AIMS Mathematics, 9(10): 26961–26982.
DOI:10.3934/math.20241312
Received: 12 June 2024
Revised: 23 August 2024
Accepted: 28 August 2024
Published: 18 September 2024

https://www.aimspress.com/journal/Math

Research article

The security analysis of the key exchange protocol based on the matrix power function defined over a family of non-commuting groups

Aleksejus Mihalkovich*, Jokubas Zitkevicius and Eligijus Sakalauskas

Department of Applied Mathematics, Kaunas University of Technology, Studentu str. 50, Kaunas, LT-51368, Lithuania

* **Correspondence:** Email: aleksejus.michalkovic@ktu.lt; Tel: +37060014070.

Abstract: In this paper, we revisited the previously proposed key exchange protocol based on the matrix power function. We prove that the entries of the public key matrices of both parties of the protocol are uniform. Using this result we defined a security game for our protocol and show that the malicious attacker cannot gain any significant advantage in winning this game by applying faithful representation or the linearization approaches. Moreover, we showed that the shared key is computationally indistinguishable from the imitation key if the security parameters are properly chosen.

Keywords: matrix power function; security analysis; non-commuting cryptography; uniform distribution; statistical analysis
Mathematics Subject Classification: 20F05, 62R99, 68W30, 94A60

1. Introduction

Since the groundbreaking paper [1] by W. Diffie and M. Hellman, the branch of public key cryptography has greatly developed. A simple idea of using a pair of keys , a private key PrK and a public key PuK, to produce a common key is nowadays widely applied to encrypt a message.

The general idea of the Diffie-Hellman (DH) key exchange protocol (KEP) uses the notion of a cyclic group $\mathbb{G}$ of order $|\mathbb{G}|$ together with two operations: a group operation $*$ and the multiplication by a scalar, which we denote as $g^{\alpha} = g * g * \ldots * g$, where $g \in \mathbb{G}$ and α is a scalar chosen from the ring of integers $\mathbf{Z}_{|\mathbb{G}|}$. In the context of DH KEP, the group operation is the multiplication of group elements and the multiplication by a scalar is the exponentiation. Having agreed on the cyclic group $\mathbb{G}$ and its generator g, two entities of the protocol, generally called Alice and Bob, perform the following actions to obtain a shared key [1]:

- Alice chooses at random a scalar $\alpha \in \mathbf{Z}_{|\mathbb{G}|}$ and calculates $a = g^{\alpha}$. Her private key is $PrK_A = \alpha$ and

her public key is $PuK_A = a$.
- Bob chooses at random a scalar $\beta \in \mathbf{Z}_{|\mathbb{G}|}$ and calculates $b = g^\beta$. His private key is $PrK_B = \beta$ and his public key is $PuK_B = b$.
- Alice and Bob swap their public keys PuK_A and PuK_B.
- Alice calculates the shared key as b^α whereas Bob computes a^β. The result is a shared key $k_0 = g^{\alpha\beta}$.

Note here and elsewhere in this paper that the random choice is uniform in the appropriate set, e.g., α and β are independent random variables uniformly distributed in $\mathbf{Z}_{|\mathbb{G}|}$.

The security of DH KEP relies on the difficulty of calculating the discrete logarithm of the public key, say a, base g. The importance of this problem can be seen from the security game—an interaction between the attacker $\mathcal{A}$ and the challenger C, where the goal of $\mathcal{A}$ is to gain any kind of valuable information to predict the outcome of actions performed by the challenger with a probability significantly different from the coin toss experiment. The following security game is aimed at distinguishing between the shared key k_0 and a random element $k_1 \in \mathbb{G}$ [2]:

Security Game 1. *Let the cyclic group $\mathbb{G}$ and its generator g be fixed. For the randomly chosen value of $\delta \in \{0, 1\}$, we define following experiment:*

(1) The challenger C generates the private keys of both Alice and Bob, $PrK_A = \alpha$ and $PrK_B = \beta$, respectively.
(2) C generates public keys $a = g^\alpha$ and $b = g^\beta$.
(3) If $\delta = 0$, then C calculates the shared key $k_0 = g^{\alpha\beta}$.
(4) If $\delta = 1$, then C generates a random value γ and computes $k_1 = g^\gamma$.
(5) The challenger sends the triplet (a, b, k_δ) to $\mathcal{A}$.

