Proceedings of the Federated Conference on Computer Science and Information Systems pp. 327–332 DOI: 10.15439/2019F70 ISSN 2300-5963 ACSIS, Vol. 18

Information theoretical secure key sharing protocol for noiseless public constant parameter channels without cryptographic assumptions

Valery Korzhik, Vladimir Starostin,
Muaed Kabardov, Aleksandr Gerasimovich,
Victor Yakovlev, Aleksey Zhuvikin
The Bonch-Bruevich Saint-Petersburg State
University of Telecommunications,
Saint-Petersburg, Russia.
Email: val-korzhik@yandex.ru, : star_vs_47@mail.ru

Guillermo Morales-Luna
Computer Science Department
CINVESTAV-IPIV,
Mexico City, Mexico.
gmorales@cs.cinvestav.mx

***Abstract*— We propose a new key sharing protocol executed through any constant parameter noiseless public channel (as Internet itself) without any cryptographic assumptions and protocol restrictions on SNR in the eavesdropper channels. This protocol is based on extraction by legitimate users of eigenvalues from randomly generated matrices. A similar protocol was proposed recently by G. Qin and Z. Ding. But we prove that, in fact, this protocol is insecure and we modify it to be both reliable and secure using artificial noise and privacy amplification procedure. Results of simulation prove these statements.**

***Index terms*: key sharing protocol, physical layer security, privacy amplification, Shannon information.**

I. Introduction

Solving the key sharing problem between legitimate users, connected by some telecommunication channels, has been in research focus within many years and it is still completely unsolved.

A protocol based on some cryptographic assumption (factoring problem, discrete log problem, error correction algorithm ctr. [1]) has been proposed by Diffie and Hellman [2] many years ago. There are known key distribution protocols based on "key commutative property" of the encryption algorithms [3]. But the corresponding protocol requires to hide the identity of the message sender [4], which is indeed a further cryptographic assumption.

It was developed in recent years a new approach to key distribution problem based on the notion of *physical layer security* (PHY) (see excellent survey [5]). This approach exploits some physical properties of real communication channels connecting legitimate users sharing a secret key in the presence of eavesdroppers. In line with this setting it was published a pioneer paper by A. Wyner [6] and its extension in the papers [7, 8], where legitimate channels were superior to eavesdropper ones on the SNR parameter.

Next, due to advanced Maurer's papers [9, 10], such approach was extended with the use of so-called *public discussion* and privacy amplification. It enables to transform disadvantage on SNR for legitimate users against eavesdroppers into advantage at the cost of exchange by additional information on public channels.

Other PHY-based protocols execute channels with random parameters (say, fading channels with multipath wave propagation) [9, 10, 11]. And this technique was used also in MIMO-based systems intended for a communication between mobile units [12, 13]. Effective key distribution problem can be solved also in frame of the so-called *quantum cryptography* where special quantum channels and devices [14] should be executed. But it is worth to note that all the key sharing methods mentioned above have been designed for known SNR in the eavesdropper channels or for the case where the number of antennas in the eavesdropper MIMO-based system is limited by some value. However such requirements to enemy system is obviously unrealistic.

Also there is a demand to share secret keys between users connected by constant (practically noiseless) channels (as Internet itself) and without any cryptographic assumption due to a risk of quantum computers to be applied in the future.

In section 2 we remind the key sharing protocol based on extraction of matrix eigenvalues described in [15] as Scheme EVSKey and confirm that it is in fact insecure [16]. Next, we extend this protocol in order to provide the upper bound for SNR in eavesdropper channel. In section 3 we present some channels transform primitives. Section 4 is devoted to results of simulation. In section 5 we optimize protocol parameters to provide both security and reliability of the shared key. Section 6 concludes the paper and proposes some open problem for further investigation.

II. Key sharing protocol based on extraction of matrix characteristic polynomials

Let us remind the scheme EVSKey [15] used in the current paper in order to generate the binary raw sequence for further creation of the shared key. The scenario corresponding to this scheme is presented in Fig. 1.

Before a transmission, Alice (A) and Bob (B) generate their own reference matrices $X_A, X_B \in \mathbb{C}^{n\times m}$ with independent matrix elements distributed according to $CN(0,\sigma_X^2)$ as well as random unitary matrices $G_A, G_B \in \mathbb{C}^{n\times n}$ where n is number of antennas employed by each user and m is the length of pilot signal.

