A new code-based public-key cryptosystem resistant to quantum computer attacks

Elena Egorova¹, G Kabatiansky¹, E Krouk² and C Tavernier ³

¹ Skoltech, Moscow, Russia
This research was supported by RFBR grants 16-01-00716 and 18-07-01427
² Higher School of Economics, MIEM, Moscow, Russia
This research was supported by RFBR grant 16-01-00716
³ ASSYSTEM, France
E-mail: egorovahelene@gmail.com; g.kabatyansky@skoltech.ru; ekrouk@hse.ru;
tavernier.cedric@gmail.com

Abstract. We propose a new type of public-key cryptosystems (PKC) which is based on repetition of different error-correcting codes. We give a brief analysis of some well known attacks on code-based PKC, including structural ones and show that the scheme could be used as a perspective post-quantum PKC.

1. Introduction
It is well known that with appearance of enough large and stable quantum computer main known public-key cryptosystems (PKC) will be broken by Shor’s algorithm [1] and network communications (as well as some other activities) become insecure. PKC which are resistant against known attacks via hypothetical quantum computer are called post-quantum PKC. Among few directions of construction post-quantum PKC like lattice-based cryptography, hash-based cryptography the code-based cryptography is the oldest one and the most elaborated.

First PKC based on linear error-correcting codes was proposed in [2] forty years ago. Since then not many principal improvements were done. Among them a PKC, which is based on syndrome decoding of linear codes, was suggested in [3]. After that many different families of error-correcting codes were investigated as a candidate to replace Goppa codes used in [2]. In this paper we propose a new PKC based on repetition of different error-correcting codes what is a generalization of the construction of [4]. We show that our scheme is resistant to structural attacks of [5], [6] proposed for breaking [4] and we analyse also different types of general decoding attack in our case. We also show how the new PKC can be used for digital signature purpose. Importance of our scheme as well as some other code-based schemes is that there are no known effective attacks on these scheme based on a quantum computer similar to Shor’s attack [1] on the famous RSA cryptosystem [7].

2. Main known code-based PKC
Let us start from recalling some basic facts about McEliece scheme [2] and its modifications. McEliece PKC works in the following way - a user \( A \) chooses a generator \( k \times n \) matrix \( G_A \) of
some linear \((n, k)\)-code \(C_A\), which has decoding algorithm \(\Phi\) correcting \(t\) errors with polynomial in \(n\) complexity. This matrix is a secret, known only to \(A\). The user \(A\) also chooses randomly two matrices: \(k \times k\) nonsingular matrix \(S_A\) and \(n \times n\) permutation matrix \(P_A\) and then construct public matrix \(G_{pub} = S_AG_A P_A\), which will be known to all other users. In order to deliver a message \(m\) of length \(k\) to the user \(A\) any other user transmits to \(A\) the following vector of length \(n\)

\[
y = mG_{pub} + e,
\]

where \(e\) is a vector of weight \(t\) which is randomly generated by \(B\). The user \(A\) after receiving vector \(y\) calculates

\[
y' = yP_A^{-1} = mG_{pub}P_A^{-1} + eP_A^{-1} = (mS)G_A + e',
\]

where \(e' = eP_A^{-1}\) and \(wt(e') = wt(e) \leq t\) since \(P\) is a permutation. Then \(A\) applies the decoding algorithm \(\Psi\) to vector \(y' = m'G + e'\) in order to obtain vector \(m' = mS\) and finally \(A\) receives \(m := m'S^{-1}\). Any other user will deal either with the problem of correcting \(t\) errors by looking like arbitrary linear code or with the problem of reconstructing code structure from its public-key matrix (its called structural attack). In the original paper [2] irreducible Goppa codes [8] were chosen as the family of codes for the scheme. In particular, it was suggested to use Goppa code of length \(n = 1024\) with \(k = 524\) information bits (code dimension) and the minimal code distance \(d = 101\).

Later H. Niederreiter proposed a scheme, which is in some sense dual to McEliece scheme, [3]. Saying informally, the user \(A\) does the same manipulations but with a parity-check matrix \(H_A\) of an \((n, k)\)-code \(C_A\). Namely, the matrix \(H_{pub} = S_AH_A P_A\), where \(S_A\) is randomly chosen \((n-k) \times (n-k)\) nonsingular matrix, is a public-key matrix. The set of messages is the set of all binary vectors \(e\) of length \(n\) such that \(wt(e) = t\) and the corresponding ciphertext defined as \(s = H_{pub}e^T\). These two schemes may have different parameters but they have equivalent security, i.e., if one PKC can be broken than the other also, see [9] or just a few lines prove of this fact in [4]. Because of similarity and equivalent security of these two schemes we consider them below as a single scheme. Note that in [10] it was proposed how to send more information via McEliece scheme by choosing the error-vector \(e\) not random but as carrying some information bits while maintaining the same security.