Relying on the obtained data, $\mathcal{A}$ outputs a guess $\delta_{\mathcal{A}}$. He wins the game if $\delta = \delta_{\mathcal{A}}$.

The presented security game is a basis of the decisional Diffie-Hellman (DDH) assumption, i.e., given the public key of Alice and Bob, the attacker $\mathcal{A}$ cannot distinguish between their shared key $k_0 = g^{\alpha\beta}$ and a random element $k_1 \in \mathbb{G}$ with a probability significantly different from $\frac{1}{2}$.

Security Game 1 can also be generalized to restrict the domain of the secret elements, e.g., by using polynomials [3]. Furthermore, instead of the standard exponentiation, any conjectured one-way function (OWF) $y = \varphi(x)$ can be used. Denoting the set of all possible arguments of φ by $\mathbb{X}$ and the set of all possible values by $\mathbb{Y}$ we obtain the following general form of the DH KEP:

- Alice randomly chooses her private key $\alpha \in \mathbb{X}$. Her public key is $a = \varphi(\alpha)$;
- Bob randomly chooses his private key $\beta \in \mathbb{X}$. His public key is $b = \varphi(\beta)$;
- Using the mathematical properties of the OWF φ and a binary operation $*$ defined in $\mathbb{X}$, Alice and Bob agree on the shared key $k_0 = \varphi(\alpha * \beta)$.

The generalized decisional Diffie-Hellman assumption states that, given the public keys of Alice and Bob, $\mathcal{A}$ cannot distinguish between the shared key $k_0 = \varphi(\alpha * \beta)$ and the imitation key $k_1 = \varphi(\gamma)$, where $\gamma \in \mathbb{X}$ is randomly chosen.

Note that if $\mathcal{A}$ can solve the discrete logarithm problem (DLP), i.e., if he can find such a scalar $\hat{\alpha}$ that $g^{\hat{\alpha}} = a$ (or, more generally, $\varphi(\alpha) = a$), then he wins with probability $\mathbb{P}[\delta = \delta_{\mathcal{A}}] = 1$, since he can

simply check if $b^{\hat{\alpha}} = k_\delta$ (a similar action can be performed in the general case). For this reason, the sets $\mathbb{X}$ and $\mathbb{Y}$ have to be large enough to ensure that the DLP is hard to solve in reasonable time. This problem is also related to the computational Diffie-Hellman (CDH) assumption, which states that it is hard for $\mathcal{A}$ to calculate the shared key $k_0 = \varphi(\alpha * \beta)$ given public keys $a, b \in \mathbb{Y}$ of Alice and Bob.

Shortly after DH KEP, another widely used idea was introduced by R. Rivest, A. Shamir, and L. Adleman in [4]. The security of their proposal is based on the integer factorization problem (IFP), i.e., finding the prime factors p, q of a composite integer $n = pq$.

However, in his paper [5], P. Shor has shown that both DLP and IFP can be solved in polynomial time using quantum computers. Therefore, due to constant developments in this area, we are on the verge of extinction of cryptosystems based on these problems, meaning that the days of DH KEP and RSA are numbered. The thread is so serious that in 2016 the National Institute of Standards and Technology (NIST) announced a call for proposals of quantum-safe cryptographic schemes for their future standardization [6]. As of 2022, three digital signature algorithms and one public-key encryption algorithm have been selected as finalists [7]. Three out of four of these algorithms are based on hard mathematical problems in lattices whereas the SPHINCS+ is based on the security of hash functions. Furthermore, the development of quantum-safe algorithms continues [8].