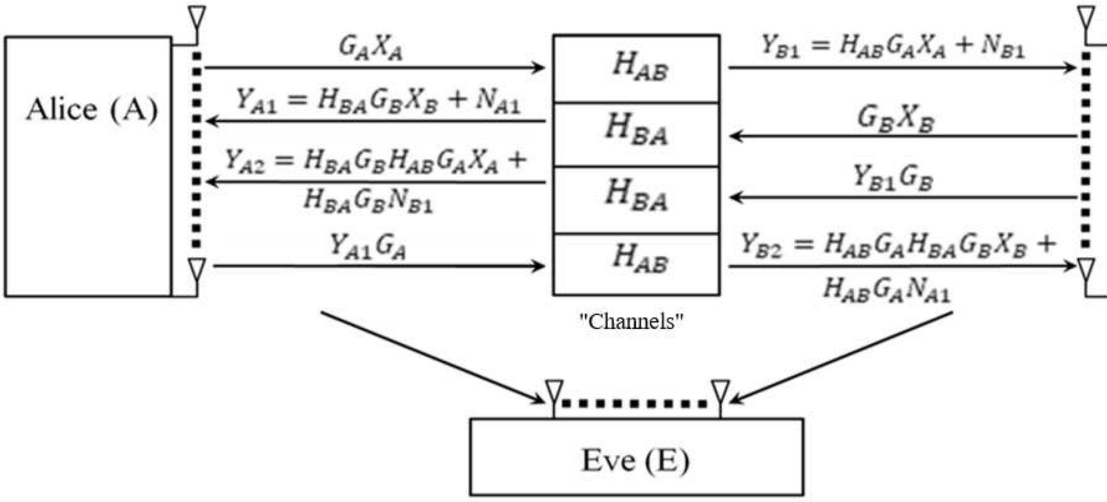
Alice (A)
Bob (B)
$G_A X_A$
$Y_{B1} = H_{AB} G_A X_A + N_{B1}$
$Y_{A1} = H_{BA} G_B X_B + N_{A1}$
$G_B X_B$
$Y_{A2} = H_{BA} G_B H_{AB} G_A X_A + H_{BA} G_B N_{B1}$
$Y_{B1} G_B$
$Y_{A1} G_A$
$Y_{B2} = H_{AB} G_A H_{BA} G_B X_B + H_{AB} G_A N_{A1}$
H_{AB}
H_{BA}
H_{BA}
H_{AB}
"Channels"
Eve (E)

Fig. 1. The scenario corresponding to Scheme EVSKey.

Since in our case A and B are connected by constant noiseless public channels, the original channel matrices H_{AB}, H_{BA} are generated by A and B as random matrices $(h_{ABij})_{ij}, (h_{BAij})_{ij} \sim CN(0, \sigma_W^2)$. N_{A1}, N_{B1} are AWGN matrices $(n_{A1ij})_{ij}, (n_{B1ij})_{ij} \sim CN(0, \sigma_e^2)$ generated by users A and B, respectively, as matrices of *artificially created Gaussian noises*. We let further $\sigma_W^2 = \sigma_X^2 = 1$, $\sigma_e^2 = \sigma^2$. Let us introduce the following matrices: $P = H_{BA} G_B$, $Q = H_{AB} G_A$. Then PQ and QP can be estimated by users via the least square method as:

$$PQ = Y_{A2}(X_A)^{-1} \tag{1}$$

$$QP = Y_{B2}(X_B)^{-1} \tag{2}$$

In [15] it was proved that matrices PQ and QP have the same non-zero eigenvalues. In [16], it has been proved an extension of such statement, that in fact they have the same *characteristic polynomials* (CP):

$$CP[PQ] = CP[QP] \tag{3}$$

Thus, from (3), we have that the legitimate users A and B are able to extract the same characteristic polynomials after a completion of protocol through noiseless channels although matrices PQ and QP can be different. The artificially added noises N_{A1}, N_{B1} result in errors between shared key bits extracted from quantized characteristic polynomial coefficients, eigenvalues or traces. Therefore, these errors have to be corrected by an additional procedure. Hence, the following question arises – what is the goal of adding artificial noises? The reason is a noising of eavesdropper channel in such a way that power of this noise cannot be decreased by any eavesdropper E!