3. Repetition of different codes and a new code-based PKC

Recall that for any error-correcting code \(C\) of length \(n\), cardinality \(M\) and with the minimal code distance \(d\) its \(L\)-repetition is the code of length \(nL\), cardinality \(M\) and with the minimal code distance \(dL\) which consists of the following codewords \(v^L = (v|v|...|v)\), where \(||\) denotes words concatenation.

Sidelnikov proposed how to use this construction in order to build PKC. Namely, he used \(u\)-times repetition of binary Reed-Muller codes \(RM(s, m)\) of rather small order \(s\), where for different copies of RM-codes different generator (or parity-check) matrices are used. Recall that \(RM(s, m)\) of length \(n = 2^m\) has the number of information bits equals to \(k = \sum_{i=0}^{s} C_i^m\) and distance \(d = 2^{m-s}\). The corresponding public generator \(k \times un\) matrix of the Sidelnikov scheme has form \(G_{pub} = FP\), where \(P\) is random \(nu \times nu\) permutation matrix and \(F\) is \(k \times un\) matrix obtained by concatenation of matrices \(F_1, ..., F_u\), where the parameter \(u\) is a positive integer. Each matrix \(F_i\) has form \(F_i = S_iG_{RM}\), where \(G_{RM}\) is the ordinary generator matrix of \(RM(r, m)\) code and matrices \(S_i\) are random \(k \times k\) nonsingular matrices chosen independently. Since matrices \(S_i\) are not equal, as it should be for McEliece scheme, the Sidelnikov scheme is (slightly) different from the classical McEliece PKC. Now we generalize the repetition construction to the
Consider two linear codes $V_1$ and $V_2$ of the same dimension $k$ but different lengths $n_1$ and $n_2$ and let $G_1$ and $G_2$ be some generator matrices of these codes. Define a $k \times n$ generator matrix $G_{1,2} = G_1||G_2$ of a new code, which we call pseudo-repetition ("pseudo" because codes are different), as the concatenation of matrices $G_1$ and $G_2$, hence $n = n_1 + n_2$. Denote this code as $V_1 \bigoplus V_2$. Surely this construction can be defined for any multiplicity $u$ of pseudo-repetition, in particular, when $V_1 = V_2 = \ldots = V_u = RM(s, m)$ we have the Sidelnikov scheme.

In order to create a code-based PKC one needs to define which and how errors will be corrected. Let $\psi_1$ and $\psi_2$ be decoding algorithms of codes $V_1$ and $V_2$ capable to correct $t_1$ and $t_2$ errors correspondingly. Then define the following decoding algorithm for code $V_1 \bigoplus V_2$ capable to correct up to $t = t_1 + t_2$. Let $e_1$ be the error vector on first $n_1$ positions and let $e_2$ be the error vector on second (last) $n_2$ positions. Then $wt(e_1) \leq t_1$ or $wt(e_2) \leq t_2$ (or both) since $wt(e_1 + e_2) \leq t_1 + t_2$ and hence we can decode the corresponding code and find either the corresponding codeword (or the error-vector if one deals with Niederreiter PKC). It may happen that the other part (with more errors than the corresponding $t_i$) also can be decoded but incorrectly. Anyway we recover codevectors in both cases and fortunately for the right choice corresponding distance will be at most $t$ and for the wrong one with the probability very close to 1 the corresponding total distance will be larger than $t$.

Note that the sender (Bob) cannot create (unfortunately) an error vector in such a way that both $wt(e_1) \leq t_1$ and $wt(e_2) \leq t_2$ since separation of the coordinate set on subsets corresponding two codes is unknown (keep secret).