One of the cryptographic primitives developed in the early days of quantum-secure algorithm research was presented by a group of Korean researchers in [9]. The authors used a conjugation relation in the so-called braid group as a basis of their KEP. This is one of the first examples of a KEP based on a conjugation search problem (CSP) defined in a non-commuting group. Interestingly enough, the conjugation relation resembles exponentiation in the following way: if x and y are two commuting elements of a non-commuting group $\mathbb{G}$ and $g \in \mathbb{G}$ is random, then

$$xywy^{-1}x^{-1} = yxwx^{-1}y^{-1} = (xy)w(xy)^{-1}.$$

In fact, we have just presented the expression for the shared key k_0 of the Ko-Lee KEP. Moreover, we have the private keys $PrK_A = x$, $PrK_B = y$ and the public keys can be written as $PuK_A = g^x$ and $PuK_B = g^y$, where we use the notation $g^x = xgx^{-1}$. Therefore, $k_0 = g^{xy}$ and we can adapt the Security Game 1 to fit the Ko-Lee protocol with minor changes:

Security Game 2. *Let the non-commuting group $\mathbb{G}$, its commuting subgroup $\mathbb{H}$, and an element $g \in \mathbb{G} \setminus \mathbb{H}$ be fixed. For the randomly chosen value of $\delta \in \{0, 1\}$, we define following experiment:*

(1) The challenger C generates the private keys of both Alice and Bob, $x \in \mathbb{H}$ and $y \in \mathbb{H}$, respectively.
(2) C generates public keys $a = g^x$ and $b = g^y$.
(3) If $\delta = 0$, then C calculates the shared key $k_0 = g^{xy}$.
(4) If $\delta = 1$, then C generates a random value $z \in \mathbb{H}$ and computes the imitation key $k_1 = g^z$.
(5) The challenger sends the triplet (a, b, k_δ) to $\mathcal{A}$.

Relying on the obtained data, $\mathcal{A}$ outputs a guess $\delta_{\mathcal{A}}$. He wins the game if $\delta = \delta_{\mathcal{A}}$.

Unfortunately, this proposal suffered a failure due to the result by V. Shpilrain, who showed in [10] that CSP is unnecessary and insufficient to ensure the security of Ko-Lee protocol. In fact, CSP can be replaced by a double coset problem, i.e. the attacker $\mathcal{A}$ can swap the private key $PrK_A = x$ for a pair (x_1, x_2) such that $x_1, x_2 \in \mathbb{H}$ and $a = x_1 g x_2$, and win Security Game 2 by finding this pair rather than the original key.

Since 2007, E. Sakalauskas together with coauthors has been developing cryptographic primitives using non-commuting algebraic structures. Recent work in this field is devoted to the study of one particular highly non-linear mapping called the matrix power function (MPF). This mapping was first introduced by E. Sakalauskas in [11]. During these years, we demonstrated various applications of MPF in symmetric and asymmetric cryptography. However, early protocols presented in [11] and [12] were broken using methods of linear algebra in [13]. Though in papers [14] and [15], we were able to fix the flaws found by the authors of [13], and recently we turned our attention to sampling the base matrix entries from non-commuting groups.

In our recent papers, we have presented a KEP [16] and the sigma identification protocol [17] based on MPF defined over the non-commuting modular group generally denoted by $\mathbb{M}_{2^t}$. Our previous results have shown some promise that the proposed cryptographic primitives might be quantum-safe [16]. In this paper, we aim to formalize the previously presented results by adapting Security Game 1 to fit our proposal. Furthermore, we present the theoretical results on the distribution of the public key entries. We also back up these results by statistical research which demonstrates that the attacker $\mathcal{A}$ cannot distinguish the shared key from an imitation key.

2. Mathematical background

Let us recall definitions of the MPF mapping and the family of non-commuting groups we use in this paper.

Formally, MPF is a mapping acting on matrices in a way similar to classic matrix multiplication. However, instead of addition and multiplication operations defined for group elements, similarly to DH KEP, we use a group operation along with multiplication by a scalar. This way, we can randomly sample the entries of the base matrix from a multiplicative group $\mathbb{G}$ without a need to define another group operation other than the one defined already. We call $\mathbb{G}$ a *platform group*. We also denote the set of $m \times m$ matrices with entries from $\mathbb{G}$ by $\mathbb{G}^{m\times m}$.