But firstly we should demonstrate that E is able to intercept even noisy key bits because it was claimed in [15] that it is impossible. Unfortunately the last statement is wrong and in [16] there has been described the procedure about how E is able to intercept key bits, not necessary in the case when she has a close location to legitimate users. In fact, for noiseless channels, if E intercepts $Y_{A1}, Y_{A2}, Y_{B1}, Y_{B2}$, where $Y_{A1} = H_{BA} G_B X_B$, $Y_{A2} = H_{BA} G_B H_{AB} G_A X_A$, $Y_{B1} = H_{AB} G_A X_A$, $Y_{B2} = H_{AB} G_A H_{BA} G_B X_B$, she can compute the matrix Y:

$$Y = Y_{A2}(Y_{B1})^{-1} Y_{B2}(Y_{A1})^{-1} \tag{4}$$

(We note that pseudo-inverse matrices can be found by Penrose's procedure [17] as

$$(X_P)^{-1} = X^{\dagger}(XX^{\dagger})^{-1}). \tag{5}$$

Here "†" is conjugate transpose. It was proved in [16] that matrix Y is *similar* to matrix QP, thus they have the same characteristic polynomials for nonsingular matrices [18].

Hence the original scheme EVSKey is useless for key sharing but fortunately it can be used as a primary protocol providing lower noisy bound for eavesdropper that cannot be decreased because it is controlled by the legitimate users.

But before we present the following part of key sharing protocol, it is important to show that both artificial noises N_{A1}, N_{B1} should be added, otherwise eavesdropper can be able to intercept the legitimate key without any errors. Indeed, let us assume that only B creates artificial noise. Then we get:

$$Y_{B1} = QX_A + N_{B1},\ Y_{A2} = PY_{B1}$$

$$Y_{A1} = PX_B,\ Y_{B2} = QY_{A1} \tag{6}$$

Next, A extracts CP from the matrix:

$$Y_{A2}X_A^{-1} = PY_{B1}X_A^{-1} = PQ + PN_{B1}X_A^{-1}, \tag{7}$$

whereas B extracts the key from CP of the matrix:

$$Y_{B2}X_B^{-1} = QY_{A1}X_B^{-1} = QP \tag{8}$$

The eavesdropper E extracts the key from CP of the matrix:

$$Y_{A2}(Y_{B1})^{-1} Y_{B2}(Y_{A1})^{-1} = PQ \tag{9}$$

Thus (3) implies that E gets exactly the same key as legitimate user B. This means that such situation has to be excluded.

III. Description of channel transform primitives

In the following section there will be presented the results of simulation regarding the key bit errors under the provision of two artificial noises N_{A1}, N_{B1}. If such results give advantage to legitimate users against eavesdroppers, that is $P_l < P_e$, where P_l, P_e are the key *basic bit error rate* (BER) for legitimate users and eavesdropper, respectively, then we can apply privacy amplification theorem [11]. It states that such algorithm exists which provides an approaching to zero both key BER for legitimate users and Shannon information leaking to eavesdropper with the *key generation rate*:

$$R = h(P_e) - h(P_l), \tag{10}$$

where

$$h(x) = -(x\log_2 x + (1-x)\log_2(1-x))$$

is the entropy function. But, for opposite situation when it occurs that the key BER's satisfy to inequality $P_l > P_e$, it is necessary to apply in advance some additional protocol (primitive) that reduces the previous inequality to opposite one $(P_l < P_e)$.

In [11] several examples of such primitives are given. It seems that the best of them is protocol known as "*a preference improvement of the main channel*" (PIMC). Let us consider the protocol PIMC in more detail, when there are two binary statistically independent symmetric channels without memory (BSC: *binary symmetric channels*): one with BER P_l and another with BER P_e and $P_l > P_e$. Then legitimate user A has to repeat S times each bit transmitting over main channel with BER equal to P_l. Another legitimate user B receives only such S-blocks which consist of all zeros or ones and takes corresponding decision. He informs over public noiseless channel about blocks that he has accepted and erases other blocks. It is easy to see that such protocol forms the following BER for B:

$$\widetilde{P_l} = \frac{P_l^S}{P_l^S + (1-P_l)^S} \tag{11}$$

At the same time eavesdropper E intercepts S-blocks over BSC with BER P_e and controls public noiseless channels. E knows exactly which S-blocks are accepted by B. But because E's channel is statistically independent with the main channel (A→B), she should take decision about bits corresponding to S-block using *majority rule*. This means that she takes a decision that S-block carries bit "0", if this block has more zeros than ones and decision about bit "1", if the number of ones in that S-block is larger than the number of zeros. Then the BER after such decision will be for odd S the following:

$$\widetilde{P_e} = \sum_{i=\frac{S+1}{2}}^{S} \binom{S}{i} P_e^i (1-P_e)^{S-i} \tag{12}$$

But unfortunately, it seems to be impossible to repeat bits if they were extracted from CP's of the matrices PQ and QP!