Now let us consider a "toy example" to illustrate our idea. Let $V_1$ be $RM(4,10)$ code. It has length $n_1 = 1024$, dimension $k = 386$ and $d_1 = 64$. There are different decoding algorithms capable to decode up to $t_1 = 31$ errors (so-called bounded distance decoding), moreover there is a list decoding algorithm capable to decode up to the minimal code distance, see [11], in particular, capable to decode up to $t_1 = 61$ with rather small (polynomial) complexity. The size of the corresponding list is small and to find the codeword at the minimal distance from the received vector is rather easy, see [11] for details. Let $V_2$ be Goppa code of length $n_2 = 512$, the same dimension $k = 386$ and the constructive distance $d_2 = 29$, see [8]. Hence the standard decoding algorithm of Goppa codes can correct up to $t_2 = 14$ errors. Let us consider two decoding scenarios. In both scenarios we decode Goppa code with correction 14 errors (or less). Difference is only by decoding of the RM code. In the first scenario we decode RM code by bounded distance decoding and it outputs at most one codeword. Then $t = 31 + 14 = 45$ and the sender should generate this errors randomly on the total length $n = 1024 + 512 = 1536$. Therefore the expected number of errors on the positions of RM code will be $45 \times 2/3 = 30$ and the expected number of errors on the positions of Goppa code will be 15, what is very close to the decoding capacity of both algorithms. Hence in this scenario we should try to decode both codes.

In the second case $t = 61 + 14 = 75$. Therefore the expected number of errors on the positions of RM code will be $75 \times 2/3 = 50$ and the expected number of errors on the positions of Goppa code will be 25, what is significantly bigger than the decoding capability of the corresponding algorithm. So in this scenario the receiver (Alice) first will try to decode (by list decoding algorithm) $RM(4,10)$ code, and only in the very rare case when the number of errors on positions of RM code is larger than 61 and its decoding failed then decode Goppa code. Surely the encryption rate is bigger for the second scenario.
4. Analysis of known attacks

The main idea of the new scheme is to hide which positions are positions of RM code and which are - of Goppa code. Let us first consider structural attacks on Sidelnikov scheme proposed in [5, 6]. Note that these attacks work well even for Sidelnikov PKC only in the case of $u = 1$, i.e., without repetition and hence positions of RM code are known.

Now consider the example of the previous section. Both attacks based on the search of minimal codevectors in the dual code. Note that for code $V_1 \biguplus V_2$ its dual code contains codes $V_1 \biguplus ||V_2||$, where $||$ denotes concatenation. The dual to RM(4,10) is RM(5,10) and hence the minimal weight is 32. It is known that for binary Reed-Muller code $RM(s,m)$ the number $A_d(s,m)$ of codevectors of weight $d = 2^{m-s}$ equals to

$$A_d(s,m) = 2^s \prod_{i=0}^{m-s-1} \frac{2^{m-i} - 1}{2^{m-s-i} - 1}$$

Hence for the considered case the number $A_{32}$ of codevectors of weight 32 in $RM(5,10)$ is surely less than $2^5 \prod_{i=0}^{4} 2^{9} = 2^{35}$. On the other hand, the total number of binary vectors of weight 32 and length 1536 is

$$\binom{1536}{32} > (3 \times 2^9)^{32}/32^{32} > 2^{48+288-160} = 2^{176}$$

Hence the probability that a randomly chosen vector of weight 32 belongs to the dual to RM(4,10) code is negligibly small, i.e., less than $2^{-176}$. Hence this attempt to guess a minimal vector from the dual to RM(4,10)-code failed.

Consider more sophisticated attack. Let $H$ be $1150 \times 1536$ pubic parity-check matrix (straightforward calculated from $G_{pub}$). One may think about a desirable minimal vector $v$ of weight 32 (such that $Hv^T = 0$) as the sum of two vectors of weight 16, i.e., $v = v_L + v_R$ and $Hv_L^T = Hv_R^T$. Let us evaluate the list $L$ of syndroms for all vectors of length 1536 and weight 16 and then arrange this list, which size is $L = \binom{1536}{16} > (3 \times 2^9)^{16}/16^{16} > 2^{24+144-64} = 2^{104}$, in lexicographical order what takes at least $1150 \times |L| > 2^{114}$ operations. Then by merging the list with its copy one can find all minimal codevectors of weight 32, but the number of operations is infeasible and a structural attacks of [5, 6] failed.

Surprisingly the direct attack on the new system is more efficient. Namely, consider the information set decoding algorithm [12] which was proposed as an attack in the original McEliece paper and later on further developed in many papers. In order to decode an $(1536, 386)$-code with correction up to $t = 75$ errors one just chooses randomly 386 positions assuming that they are free from errors with the probability $p = \binom{1536}{1531}/1531^{75} \approx 2^{-31}$ and then recover the corresponding codevector and check if this vector has (Hamming) distance from the decoded one not more than $t$. Recovery of a codevector takes $386 \times 1150 \approx 4 \times 10^5$ binary operations, so the total average number of binary operations is $\approx 8 \times 10^{14}$ what is not sufficiently large to consider this example enough secure. To make it resistant to decoding attack one needs to increase code parameters (length, dimension, distance) in a few times.

