quad (24)$$

is the Renyi information.

If we let the final key length be 256 bits, which is sufficient enough for cryptographic standards like AES, GOST, etc. [13], then the initial length of string k can be found from (23), (24) and given known value of $\tilde{p}_e^8$, it will be equal to $2.894*10^4$.

In order to estimate KSP efficiency it is necessary to calculate the key rate. If the error correction code is not used, then the rate KSP can be found as

$$R = \frac{l \cdot R_m^{(1)}(s) \cdot R_m^{(2)}(s) \cdot R_{DMC}}{k}, \quad (25)$$

where $R_m^{(1)} = \frac{1}{s}P_{ac}(1)$, $R_m^{(2)} = \frac{1}{s}P_{ac}(2)$ are transmission rates for first and second iteration PIMC protocol respectively and $P_{ac}(i)$ is the probability of the s-block acceptance by B in i-th iteration (i=1,2). 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 $R = 8.94 \cdot 10^{-8}$. This means that in order to share 256 key bits it is necessary to form $1.12 \cdot 10^7$ raw key bits transmitting over the ordinary Internet. At a speed of 10-100

Mbit/s as it is a common transmission rate over the Internet channel, the execution of KSP will take about 1-10 sec.

If you need to reduce the time of KSP execution then the error correcting code implementation can be used. But in such scenario we should take into account that eavesdropper is able to intercept all check sybomls transmitted 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 given in (23). We have to use so called *modified privacy amplificated procedure* (MPAP) discribed in [14]. It implies the use of the hash functions taken from the ununiversal$_2$ class and consequently in a "puncturing" of some bits (see [14] for details). After evaluation of the MPAP performance we get the following upper bound for information leakage

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

where all items in (26) are the same as in (23), except for *r,* that is the number of check bits of the chosen error correcting code, and additional factor 0.42. In the coding-based scenario it is more effective to select the parameter $p = 0.15$ and to choose the number of iterations of IDBC as v=16. Then we calculate that $p_m^{16}(s=4) = 2.32\cdot 10^{-3}$, $p_e^{16}(s=4) = 0.47$.

In order to decrease the probability of the incorrect block decoding P_{ed} the linear error correcting *(n,k,d)*-code is considered where n is a block length, k is the amount of information symbols that are needed for generating the final key length l=256 and d is the minimal code distance. 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 codes can be found by Varshamov-Gilbert bound [15]

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

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=539, k=431, r=108 and d=18 which provides $P_{ed} = 7.8\cdot 10^{-6}$. The bound (26) implies information leakage to eavesdropper $I = 4.94 \cdot 10^{-11}$ bits.

When error correction code is used, the key rate can be found as follows

$$R = \frac{l}{k/(R_m^{(1)}(s)\cdot R_m^{(2)}(s)\cdot R_{DMC}) + r}$$

For the parameters indicated above, we get $R = 1.674\cdot 10^{-7}$.

It means that in order to share 256 bit key it is nessary to form and transmit over the ordinary Internet channel $5.97\cdot 10^6$ raw key bits. Thus, if the transmittion rate is 10-100 Mbit/s it takes less then 0.6 s. 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 [4] how *lower density parity check codes (LDPC*) are useful for that case.

V. Additional problems of the proposed KSP

For proposed KSP (as well as it is true for any other similar protocols) it is required to perform *user authentication* procedure. Otherwise, an adversary be able to impersonate legitimate users when communicating with them over public channels and end up sharing by a common keys. There are different methods in order to provide user authentication for KSP. Thus, in the book [16] was proposed the simplest method for PGP system based on the known voice of the corresponding users where telephone can be used before activation of KSP. Then short hashes computed initially on shared keys are transmitted in a human voice, which, in turn, that can be recognized as known voice of the correspondent. But such approach is not very reliable in the face of attacks that imitate real voice of known persons by artificial methods. In order to avoid such popular attack as "man in the middle", the more sophisticated methods must be used. Creation of protected IPSec tunnels is one of possible approaches for which users exchange by random bits as well as code blocks needed for execution of subprotocols IPIMC and DBC. In this case tunnel security is provided by the use of short secret keys.

Another approach requires the use of a digital signature (DS) before performance of privacy amplification procedure of our protocol. Such DS has a particularity that it does not require to communicate sign message itself but only DS. It is worth to note that although in that case correspondents should possess secret keys in advance, they could be generated by legitimate users themselves and only public keys for DS verification should be sent to specialized certified center in order to ensure their authenticity. *Then such center be unable to impersonate legitimate user during a performance of KSP*.

Experimental verification of the proposed KSP (except of privacy amplification procedure) was carried out on the basis of special software, which in turn consists of two programs that allow forming key sequences in automatic mode. During program execution one legitimate user open sockets for control of all connections while another one creates corresponding connections. Matching of parameters also should be done initially and next exchange by packets follows. It is worth to note also that a connecting user is working in line with a part of KSP for the user A, whereas the expecting user is working with a part of KSP for the user B.