The multiplication by a scalar operation is defined exactly as DH KEP suggests, i.e., for any element $g \in \mathbb{G}$ and a scalar α, we define $g^\alpha = g * g * \ldots * g$, i.e., g is multiplied by itself α times. As in the case of DH KEP, the scalars are randomly chosen from the ring of integers $\mathbb{Z}_{|\mathbb{G}|}$. We refer to it as a *power ring*.

Using these operations, we can define one-sided MPFs as well as the two-sided MPF as follows [11]:

Definition 1. *Let* $\mathbf{W} \in \mathbb{G}^{m\times m}$ *and* $\mathbf{X} \in \mathbb{Z}_{|\mathbb{G}|}^{m\times m}$. *The left-sided MPF is an action* $LMPF(\mathbf{X}, \mathbf{W}) : \mathbb{Z}_{|\mathbb{G}|}^{m\times m} \times \mathbb{G}^{m\times m} \to \mathbb{G}^{m\times m}$ *denoted as*

$$^{\mathbf{X}}\mathbf{W} = \mathbf{E}_L, \tag{2.1}$$

where the entries of LMPF value matrix $\mathbf{E}_L$ *are calculated as follows:*

$$(e_L)_{ij} = \prod_{k=1}^{m} w_{kj}^{x_{ik}}.$$

Definition 2. *Let* $\mathbf{W} \in \mathbb{G}^{m\times m}$ *and* $\mathbf{Y} \in \mathbb{Z}_{|\mathbb{G}|}^{m\times m}$. *The right-sided MPF is an action* $RMPF(\mathbf{W}, \mathbf{Y}) : \mathbb{G}^{m\times m} \times \mathbb{Z}_{|\mathbb{G}|}^{m\times m} \to \mathbb{G}^{m\times m}$ *denoted as*

$$\mathbf{W}^{\mathbf{Y}} = \mathbf{E}_R, \tag{2.2}$$

where the entries of MPF value matrix $\mathbf{E}_R$ *are calculated as follows:*

$$(e_R)_{ij} = \prod_{k=1}^{m} w_{ik}^{y_{kj}}.$$

Definition 3. *Let* $\mathbf{W} \in \mathbb{G}^{m\times m}$ *and* $\mathbf{X}, \mathbf{Y} \in \mathbb{Z}_{|\mathbb{G}|}^{m\times m}$. *The two-sided MPF is an action* $MPF(\mathbf{X}, \mathbf{W}, \mathbf{Y}) : \mathbb{Z}_{|\mathbb{G}|}^{m\times m} \times \mathbb{G}^{m\times m} \times \mathbb{Z}_{|\mathbb{G}|}^{m\times m} \to \mathbb{G}^{m\times m}$ *denoted as*

$$^{\mathbf{X}}\mathbf{W}^{\mathbf{Y}} = \mathbf{E}, \tag{2.3}$$

where the entries of MPF value matrix $\mathbf{E}$ *are calculated as follows:*

$$e_{ij} = \prod_{k=1}^{m}\prod_{l=1}^{m} w_{kl}^{x_{ik}y_{lj}}.$$

It has been previously shown in [11] that for a commuting platform group $\mathbb{G}$, the two-sided MPF can be defined in terms of one-sided MPFs due to the associativity property, i.e.,

$$\left(^{\mathbf{X}}\mathbf{W}\right)^{\mathbf{Y}} = {}^{\mathbf{X}}\left(\mathbf{W}^{\mathbf{Y}}\right) = {}^{\mathbf{X}}\mathbf{W}^{\mathbf{Y}}. \tag{2.4}$$

However, if $\mathbb{G}$ is non-commuting, then in general Eq (2.4), does not hold. This is one of the key features of the MPF mapping which we use in our research to inflict restrictions on the public parameters of our KEP.

