8. *Semenov A., Otpuschennikov I., Gribanova I., et al.* Translation of algorithmic descriptions of discrete functions to SAT with application to cryptanalysis problems // Log. Methods Comput. Sci. 2020. V. 16. Iss. 1. P. 29:1–29:42.
9. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986.
10. *Dowling W. F. and Gallier J. H.* Linear-time algorithms for testing the satisfiability of propositional horn formulae // J. Log. Program. 1984. No. 1(3). P. 267–284.
11. *Biere A.* The AIGER And-Inverter Graph (AIG) format version 20071012. Tech. Report 07/1. Institute for Formal Models and Verification, Johannes Kepler University. 2007.
12. *Kipnis A. and Shamir A.* Cryptanalysis of the HFE public key cryptosystem by relinearization // LNCS. 1999. V. 1666. P. 19–30.
13. *Семёнов А. А., Антонов К. В., Отпущенников И. В.* Поиск линеаризующих множеств в алгебраическом криптоанализе как задача псевдобулевой оптимизации // Прикладная дискретная математика. Приложение. 2019. № 12. С. 130–134.
14. *Антонов К.В., Семёнов А.А.* Применение SAT-оракулов для генерации дополнительных линейных ограничений в задачах криптоанализа некоторых легковесных шифров. Прикладная дискретная математика. Приложение. 2020. № 13. С. 114–119.
15. *Грибанова И. А., Семёнов А. А.* Об аргументации отсутствия свойств случайного оракула у некоторых криптографических хеш-функций // Прикладная дискретная математика. Приложение. 2019. № 12. С. 95–98.
16. ЦКП Иркутский суперкомпьютерный центр СО РАН. `http://hpc.icc.ru`.
17. *Beaulieu R., Shors D., Smith J., et al.* The Simon and Speck lightweight block ciphers // Proc. 52nd Ann. Design Automation Conf. New York, USA, 2015. P. 175:1–175:6.
18. *Антонов К. В., Семёнов А. А.* Применение метаэвристических алгоритмов псевдобулевой оптимизации к поиску линеаризующих множеств в криптоанализе криптографических генераторов // Материалы 6-й Междунар. школы-семинара «Синтаксис и семантика логических систем». Иркутск: ИГУ, 2019. С. 13–18.
19. *Gribanova I. and Semenov A.* Using automatic generation of relaxation constraints to improve the preimage attack on 39-step MD4 // Proc. 41st Intern. Convention Inform. Commun. Technol. Electr. Microelectr. (MIPRO). IEEE, 2018. P. 1174–1179.
20. *Gribanova I. and Semenov A.* Parallel guess-and-determine preimage attack with realistic complexity estimation for MD4-40 cryptographic hash function // Материалы конф. «Параллельные вычислительные технологии (ПаВТ) 2019» (Калининград, 2–4 апреля 2019). С. 8–18.

УДК 621.391.7 DOI 10.17223/2226308X/14/24

CHOOSING PARAMETERS FOR ONE IND-CCA2 SECURE McEliece MODIFICATION IN THE STANDARD MODEL

Y. V. Kosolapov, O. Y. Turchenko

The paper is devoted to choosing parameters for one IND-CCA2-secure McEliece modification in the standard model. In particular, the underlying code, plaintext length and one-time strong signature scheme are suggested. The choice of parameters for the scheme was based on efficiency, on the one hand, and security, on the other. Also, experiments for the suggested parameters are provided using the NIST statistical test suite.

Keywords: *post-quantum cryptography, McEliece-type cryptosystem,* IND-CCA2-*security, NIST statistical test suite.*

1. Introduction

The development of post-quantum cryptosystems resistant to adaptive chosen ciphertext attacks (IND-CCA2 secure cryptosystems) is currently relevant. In particular, NIST hold competitions for the formation of post-quantum cryptography standards [1]. One of the most successful candidates [2] is based on the idea of random oracle. However, since random oracle is only theoretical function, then the construction of IND-CCA2 secure post-quantum cryptosystems without random oracles (standard model) is also an interesting task. One of the ways to construct such scheme is to modify McEliece cryptosystem [3]. For instance, in [4–6] authors modified McEliece cryptosystem using correlated products method [7]. This paper is devoted to choosing practical parameters for cryptosystem from [5].