In order to avoid this problem let us modify slightly our previous protocol as it is shown in Fig. 2. We can see that just after a generation of "raw" bits from matrices PQ and QP, user B generates truly random binary string γ that is XOR-ed with B's raw bits KB and it is transmitted over public and noiseless channel to user A that adds this string with her raw bits KA in order to get:

$$\widetilde{K_A} = K_B \oplus \gamma \oplus K_A = K_A \oplus \varepsilon_{AB} \oplus K_A \oplus \gamma = \gamma \oplus \varepsilon_{AB}, \tag{13}$$

where ε_{AB} is discrete noise string between raw key strings KA and KB. It is easy to see that in such setting the user B is able already to repeat S-times each bit of γ in order to perform the previous protocol. From now on we consider just γ as a new key string, transmitted to A over BSC with BER equal to P_l.

At the same time E, having received $K_B \oplus \gamma$ and her raw key K_e, extracted by (4), sums these sequences up. This gives:

$$\widetilde{K_e} = K_e \oplus \gamma \oplus K_B = K_B \oplus \varepsilon_{BE} \oplus \gamma \oplus K_B = \gamma \oplus \varepsilon_{BE}, \tag{14}$$

where ε_{BE} is discrete noise string between raw key strings K_B and K_e, that is equivalently to a transmission of key string γ to eavesdropper E over BSC with BER equal to $\widetilde{P_e}$.

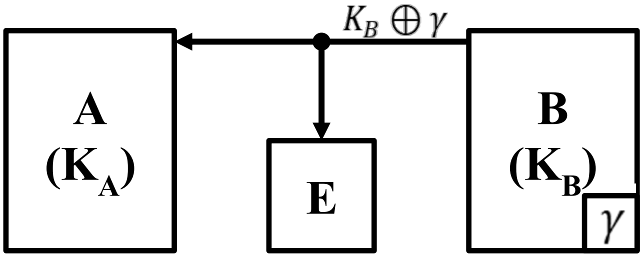

Fig. 2. Modified key sharing protocol.

IV. Results of Simulation

A. Using quantized matrix traces as the raw key bits

Since the traces of matrices are complex, they can be quantized both on amplitude and on phase. It was proved in [16] that the quantized intervals on amplitude of the traces providing equal probabilities of their occurrence should be chosen as follows:

$$r_{k-1} \le |Z| < r_k,\ k = 1, 2, \dots, N, \tag{15}$$

where Z is the trace of the matrices, $r_k = \sigma_Z \sqrt{-\ln(1-\frac{k}{N})}$, $\sigma_z^2 = n^2\sigma_w^2(\sigma_w^2 + \sigma_e^2) + n\sigma_e^2$, N is the number of intervals. In table 1 there are presented the results of BER simulation for N = 16, different values of NSR, and different matrix sizes $n \times m$. We see from this Table that for all parameters $P_l > P_e$ and hence it is necessary to execute the protocol PIMC (see section III) in order to reduce to opposite situation $P_l < P_e$, that will be demonstrated in the sequel.

B. Using quantized matrix eigenvalues as the raw key bits

Unfortunately, there appears one problem in this case – how to compare the numbering of eigenvalues adopted by different users? Let us denote by NP, NA the numbers of quantization intervals on phase and on amplitude respectively.

Table 1.
Simulation results of the BER for extraction them from matrices traces both legal users (P_l) and eavesdropper (P_e) with 8 sectors and 8 rings, uniform phase quantization and amplitude step quantization by (15)

n/m σ²	4x4	4x6	4x12	8x8	8x16	16x16
	P_l, P_e					
0.1	0.348 0.274	0.255 0.191	0.155 0.116	0.363 0.291	0.152 0.118	0.364 0.303
0.01	0.212 0.157	0.104 0.075	0.058 0.044	0.209 0.139	0.055 0.043	0.219 0.158
0.001	0.098 0.063	0.032 0.022	0.013 0.011	0.085 0.063	0.015 0.012	0.098 0.064

Let $N = N_P \times N_A$ be total number of quantization intervals. Then we find the number of eigenvalues that hits each of the N interval (cells). After a completion of eigenvalues extraction, we get a string of integers $g_1, g_2, \cdots, g_i, \cdots$, where g_i is the number of the *i*-th cell containing at least one eigenvalue. If several eigenvalues occur in the same cell, then the cell number is repeated as $g_i, g_i, \cdots$…. Next each number g_i is presented as a bit string and such strings are connected in a consecutive binary manner. The final binary string forms the raw shared key. It is easy to see that the total number of bits for each session of protocol can be computed as [16]

$$\log_2\binom{N+n-1}{n} = \log_2\frac{(N+n-1)(N+n-2)\ldots N}{n!} \tag{16}$$

In Table 2 there are presented the results of BER simulation for different matrix sizes and different NSR for eigenvalues extracted from matrices where each eigenvalue is quantized on 8 sectors and 8 rings.