Software is written in language C# and it can run on any devices under operation systems Windows or Linux.

After starting the programs, we fix duplex channel by means of sockets. Rate of KSP is limited firstly by channel capacity with a linear dependence against this parameter until the maximum rate of random generator is reached, about 140 Mbit/s. for devises, which were taken for a testing (processor

11th Gen Intel Core i7-11700KF with core rate 4,9GGz at single-core mode).

It is assumed that the random numbers required for KSP to work as a part of System Security Cryptography Library. After a generation of the requested number of bits they are packed into packet size of 256 bytes (the sizes can vary) and another user is informed about it. Packet exchange is organized as one by one procedure that allows synchronizing users. Such strategy allows to process previous step without beginning of the next step processing. The resulting data blocks are used for a generation in the future the raw bit blocks.

The obtained raw key bit strings are used in the IPIMC subprotocol next. User *A* is sending 256 data bits (words of repetition code) in one pocket. Feedback from the user B is performed by sending of data block that carries the message about whether the corresponding block is accepted or not. The second iteration of the IPIMC protocol is realized as far as be accumulated the requested number of bits after a completion of the first iteration. After that is fulfilled protocol DBC by every user individually.

Software testing was performed both with the help of virtual machine and on real channel with the use of tunnels. In the case of using a virtual machine, the time taking to generate 1024 key bits is about 10 second, whereas for real communication network this time may be longer and depends on the speed of data transfer over chosen Internet.

VI. Conclusion

The current paper presents in our opinion the most effective method to share secret and reliable keys over such public and noiseless channels as the Internet under the condition that nothing cryptographic assumptions have been used for a providing of key security. In contrast to our previous paper, where was executed an exchange by real numbers over the Internet, our contemporary approach uses an exchange over the same channel by binary sequences. Such modification allows to reduce the value of channel traffic into 8-32 times and also to simplify signal processing both at the transmitting and receiving sides.

It is worth to note that theoretical estimates both security and reliability are fully justified by the results of simulations. Moreover, in the current version we have proved firstly that a processing of signal by eavesdropper cannot be improved (see Lemmas 1 and 2).

Problem of user's authentication, which, of course, is common to all KSP is discussed also in the current paper. This should be carefully studied in the future. We have touched slightly a problem of software design of our protocol that affects practical time requested for completion of full KSP.

In the future we hope to collaborate with some computer-oriented company in order to design the full pack of the requested programs. Thus, an opportunity could be taking to ensure a confidentiality for ordinary users of the Internet. So, our KSP is a real step ahead to a privacy of Civil Community.

References

[1] 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*, 2008, v. 54 n. 6), pp. 2535-2549.

[2] V. Starostin et al. "Key Generation Protocol Executing Through Non-reciprocal Fading Channel", *IJCSA Special Issue,* 2019,vol.16,N1,

[3] V.Korzhik et al."Secret Key Agreement over Multipass Channel Exploiting a Variable-Directional Antenna", *Int. Conf. of Advance Comp. Science and Application*s, 2012, vol.3,N 1,pp.172-178,.

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

[5] V. Yakovlev, V. Korzhik, M. Akhmetsina, A. Zhuvikin. "Key Sharing Protocol Using Exchange by Integers over Public Noiseless Channels Between Users that Provides Security executing without Cryptographic Assumption", *The 31th Conference of Open Innovations Association FRUCT*, Helsinki Finland, 27-29 April 2022, pp. 363-379.

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

[7] Dyakonov M.I. "Is Fault Tolerant Quantum Computation Really Possible? In Luryi S.,Xu J,Zaslavsky A. *Future Trends in Microelectronics*. John Wiley and Sons 2007, p.4-18.

[8] O Goldreich, S Goldwasser. "Public-Key Cryptosystems from lattice reduction". *Lecture Noted in Computer Science*, 1997. vol.1294, p.112-132.

[9] O.Logachev et al. *Boolean functions in coding theory and cryptography*, Moscow, 2004, (in Russian).

[10] John Proakis, *Digital Communications*, McGraw Hill, 1995.

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

[12] U.Maurer "Secret key agreement by public discussion from common information". *IEEE Trans.on Inf.Theory*, 1993, n 3, p.733-742.

[13] 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,.

[14] V. Korjik, G. Morales-Luna, and V. Balakirsky. "Privacy ampli cation theorem for noisy main channel", *Lecture Notes in Computer Science*, 2200 (2001), pp. 18-26.

[15] F,J.Mac Williams, N Sloane. *The Theory of Error Correcting Codes*" Bell Lab,1979.

[16] B.Schneier. *Applied Cryptography*, John Willey and Sons,1994.