2. Cryptosystem from [5]

Let n, t be natural, $[n] = \{1, \ldots, n\}$, $\beta \subseteq [n]$, $2^{[n]}$ is the set of all subsets of $[n]$, $\mathbb{F}_2$ be a Galois field of cardinality 2. The support of the vector $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{F}_2^n$ is the set $\mathrm{supp}(\mathbf{v}) = \{i : v_i \neq 0\}$ and the Hamming weight of this vector is a number $\mathrm{wt}(\mathbf{v}) = |\mathrm{supp}(\mathbf{v})|$. If S is a finite set, then $s \in_{\mathrm{R}} S$ denotes the operation of picking an element at random and uniformly from S. Denote by $\mathcal{E}_{n,t,\beta}$ the subset of $\mathbb{F}_2^n$ such that any vector $\mathbf{e} = (e_1, \ldots, e_n) \in \mathcal{E}_{n,t,\beta}$ has Hamming weight t and $e_i = 0$ for any $i \in \beta$. We will write $\mathcal{E}_{n,t}$ when $\beta = \varnothing$. For the vector $\mathbf{v} \in \mathbb{F}_2^k$ and the ordered set $\omega = \{\omega_1, \ldots, \omega_l\} \subseteq [k]$, where $\omega_1 < \ldots < \omega_l$, we consider the projection operator $\Pi_\omega : \mathbb{F}_2^k \to \mathbb{F}_2^{|\omega|}$ acting according to the rule: $\Pi_\omega(\mathbf{v}) = (v_{\omega_1}, \ldots, v_{i_{\omega_l}})$. For ω, consider a subset $\mathcal{G}(\omega)$ of symmetric group $\mathcal{S}_k$ acting on the elements of the set $[k]$:

$$\mathcal{G}(\omega) = \{\pi \in \mathcal{S}_k : \pi(1) = \omega_1, \ldots, \pi(l) = \omega_l\}.$$

With every permutation π from $\mathcal{G}(\omega)$ we associate a permutation $(k \times k)$-matrix R_π.

Now we consider construction from [5]. Recall that a public key cryptosystem is a triplet of algorithms, i.e., $\Sigma = (\mathcal{K}, \mathcal{E}, \mathcal{D})$, where $\mathcal{K}$ is a generation algorithm, $\mathcal{E}$ is an encryption algorithm, $\mathcal{D}$ is a decryption algorithm. We will write $\{\mathbf{m}\}_{pk}^{\Sigma}$ as encryption of the message $\mathbf{m}$ with the key pk and $\{\mathbf{c}\}_{sk}^{\Sigma}$ as decryption of the ciphertext $\mathbf{c}$ on the secret key sk. For McEliece cryptosystem, we denote such triplet Σ as McE.

In the cryptosystem Σ [5], key generation algorithm $\mathcal{K}_\Sigma$ takes as input two security parameters $N, s \in \mathbb{N}$ and outputs a public-key pk and a secret key sk of the form

$$pk = ((pk_i^0, pk_i^1))_{i=1}^s, \quad sk = ((sk_i^0, sk_i^1))_{i=1}^s,$$

where pk_i^b, sk_i^b are generated by $\mathcal{K}_{McE}$, $b \in \{0, 1\}$, $i \in [s]$. The encryption algorithm $\mathcal{E}_\Sigma$ takes as input a message $\mathbf{m} = (\mathbf{m}_1 \parallel \ldots \parallel \mathbf{m}_s)$, where $\mathbf{m}_i \in \mathbb{F}_2^l$, and a public-key pk. Then $\mathcal{E}_\Sigma$ generates two keys $\mathbf{dsk}, \mathbf{vk}$ for one-time strong unforgeable signature scheme, where $\mathbf{vk} = (vk_1, \ldots, vk_s)$, and outputs ciphertext

$$\tilde{\mathbf{c}} = \mathbf{c} \parallel \mathbf{vk} \parallel \sigma,$$

where $\mathbf{c} = \mathbf{c}_1 \parallel \ldots \parallel \mathbf{c}_s$ and σ is a signature of vector $\mathbf{c}$ with the key $\mathbf{dsk}$. Each $\mathbf{c}_i$ has the form