We also emphasize the importance of the two operations used in the definition of MPF as opposed to the classic matrix multiplication, i.e., one group operation and a multiplication by a scalar as opposed to two group operations. This fact was previously misinterpreted by M. Durcheva in her paper [18]. She used a tropical group as a platform. However, she also used the same ring $\mathbb{R}$ to define both the tropical semiring $\langle \mathbb{R}, \min, + \rangle$ and the semiring of scalars, i.e., entries of power matrices are in $\mathbb{R}$. Therefore, she essentially used two group operations of $\langle \mathbb{R}, +, \cdot \rangle$ to define her version of MPFs, since '+' comes from the tropical semiring as a "multiplication" action and the '$\cdot$' comes from 'exponentiation' by a scalar from $\mathbb{R}$ (see the toy example in [18]). This choice of operations turns her mappings into simple matrix multiplications rather than MPFs in a sense presented here. Due to the simplicity of operations used in [18], it took less than a year to break her proposal [19]. However, one could use, for example elliptic curves with multiplication by a scalar to define an MPF mapping. Interestingly enough, the above definitions would exactly match the classic matrix multiplication in notation, but would still be considered MPFs due to a group operation of elliptic curves (addition), and the multiplication by a scalar. While extra research might be needed, we think that the security of a KEP based on MPF defined over elliptic curves leaves much to be desired since it is similar to the schemes presented in [11] and [12]. Therefore, linear algebra can be used to compromise the scheme due to the existence of the discrete logarithm mapping in elliptic curves.

Let us also consider another example, namely a group of invertible integers modulo $n = p_1^{k_1} p_2^{k_2} \dots p_r^{k_r}$ denoted as $\mathbb{Z}_n^*$. Notably, the group $\mathbb{Z}_n^*$ is not cyclic and hence it may seem that defining a discrete logarithm mapping is not possible. However, all of the groups $\mathbb{Z}_{p_j^{k_j}}^*$, where $j = 1, 2, \dots r$ are cyclic and hence the Chinese remainder theorem can be used to define an analog of a discrete logarithm

mapping by considering a Cartesian product $\mathbb{Z}^*_{p_1^{k_1}} \times \mathbb{Z}^*_{p_2^{k_2}} \times \ldots \times \mathbb{Z}^*_{p_r^{k_r}}$ which is isomorphic to the original group. Therefore, for any element $w \in \mathbb{Z}^*_n$, we can define a mapping $\mathrm{dlog}_{g_1,g_2,\ldots,g_r}(w)$ as follows:

$$\mathrm{dlog}_{g_1,g_2,\ldots,g_r}(w) = \left(\mathrm{dlog}_{g_1}(w \bmod p_1^{k_1}), \mathrm{dlog}_{g_2}(w \bmod p_2^{k_2}), \ldots, \mathrm{dlog}_{g_r}(w \bmod p_r^{k_r})\right), \tag{2.5}$$

where g_j generates the group $\mathbb{Z}^*_{p_j^{k_j}}$. This mapping along with several facts from number theory and linear algebra can be used to break the MPF-based schemes if $\mathbb{Z}^*_n$ is used as a platform group.

To prevent applications of these attacks to our new schemes, we use non-commuting platform groups to define MPFs. Currently our attention has turned to the family of modular cyclic groups denoted by $\mathbb{M}_{2^t}$ and defined in the following way (e is the identity of this group) [20]:

$$\mathbb{M}_{2^t} = \langle a, b | a^{2^{t-1}} = e, b^2 = e, ab = ba^{2^{t-2}+1} \rangle. \tag{2.6}$$

We can see from the presented definition that the generators a and b do not commute. Furthermore, the group $\mathbb{M}_{2^t}$ contains exactly 2^t elements and the maximal multiplicative order is 2^{t-1}. Therefore, the multiplication by a scalar operation uses integers from $\mathbb{Z}_{2^{t-1}}$.

Despite the word 'cyclic' in its name, there is no single element that would generate the whole group as opposed to, for example the ring of integers $\mathbb{Z}_p$, where p is a prime number. Moreover, due to results presented in [21–23], these groups cannot be split into a product of two or more cyclic groups. Hence no mapping analog of the discrete logarithm can be defined in the considered family of groups as opposed to the examples presented above.