TABLE 2.
SIMULATION RESULTS OF THE BER FOR EXTRACTION OF THEM FROM MATRIX EIGENVALUES BOTH LEGAL USERS (P_l) AND EAVESDROPPER (P_e,) WITH 8 SECTORS AND 8 RINGS FOR EACH EIGENVALUE

n/m σ²	4x4	4x6	4x12	8x8	8x16	16x16
	P_l, P_e					
0.1	0.348 0.288	0.262 0.204	0.170 0.121	0.350 0.302	0.207 0.159	0.207 0.159
0.01	0.215 0.156	0.115 0.080	0.069 0.049	0.235 0.175	0.085 0.057	0.085 0.057
0.001	0.104 0.068	0.037 0.027	0.022 0.014	0.127 0.082	0.029 0.021	0.029 0.021

We see from this Table also that, as before, for all BER parameters, $P_l > P_e$, hence it is necessary to execute the protocol PIMC in order to provide the opposite situation ($\tilde{P_l} < \tilde{P_e}$). We show in the sequel how to do it.

V. OPTIMIZATION OF KEY-SHARING PROTOCOL PARAMETERS IN ORDER TO PROVIDE GIVEN SECURITY AND RELIABILITY

It has been proved by the *Enhanced Privacy Amplification Theorem* [19], that the eavesdropper's expected Shannon information I_0 about the final key sequence shared by legitimate users, satisfies the inequality:

$$I_o \leq \frac{2^{-(k-t_c-l_0-r)}}{\alpha \ln 2}, \tag{17}$$

where k is the length of the string x generated by A and B after a completion of the protocol PIMC, t_c is the Renyi (or collision) information obtained by eavesdropper E about the string x received by E through a BSC with BER equal to $\tilde{P}_e$, r is the number of check bits sent by one of legitimate users to another one in order to reconcile their string, l_0 is the length of the final key, α is a coefficient that approaches to 0.42 for any fixed r, as k, r and $k-r$ are increasing (we recall that the *privacy amplification procedure*, providing the inequality (17), can be performed in two stages: firstly with the use of a hash function chosen randomly from universal$_2$ class and, secondly, by special *"puncturing" of hash string* [19]).

Let us consider a scenario, that allows to optimize parameters: k, r, S (see (11), (12)) for given prior values l_o, I_o and $\widetilde{P_{ed}}$ – the probability of incorrect decoding of final key string.

1. Given I_o, find the bound value

$$k - t_c - l_o - r = -\log_2(I_o\, \alpha \ln 2) = \lambda_1 \tag{18}$$

2. Calculate the value Renyi entropy [19]:

$$H_c = -\log_2\left(\tilde{P_e}^2 + (1-\tilde{P_e})^2\right) \tag{19}$$

3. Taking into account the relation

$$t_c = k - kH_c, \tag{20}$$

we get by (18)

$$kH_c - r = \lambda_1 + l_o. \tag{21}$$

4. In order to provide a decreasing of $\widetilde{P_{ed}}$ for bit string of length k and with execution of r check bits it is necessary to satisfy Shannon's inequality [21]:

$$\frac{k}{k+r} < C, \tag{22}$$

where

$$C = 1 + \tilde{P_l}\log_2\tilde{P_l} + (1-\tilde{P_l})\log_2(1-\tilde{P_l}) \tag{23}$$

5. Substituting (19) into (20) and considering (21) jointly with (22) (taken as equality) it is possible to solve the linear system of equations with respect to k and r.

We can take different values P_l, P_e from simulation results (see Tables 1, 2) and, by varying the parameter S into (11), (12), to obtain the new values $\tilde{P_l}$, $\tilde{P_e}$, that would improve our protocol. For example one could increase the length of final key l_o or to make it more secure by decreasing the value I_o. It is worth to note that we do not find so far a final key reliability in terms of the value $\widetilde{P_{ed}}$ but *we only guaranty* (due to Shannon's theorem) the existence of such encoding and decoding procedures that provide an approaching of this probability to zero.