5. Conclusion

We proposed a new public-key cryptosystem based on (pseudo)repetition of different error-correcting codes (but of the same dimension) which is a generalization of Sidelnikov PKC [4]. Even for rather small values of public key length this system is resistant against structural attacks developed in [5], [6] for breaking Sidelnikov PKC. On the other hand, the “toy example” of the system can be broken via well-known decoding algorithms of arbitrary linear codes, see [12]
and its improvement in [13],[14]. To avoid this attack one needs to consider longer codes with bigger error-correction capability or even the same code but to increase its decoding radius by using more complicated (but still of polynomial complexity) decoding algorithms of [16]. Then it becomes possible to correct up to 100 errors just by RM-code (but with small probability of failure) plus 14 errors by Goppa code. Hence even the “toy” example code can correct totally up to 114 errors and therefore the complexity of the corresponding information set decoding is of order $10^{27}$ what looks much more reliable than $10^{15}$.

Known results shows that currently quantum algorithms speed up decoding of linear codes insignificantly, for instance, the complexity of May-Meurer-Thomae ISD algorithm drops down from $2^{0.05904n+o(n)}$ to $2^{0.05806n+o(n)}$ [17]. Therefore the proposed scheme can be considered as a candidate for post-quantum cryptography. Moreover, one of the main advantage of the new PKC is that it based on decoding algorithms which capable to correct errors beyond so-called bounded decoding, i.e., beyond half of the code minimal distance. Note that almost all previous code-based PKC use only bounded distance decoding but refer to hardness of decoding linear codes what is not correct since NP-hardness is proved only for maximum-likelihood decoding (i.e., minimum distance decoding), see [18],[19]. We use decoding algorithms which are much closer maximum-likelihood decoding what gives an additional reason that the new PKC should be resistant to quantum computer attacks.

One of the direction of further research is to find another, better codes and make the corresponding cryptoanalysis more solid. Another direction of further research is to apply the new scheme of codes pseudo-repetition to constructing code-based digital signature in style of [20] or [21].

References
[1] Shor P W 1997 SIAM Journal on Computing 26 1484-509
[2] McEliece R J 1978 Jet Propulsion Lab 114-6
[3] Niederreiter H 1986 Problems of Control and Information Theory 15 159-66
[4] Sidelnikov V M 1994 Discrete Mathematics 6 3-20
[5] Minder L and Shokrollahi 2007 LNCS 4515 347-60
[6] Borodin M A and Chizhov I V 2014 Discrete Math. Appl. 4 191–207
[7] Rivest R, Shamir A and Adleman L 1978 Communications of the ACM 21 120-6
[8] Goppa V D 1970 Problems of Information Transmission 6 207–12
[9] Li X Y, Deng R H and Wang X M 1994 IEEE Transactions on Information Theory 40 271–3
[10] Biswas B and Sendrier N 2008 PQCrypto 2008 47–62
[11] Dumer I, Kabatiansky G and Tavernier C 2008 ACCT 2008 82–5
[12] Prange E 1962 IRE Transactions on Information Theory 8 5–9
[13] Kruk E 1989 Problems of Information Transmission 25 251–4
[14] Barg A, Krouk E and Tilborg H C A 1999 IEEE Transactions on Information Theory 45 1392–1405
[15] Dumer I 1991 Proceedings of the 5th Joint Soviet-Swedish International Workshop on Information Theory 50–52
[16] Dumer I 2004 IEEE Transactions on Information Theory 50 811–23
[17] Kirshanova E 2018 Lecture Notes in Computer Science 10786 507–27
[18] Berlekamp E R, McEliece R J and van Tilborg H C A 1978 IEEE Transactions on Information Theory 24 384–6
[19] Guruswami V and Vardy A 2005 IEEE Transactions on Information Theory 51 2249 – 56
[20] Kabatianskii G, Krouk E and Smeets B 1999 Lecture Notes in Computer Science 1355 161-77
[21] Courtois N T, Finiasz M and Sendrier N 2001 Lecture Notes in Computer Science 2248 157–74