However, in order to define a working scheme, we make use of the following two cyclic subgroups of maximal multiplicative order:

$$\langle a \rangle = \{e, a, a^2, \ldots, a^{2^{t-1}-1}\}, \tag{2.7}$$

$$\langle ba \rangle = \{e, ba, (ba)^2, \ldots, (ba)^{2^{t-1}-1}\}. \tag{2.8}$$

Note that the elements from distinct subgroups in general do not commute. However, they both share even powers of a, which make up a half of the elements in each subgroup.

In the next section, we present the key exchange protocol previously presented in [16]. We slightly modify the previous version of our protocol to make the proof of validity more clear. Furthermore, the natural restrictions on the public power matrices are modified to be more convenient to work with.

3. Key exchange protocol

We assume that Alice and Bob have agreed on a platform group $\mathbb{M}_{2^t}$. The base matrix $\mathbf{W}$ is chosen at random to fit the following form:

Template 1. *Choose the base matrix* $\mathbf{W}$ *so that*

$$\mathbf{W} = \begin{pmatrix} ba^{2\omega_{11}+1} & a^{\omega_{12}} & \ldots & b^{\alpha_{1c}}a^{2\omega_{1c}+\alpha_{1c}} & \ldots & ba^{2\omega_{1m}+1} \\ a^{2\omega_{21}} & a^{\omega_{22}} & \ldots & b^{\alpha_{2c}}a^{2\omega_{2c}+\alpha_{2c}} & \ldots & a^{2\omega_{2m}} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a^{2\omega_{i1}} & a^{\omega_{i2}} & \ldots & b^{\alpha_{ic}}a^{2\omega_{ic}+\alpha_{ic}} & \ldots & a^{2\omega_{im}} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a^{2\omega_{(m-1)1}} & \ldots & \ldots & \ldots & \ldots & a^{2\omega_{(m-1)m}} \\ ba^{2\omega_{m1}+1} & a^{\omega_{m2}} & \ldots & b^{\alpha_{mc}}a^{2\omega_{mc}+\alpha_{mc}} & \ldots & ba^{2\omega_{mm}+1} \end{pmatrix}. \tag{3.1}$$

We can see that each column of matrix $\mathbf{W}$ consists of commuting entries chosen from one of two cyclic subgroups $\langle a\rangle$ or $\langle ba\rangle$ defined above. However, in general, the entries of matrix $\mathbf{W}$ do not commute. Also note that we fixed the parity of powers of generators a and b in the first and last columns. We use this fact to choose an appropriate template for the left power matrices.

Template 2. *Choose matrix* $\mathbf{X}$ *in Eq* (2.1) *so that*

$$\forall i = 1, 2, \ldots, m : x_{i1} + x_{im} \equiv 0 \bmod 2. \tag{3.2}$$

We also use the c-th column to define a template for right power matrices.

Template 3. *Choose matrix* $\mathbf{Y}$ *in Eq* (2.2) *so that*

$$\forall j = 1, 2, \ldots m : y_{cj} \equiv 0 \bmod 2. \tag{3.3}$$

Note that due to Template 2, by left exponentiating by $\mathbf{X}$ we obtain an intermediate matrix ${}^{\mathbf{X}}\mathbf{W}$ with its entries aside from those in the c-th column distributed in the cyclic subgroup $\langle a\rangle$, whereas the entries of the c-th column are distributed in $\langle ba\rangle$. In other words, left exponentiation is performed with commuting entries. Also note that due to Template 3, during right exponentiation by $\mathbf{Y}$, we deal with commuting entries as well, since for any value of α_{ic}, we have $(b^{\alpha_{ic}}a^{2\omega_{ic}+\alpha_{ic}})^{y_{cj}} \in \langle a\rangle$.

An important fact to note is that we perform actions with commuting entries as long as we choose power matrices $\mathbf{X}$ and $\mathbf{Y}$ according to the defined templates and perform exponentiations by $\mathbf{X}$ and $\mathbf{Y}$ in that particular order.

Obviously, power matrices, satisfying either of the defined templates, are singular modulo 2 and hence non-invertible modulo 2^t. Proof of this fact is trivial and follows directly from the basic properties of the determinant and modular arithmetic.