Selection of the constructive encoding/decoding procedures requires further research. Seemingly, it should be of well known class of codes like *LDPC.* The later approaches *the Shannon limit* for large block lengths [20]. But before we face with some examples, it is necessary to fix the value I_0 by a reasonable manner. Let us present a lower bound for $\widetilde{P_{ed}}$ based on Fano's inequality [21]:

$$H(U/V) \leq h(\widetilde{P_{ed}}) + \widetilde{P_{ed}}\ \log_2(M-1), \tag{24}$$

where $H(U/V)$ is *conditional entropy* for eavesdropper E;

$$h(x) = -x\log_2 x - (1-x)\log_2(1-x), 0 \leq x \leq 1 \tag{25}$$

M is the number of possible keys (in our case it is equal to 2^{l_0}); $\widetilde{P_{ed}}$ is the probability of incorrect decoding that means a transition of the key string to another one (it is worth to note that the meaning of inequality (24) is the following: if entropy $H(U/V)$ is large, then the probability $\widetilde{P_{ed}}$ of incorrect decoding cannot be small). The graph of the function $\mu(\widetilde{P_{ed}}) = h(\widetilde{P_{ed}}) + \widetilde{P_{ed}}\ \log_2(M-1)$ is shown in Fig. 3.

We can see from Fig. 3 that if $H(U/V)$ is larger than some value, say H$_0$, then $\widetilde{P_{ed}}$ should be at least P^0_{ed} (see Fig. 3). Thus for given $M = 2^{l_0}$ and I_0, we can find the lower bound for $\widetilde{P_{ed}}$ and if it occurs very close to the value (M-1)/M (the probability of a *random key string guessing*) then it is assumed that the key sharing protocol is secure. If we let I_o = 10^{-3}, $M = 2^{64}$, then $H(U/V)$ = 64 – 0.001 = 63.999. Using the graph of $\mu(\widetilde{P_{ed}})$ we get that $\widetilde{P_{ed}}$ is sufficiently close to the case of random guessing (M-1)/M.

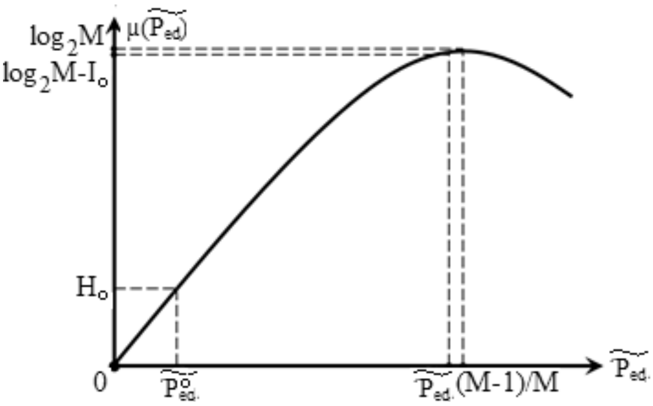

Fig. 3. Graph of function $\mu(\widetilde{P_{ed}})$ against $\widetilde{P_{ed}}$.

Examples:

1. Let us take from Table 2 the parameters $n \times m = 4 \times 4$, $\sigma^2 = 0.1$. Then $P_e = 0.29$, $P_l = 0.35$. Select S = 5 in (12), (13). Then we get by (12, 13) $\widetilde{P_l} = 0.043$, $\widetilde{P_e} = 0.15$, H_c = 0.425 and C = 0.439 by (20) and (24). Selecting λ_l = 10, l_0 = 64 and following to the scenario steps 1÷5, we get finally for key size 64 bit k = 1058, r = 374, $I_o \leq 2^{-10} \approx 10^{-3}$.

2. Let us take the same as in Example 1 initial parameters P_l, P_e and the same S = 5. But let us increase λ_l till 30. Then we get finally k = 1337, r = 472, $I_o \leq 2^{-30} \approx 10^{-9}$.

So we can see that it is possible to provide better security by changing protocol parameters. Because in this case the inequality (23) coincides with equality, it is necessary to decrease slightly the parameter k in order to provide approaching of P_{ed} to zero by Shannon theorem.

3. Let us increase the key size l_0 up to 128, because the most of contemporary encryption standards (like GOSI-2015 and AES) have namely such key sizes. We assume the same initial probabilities P_l, P_e as before and the same S = 5. Following to scenario steps 1÷5 we get the parameters: k = 2228, r = 787. $I_o \leq 2^{-30} \approx 10^{-9}$. We can see from this example that it is possible to share more longer key with a good security at the cost more longer error correcting code.

4. In this example we consider the case of key bit extraction from matrix traces (see Table 1).

Let us select the parameters $n \times m = 4 \times 4$, $\sigma^2 = 0.1$. Then we can see from Table 1 that $P_l = 0.348$, $P_e = 0.274$. Selecting parameter S = 7, we get by (12), (13) that $\widetilde{P_l}$ = 0.012, $\widetilde{P_e}$ = 0.095. Following to scenario steps 1÷5 we compute for l_0 = 128, that k = 962, r = 101, $I_o \approx 10^{-9}$.