$$\mathbf{c}_i = \mathbf{c}_i^1 \parallel \mathbf{c}_i^2 = \{(\mathbf{m}_i \parallel \mathbf{r}_i)R_\pi\}_{pk^{vk_i}}^{\mathrm{McE}} \parallel \{(\mathbf{m}_i \parallel \mathbf{r}_i \oplus \mathbf{1})R_\pi\}_{pk^{vk_i}}^{\mathrm{McE}}, \tag{1}$$

where $\mathbf{m}_i \in \mathbb{F}_2^l$, $\omega \subset_R [k]$, $|\omega| = l$, $\mathbf{r}_i \in_{\mathrm{R}} \mathbb{F}_2^{k-l}$, $\pi \in_R \mathcal{G}(\omega)$. The error vectors $\mathbf{e}_i^1$ and $\mathbf{e}_i^2$ generated in McE-encryption in the left and right parts, respectively, are chosen such that $\mathbf{e}_i^1 \in_R \mathcal{E}_{n,t}$, $\mathbf{e}_i^2 \in_R \mathcal{E}_{n,t,\mathrm{supp}(\mathbf{e}_i^1)}$. Decryption algorithm $\mathcal{D}_\Sigma$ takes as input a secret-key sk and

a ciphertext $\tilde{\mathbf{c}}$, and outputs either a message $\mathbf{m} \in \mathbb{F}_2^{sl}$ or the error symbol $\perp$. On the first step, $\mathcal{D}_\Sigma$ checks signature of the message. If check fails, then $\mathcal{D}_\Sigma$ outputs $\perp$, otherwise it computes $\mathbf{m} = \mathbf{m}_1 \parallel \ldots \parallel \mathbf{m}_s$, where

$$\mathbf{m}_i = \Pi_{\eta_i}(\{\mathbf{c}_i^1\}_{sk^{vk_i}}^{\mathrm{McE}}), \quad \eta_i = [k] \setminus \mathrm{supp}(\{\mathbf{c}_i^1\}_{sk^{vk_i}}^{\mathrm{McE}} - \{\mathbf{c}_i^2\}_{sk^{vk_i}}^{\mathrm{McE}}).$$

If $\eta_1 = \ldots = \eta_s$, then $\mathcal{D}_\Sigma$ outputs $\mathbf{m}$ else $\perp$.

Let us introduce additional notions. Denote public key pk^{vk_i} from (1) as matrix G_i, $\mathbf{1}$ as all-ones vector from $\{0,1\}^{k-l}$, and $\mathbf{0}$ as all-zeroes vector from $\{0,1\}^l$. Then for matrix G_i and secret permutation $(k \times k)$-matrix R_π, $\pi \in \mathcal{G}(\omega)$, define $(l \times n)$-matrix G_i^1 and $(k-l \times n)$-matrix G_i^2 such that

$$\begin{pmatrix} G_i^1 \\ \hline G_i^2 \end{pmatrix} = R_\pi G_i.$$

Then we can write

$$\begin{aligned} \mathbf{c}_i = \mathbf{c}_i^1 \parallel \mathbf{c}_i^2 &= \{(\mathbf{m}_i \parallel \mathbf{r}_i)R_\pi G_i \oplus \mathbf{e}_i^1\} \parallel \{(\mathbf{m}_i \parallel \mathbf{r}_i \oplus \mathbf{1})R_\pi G_i \oplus \mathbf{e}_i^1\} = \\ &= \{\mathbf{m}_i G_i^1 \oplus \mathbf{r}_i G_i^2 \oplus \mathbf{e}_i^1\} \parallel \{\mathbf{m}_i G_i^1 \oplus (\mathbf{r}_i \oplus \mathbf{1})G_i^2 \oplus \mathbf{e}_i^2\}. \end{aligned} \tag{2}$$

Now one can suggest security parameters.