Let us now assume that Alice and Bob desire to agree on a common key. Publicly known parameters are the following:

- square base $m \times m$ matrix $\mathbf{W}$ defined over $\mathbb{M}_{2^t}$ and satisfying Template 1;
- square power $m \times m$ matrices $\mathbf{L}$ and $\mathbf{R}$ defined over $\mathbb{Z}_{2^{t-1}}$ satisfying Templates 2 and 3 respectively.

Alice performs the following actions to generate private and public data:

(1) She chooses at random a vector of $2m$ coefficients $\boldsymbol{\alpha} = (\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1m}, \alpha_{21}, \alpha_{22}, \ldots, \alpha_{2m})$ and uses it to calculate two matrices as polynomials of $\mathbf{L}$ and $\mathbf{R}$, respectively:

$$\mathbf{X} = \alpha_{11}\mathbf{L} + \alpha_{12}\mathbf{L}^2 + \ldots + \alpha_{1m}\mathbf{L}^m,$$
$$\mathbf{Y} = \alpha_{21}\mathbf{R} + \alpha_{22}\mathbf{R}^2 + \ldots + \alpha_{2m}\mathbf{R}^m.$$

(2) She then uses the obtained values of $\mathbf{X}$ and $\mathbf{Y}$ to calculate matrix $\mathbf{A} = \left({}^{\mathbf{X}}\mathbf{W}\right)^{\mathbf{Y}}$.

Upon completing these steps, Alice acquires her protocol data: private key $PrK_A = \boldsymbol{\alpha}$ and her public session parameter $PuK_A = \mathbf{A}$. Alternatively, the pair $(\mathbf{X}, \mathbf{Y})$ can be kept as a private key for faster execution. As usual, Alice sends her public session parameter to Bob.

Bob performs actions similar to Alice's to obtain his data:

(1) He chooses at random a vector $\boldsymbol{\beta} = (\beta_{11}, \beta_{12}, \ldots, \beta_{2m})$ and uses it to calculate two matrices as polynomials of $\mathbf{L}$ and $\mathbf{R}$, respectively:

$$\mathbf{U} = \beta_{11}\mathbf{L} + \beta_{12}\mathbf{L}^2 + \ldots + \beta_{1m}\mathbf{L}^m,$$
$$\mathbf{V} = \beta_{21}\mathbf{R} + \beta_{22}\mathbf{R}^2 + \ldots + \beta_{2m}\mathbf{R}^m.$$

(2) He uses the obtained values of $\mathbf{U}$ and $\mathbf{V}$ to calculate matrix $\mathbf{B} = \left({}^{\mathbf{U}}\mathbf{W}\right)^{\mathbf{V}}$.

Bob now has his private key $PrK_{\mathbf{B}} = \boldsymbol{\beta}$ and his public session parameter $PuK_{\mathbf{B}} = \mathbf{B}$, which is sent to Alice.

Alice can use Bob's public session parameter to obtain the following result:

$$\mathbf{K_A} = \left({}^{\mathbf{X}}\mathbf{B}\right)^{\mathbf{Y}}. \tag{3.4}$$

Similarly, Bob can use Alice's public session parameter to obtain a matrix

$$\mathbf{K_B} = \left({}^{\mathbf{U}}\mathbf{A}\right)^{\mathbf{V}}. \tag{3.5}$$

Since Alice and Bob have two pairs of commuting matrices, i.e.,

$$\mathbf{XU} = \mathbf{UX},$$
$$\mathbf{YV} = \mathbf{VY},$$

they have agreed on a common key $\mathbf{K} = \mathbf{K}_A = \mathbf{K}_B$.