We can see that having selected such protocol parameters $n \times m$ and NSR = σ^2, we can perform a tradeoff between security (I_o), reliability (P_{ed}) and error correction procedure complexity that is proportional to k and r.

In Fig. 4 there is presented a diagram of all procedures that must be executed in order to complete the key sharing protocol among legitimate users connected by noiseless, public and constant parameter communication channel. There is a new block (verification of key string authenticity) that has not been discussed before. In fact, this procedure is requested for any key sharing protocol in presence of an active adversary (eavesdropper). Otherwise the adversary can impersonate legitimate users and eventually share with them common key. It is common to use authentication method based on the so-called *short-key* [22]. The Needham-Schroder authentication protocol [23] can be used if users have initially distributed, by some trusted center, short keys.

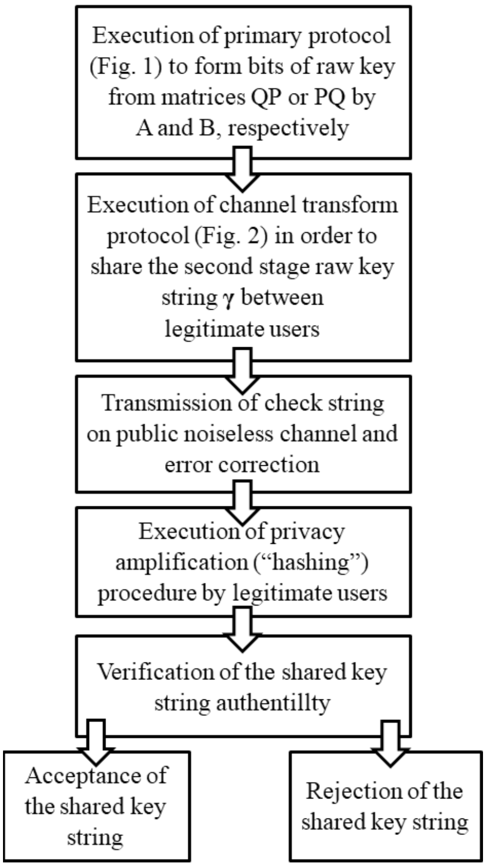
Execution of primary protocol (Fig. 1) to form bits of raw key from matrices QP or PQ by A and B, respectively
Execution of channel transform protocol (Fig. 2) in order to share the second stage raw key string γ between legitimate users
Transmission of check string on public noiseless channel and error correction
Execution of privacy amplification ("hashing") procedure by legitimate users
Verification of the shared key string authentillty
Acceptance of the shared key string
Rejection of the shared key string

Fig. 4. Diagram of the whole key sharing protocol.

Another way is if users can provide the so-called *paring procedure* during their "face to face" meeting (like Mag Pairing or Physical vibration [24, 25].

It is interesting to estimate (at least roughly) the length of the whole protocol (in transmitting channel bytes). Our computations show that the whole key sharing protocol requires about 100 Kbytes channel uses to produce 128 key bits.

VI. Conclusion

We have proposed key sharing protocol for noiseless public constant parameter communication channels (like Internet or “Direct seen”). The main novelty of our scenario is that *it is not based on some unrealistic assumptions* like given SNR, cryptographic assumption for eavesdropper (hard factoring problem) or multipath wave propagation, that is different for legitimate users and eavesdropper. The core of our protocol is the *Scheme EVSKey* proposed in [15]. But we proved that such protocol itself is insecure. Therefore we modified it by introducing artificial noise by legitimate users that does not allow to decrease this noise power by eavesdropper. Next we apply effective procedure of privacy amplification that provides both security and reliability for legitimate users. It is worth to note that good statistical properties of the final key string follow directly from such properties of truly random generated γ (see Fig. 2). It seems at a first glance that the paper [26] was devoted also to a solution of the same problem as our paper. In fact, it has only one common notion – “artificial noise”, but many differences, namely:

- we consider key sharing problem, instead of secure information transmission as in [26] ,
- in [26] it is executed either a MIMO system in fading channels or a set of “helpers”; our protocol is used in constant parameter public channel due to information exchange between two users,
- in [26] it is created noise in “zero-space”, whereas we execute special protocol imposing to eavesdropper artificial noise,
- in [26] it is provided zero noise by “zero-forcing”, but we provide a lower bound only for noise power,
- finally, in [26] it is guaranteed only some given *secrecy capacity*, but it is unknown how to realize it, namely how to provide constructive encoding/decoding procedures? But we on the contrary calculate Shannon information leakage to eavesdropper after application of the known privacy amplification procedure and find the parameters n and k for linear error correcting codes. Next investigations in the direction of artificial noise can be found in [27, 28].