3. Security parameters and experiments

3.1. Security parameters

Let us consider the general security parameters of the system: underlying linear $[n,k,d]$-code C, plaintext length l and one-time strong signature scheme. Since $(pk_i^b, sk_i^b) = \mathcal{K}_{\mathrm{McE}}(N)$, $b \in \{0,1\}$, $i \in [s]$, then one can use known results of evaluating the code parameters of the original McEliece cryptosystem. In general, in [8] it is recommended to choose cryptosystem parameters with at least 86 security bits (for 2021 year). So, according to table 1.1 from [9] it is suggested to use $[4096, 3604, 83]$-code with 129 security bits. Then to prevent finding ω from $\mathbf{c}_i^1 \oplus \mathbf{c}_i^2 = (\mathbf{0} \parallel \mathbf{1})R_\pi G_i \oplus \mathbf{e}_i^1 \oplus \mathbf{e}_i^2 = \mathbf{1}G_i^2 \oplus \mathbf{e}_i^1 \oplus \mathbf{e}_i^2$ (see (2)) we recommend to choose l with a restriction $14 \leqslant k-l \leqslant k-14$. Particularly, if $l = 3604-14$, then the adversary has to enumerate $\binom{3590}{3604}$ variants (about 129 bits) to find ω from $\mathbf{1}G_i^2$.

It is proposed to use an one-time strong signature scheme, on the one hand, resistant to quantum attacks, on the other hand, having a small public key size (since the number of repetitions s is equal to the size of the verification key). In [10] authors compared different signature schemes. So, according to table 2 from [10] we suggest to use Stern signature as a one-time strong signature scheme with a small public key size (347 bits).

3.2. Experiments

The theoretical proof of the security of the cryptosystem under consideration is based on the randomness of vectors $\mathbf{1}G_i^2 \oplus \mathbf{e}_i^1 \oplus \mathbf{e}_i^2$ and $\mathbf{r}_i G_i^2 \oplus \mathbf{e}_i^1$. Thus, the aim of experiments is to find a dependence of randomness of these vectors on the parameter l. It is important to note that in [11] authors consider similar vector to $\mathbf{r}_i G_i^2 \oplus \mathbf{e}_i^1$. Based on time complexity for the "low weight codeword" attack, the authors suggest to use specific l. In our case, to implement such attack, an adversary has to find the set ω to determine the matrix G_i^2. For l proposed above, the time complexity will be at least 2^{129}.

The experiments are carried out as follows. The NIST statistical test suite [12] is used to test the randomness of vectors. The encryption algorithm of our construction is implemented using C# language. To generate random vectors, we use a cryptographic generator from

namespace System.Security.Cryptography of C#. Since the aim of experiments is to find the dependence of randomness of cyphertexts on the parameter l, we generated several sets of random vectors from $\{0,1\}^k$ having special weight. In the case when we test randomness of vector $\mathbf{r}_i G_i^2 \oplus \mathbf{e}_i^1$, we generate random vectors from $\{0,1\}^k$ having weight less or equal $k-l$. In case when we test randomness of vector $\mathbf{1}G_i^2 \oplus \mathbf{e}_i^1 \oplus \mathbf{e}_i^2$, we generate random vectors from $\{0,1\}^k$ having weight exactly $k-l$. In particular, we generate 10000 vectors for each message type and parameter l. For the purity of the experiment, we also present the number of test passes for random vectors $\mathbf{v}$ from $\{0,1\}^k$ generated by cryptographic generator with fixed weight. The results of experiments are presented in the Table. Symbol "$*$" means that $\mathbf{r}_i$ have weight exactly 1 (otherwise $\mathrm{wt}(\mathbf{r}_i) = 0$ and $\mathbf{r}_i G_i^2 \oplus \mathbf{e}_i^1 = \mathbf{e}_i^1$).

Number of tests passed out of 10 000 conducted

$k-l$	$\mathbf{v}$, $\mathrm{wt}(\mathbf{v}) = k-l$		$\mathbf{r}_i G_i^2 \oplus \mathbf{e}_i^1$, $\mathrm{wt}(\mathbf{r}_i) \leqslant k-l$		$\mathbf{1}G_i^2 \oplus \mathbf{e}_i^1 \oplus \mathbf{e}_i^2$	
	Average	Minimum	Average	Minimum	Average	Minimum
1	714	0	9850*	9630*	9843	9610
14	1528	0	9852	9626	9852	9648
66	1859	0	9851	9636	9850	9611
112	2097	0	9852	9582	9860	9651
225	2103	0	9854	9625	9854	9650
450	2697	0	9851	9594	9847	9623
901	2756	0	9844	9606	9852	9602
1700	7302	598	9850	9601	9851	9620
1802	9881	9532	9849	9600	9844	9625
2703	2041	0	9848	9613	9853	9620
3604	714	0	9843	9576	9862	9406

Thus, the results obtained show that the considered ciphertexts pass similar number of tests for all possible values of the parameter l.

REFERENCES

1. NIST. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography`.
2. Classic McEliece: conservative code-based cryptography. `https://classic.mceliece.org/nist/mceliece-20171129.pdf`.
3. *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory. DSN Progress Report, 1978, pp. 42–44.
4. *Dottling N., Dowsley R., Quade J. M., and Nascimento A. C. A.* A CCA2 secure variant of the McEliece cryptosystem. IEEE Trans. Inform. Theory, 2012, vol. 58(10), pp. 6672–6680.
5. *Kosolapov Y. V. and Turchenko O. Y.* Efficient S-repetition method for constructing an IND-CCA2 secure McEliece modification in the standard model. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2020, vol. 13, pp. 80–84.
6. *Persichetti E.* On a CCA2-secure variant of McEliece in the standard model. Provable Security, 2018, vol. 11192, pp. 165–181.
7. *Rosen A. and Segev G.* Chosen-ciphertext security via correlated products. Proc. 6th Theory of Cryptography Conf., San Francisco, CA, USA, March 15–17, 2009, pp. 419–436.
8. *Lenstra A. K. and Verheul E. R.* Selecting cryptographic key sizes // J. Cryptology, 2004, vol. 14, pp. 446–465

9. *Bernstein D. J., Chou T., and Schwabe P.* McBits: Fast constant-time code-based cryptography. LNCS, 2013, vol. 8086, pp. 250–272.
10. *Barreto A. and Misoczki R.* A New One-Time Signature Scheme from Syndrome Decoding. IACR Cryptology ePrint Archive, 2010.
11. *Nojima R., Imai H., Kobara K., et al.* Semantic security for the McEliece cryptosystem without random oracles. Designs, Codes, Cryptogr., 2008, vol. 49, pp. 289–305.
12. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. `https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf`.

UDC 003.26 DOI 10.17223/2226308X/14/25

AN IMPROVEMENT OF CRYPTOGRAPHIC SCHEMES BASED ON THE CONJUGACY SEARCH PROBLEM[1]

V. A. Roman'kov

The key exchange protocol is a method of securely sharing cryptographic keys over a public channel. It is considered as important part of cryptographic mechanism to protect secure communications between two parties. The Diffie — Hellman protocol, based on the discrete logarithm problem, which is generally difficult to solve, is the most well-known key exchange protocol. One of the possible generalizations of the discrete logarithm problem to arbitrary noncommutative groups is the so-called conjugacy search problem: given two elements g, h of a group G and the information that $g^x = h$ for some $x \in G$, find at least one particular element x like that. Here g^x stands for $x^{-1}gx$. This problem is in the core of several known public key exchange protocols, most notably the one due to Anshel et al. and the other due to Ko et al. In recent years, effective algebraic cryptanalysis methods have been developed that have shown the vulnerability of protocols of this type. The main purpose of this short note is to describe a new tool to improve protocols based on the conjugacy search problem. This tool has been introduced by the author in some recent papers. It is based on a new mathematical concept of a marginal set.

Keywords: *cryptography, key exchange protocol, conjugacy search problem, marginal set, algorithm.*

1. Introduction

The first detailed proposal for a key exchange protocol, due to Diffie and Hellman [1], was based on the discrete logarithm problem for a finite field. This protocol is one of the earliest practical examples of public key exchange implemented within the field of cryptography. It was followed by few alternative proposals for key exchange protocols, all based on commutative algebraic structures.

Noncommutative cryptography is the area of cryptology where the cryptographic primitives, methods, and systems are based on algebraic structures like semigroups, groups and rings which are noncommutative. One of the earliest applications of a noncommutative algebraic structure for cryptographic purposes was the use of braid groups to develop the Commutator key exchange protocol by Anshel, Anshel and Goldfeld (AAG) [2] and the noncommutative key exchange protocol on braids by Ko et al. [3]. Later, several other noncommutative structures like nilpotent and polycyclic groups, and matrix groups have been identified as potential candidates for cryptographic applications.

[1]The research was supported by a grant from the Russian Science Foundation (project no. 19-71-10017).