The problems for further investigation are:

- consideration of constructive error correction procedures and
- elaboration of effective authentication algorithm against an active adversary.

References

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

[2] W. Diffie and M. E. Hellman, “New directions in cryptography,” vol. 22, no. 6, pp. 644–654, 1976.

[3] Schneier B., “Applied Cryptography”, JW Incorp., 1994.

[4] B. Alpern and F. B. Schneider, “Key exchange using ‘keyless cryptography’.” Inf. Process. Lett., vol. 16, no. 2, pp. 79–81, 1983. [Online].Available:http://dblp.unitrier.de/db/journals/ipl/ipl16.html#AlpernS83

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

[6] A. Wyner, “Wire-tap channel concept,” Bell System Technical Journal, vol. 54, pp. 1355–1387, 1975.

[7] I. Csiszár and J. Körner, “Broadcast channel with confidential messages.” IEEE Transactions on Information Theory, vol. 24, no. 2, pp. 339–348, 1978.

[8] V. Korjik and V. Yakovlev, “Non-asymptotic estimates for efficiency of code jamming in a wire-tap channel,” Problems of Information Transmission, vol. 17, pp. 223–22, 1981.

[9] L. H. Ozarow and A. D. Wyner, “Wire-tap channel II,” in Advances in Cryptology: Proceedings of EUROCRYPT 84, A Workshop on the Theory and Application of Cryptographic Techniques, Paris, France, April 9-11, 1984, Proceedings, 1984. doi: 10.1007/3-540-39757-4_5 pp. 33–50.[Online]. Available: https://doi.org/10.1007/3-540-39757-4_5

[10] V. Korjik and D. Kushnir, “Key sharing based on the wire-tap channel type ii concept with noisy main channel,” in Proc. Asiacrypt96. Springer Lecture Notes in Computer Science 1163, 1996, pp. 210–217.

[11] U. Maurer, “Secret key agreement by public discussion from common information.” IEEE Transactions on Information Theory, vol. 39, no. 3, pp.733–742, 1993.

[12] V. Yakovlev, V. I. 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, vol. 54, no. 6, pp. 2535–2549, 2008.

[13] V. Korjik and M. Bakin, “Information-theoretically secure keyless authentication,” in Proc. IEEE Symp. on IT’2000. IEEE, 2000, p. 20.

[14] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin. “Experimental quantum cryptography”, J. Cryptol., vol. 5, no. 1, pp. 3–28, Jan. 1992. [Online]. Available: http://dl.acm.org/citation.cfm?id=146395.146396

[15] D. Qin and Z. Ding, “Exploiting multi-antenna non-reciprocal channels for shared secret key generation,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 12, pp. 2693–2705, Dec 2016. doi: 10.1109/TIFS.2016.2594143

[16] Starostin V.S. et al “Key Generation protocol executing through non-reciprocal fading channels”, Intern. Journal of Computer Science and Applications, vol. 16, no. 1, pp. 1–16, 2019.

[17] Ben-Israel, Adi; Greville, Thomas N.E. , p. 7. Generalized inverses: theory and applications (2nded.). NY: Springer. ISBN0-387-00293-6, 2003.

[18] Home and Johnson, “Matrix Analysis”, Cambr. Univ.Pres. 1985.

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

[20] K. Shalkoska, Implementation of LDPC Algorithm: In C Programming Language. LAP LAMBERT Academic Publishing, 2017. ISBN9783330026049. [Online]. Available: https://books.google.com.mx/books?id=1yNcMQAACAAJ

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

[22] D. Dasgupta, A. Roy, and A. Nag, Advances in User Authentication, 1st ed. Springer Publishing Company, Incorporated, 2017. ISBN 3319588060,9783319588063

[23] R.M. Needham and M.D. Schroeder, “Using Encryption for authentication in Large Network of computers”. ACM, v21, p.993-999, 1978.

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

[25] Roy N. et al, “Faster Communication through Physical vibration”, proc USENIX Symp. Netw. Syst. Design, p. 671-675, 2016.

[26] Goel S. and Negi R., “Guaranteeing Secrecy using Artificial Noise”, IEEE Trans. of Wireless Communication, vol. 7, no 6, p. 2180-189, 2008.

[27] Bangwon Seo, “Artificial Noise Based Secure Transmission Scheme in Multiple Antenna Systems”, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 21 (2016)

[28] Liu, S., Hong, Y., & Viterbo, E. Artificial noise revisited. *IEEE Transactions on Information Theory*, *61*(7), 3901 - 3911. https://doi.org/10.1109/TIT.2015.