The proof of validity of the presented protocol relies on the fact that exponentiations are performed with commuting entries. First, due to Template 2 and since $\mathbf{UX} = \mathbf{XU}$, we have

$${}^{\mathbf{U}}\left(\left({}^{\mathbf{X}}\mathbf{W}\right)^{\mathbf{Y}}\right) = {}^{\mathbf{X}}\left(\left({}^{\mathbf{U}}\mathbf{W}\right)^{\mathbf{Y}}\right), \tag{3.6}$$

for any matrix $\mathbf{Y}$ satisfying Template 3. Second, since $\mathbf{VY} = \mathbf{YV}$ and $\mathbf{V}$ satisfies Template 3 as well, we have

$${}^{\mathbf{U}}\left(\left({}^{\mathbf{X}}\mathbf{W}\right)^{\mathbf{Y}}\right)^{\mathbf{V}} = {}^{\mathbf{X}}\left(\left({}^{\mathbf{U}}\mathbf{W}\right)^{\mathbf{V}}\right)^{\mathbf{Y}}. \tag{3.7}$$

In fact, since the public keys of both parties $\mathbf{A}$ and $\mathbf{B}$ consist of commuting entries, the parentheses in both Eqs (3.4) and (3.5) can be dropped, i.e., when calculating the shared key $\mathbf{K}_A = \mathbf{K}_B$, the order of actions does not matter. This fact, together the with commutation constraint on the right power matrices allows us to perform a switch of $\mathbf{Y}$ and $\mathbf{V}$ in Eq (3.7).

Note, however, that the proof of validity of our protocol heavily relies on the specified templates. In other words, if these template are neglected, Alice and Bob cannot agree on a shared key due to non-commutativity of the platform group $\mathbb{M}_{2^t}$. Moreover, we cannot generalize our protocol as presented in [12] since none of the invertible matrices satisfy Templates 2 or 3. Hence any malicious adversary is limited in his search for private keys by singular matrices only. Note that the use of the invertible matrices in [12] was one of the key issues which allowed the authors of [13] to break the protocol.

The protocol is also easy to practically implement, since it uses rather simple multiplication and exponentiation operations in $\mathbb{M}_{2^t}$ and addition and multiplication operations in $\mathbb{Z}_{2^{t-1}}$. Note that actions in $\mathbb{M}_{2^t}$ are similar to classical ones and hence the squaring technique to compute the power of an

element can be applied. Furthermore, polynomials are calculated using operations in $\mathbb{Z}_{2^{t-1}}$ and hence all the classic techniques can be applied to speed up the computation process. Therefore, the main time-consuming operation is the MPF calculation. However, its structure is similar to regular matrix multiplication and, hence, it can be performed in $O(m^3)$. Also, the powers of matrices $\mathbf{L}$ and $\mathbf{R}$ can be calculated once and stored in the memory of the device. Otherwise, the computational costs are bounded by $O(m^4)$.

In this paper, we present the basic version of the key exchange protocol inspired by the classic Diffie-Hellman and RSA algorithms. Just as those ideas are, our proposal is anonymous and hence it cannot offer protection against man-in-the-middle attack. This can be fixed by including the identification step in our algorithm. However, we do not consider this feature here.

4. Security analysis of our KEP

4.1. Uniform distribution of the public key matrix entries

Let us now consider the security of our KEP. First, we show that the following proposition holds:

Proposition 1. *Assume that matrices* $\mathbf{W}$, $\mathbf{L}$, *and* $\mathbf{R}$ *satisfy Templates 1–3, respectively, and are fixed. Let* α *be the vector of* $2m$ *coefficients uniformly sampled from* $\mathbb{Z}_{2^{t-1}}^{2m}$. *Then any entry of the public key matrix* $\mathbf{A}$ *is uniformly distributed in* $\langle a \rangle$.

To prove this claim, we need the following lemma:

Lemma 1. *Define the bilinear form* $s(n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_{ij} x_i y_j = \mathbf{x}^T \mathbf{\Lambda} \mathbf{y}$, *where* $\lambda_{ij} \in \mathbb{Z}_2$ *are uniformly chosen at random, and* $x_i, y_j \in \mathbb{Z}_2$ *are uniformly distributed random variables. Assume that* $\operatorname{rank} \mathbf{\Lambda} = r$. *The limit* $\lim\limits_{\substack{n \to +\infty \\ r \to n}} \mathbb{P}[s(n) = s_0] = \frac{1}{2}$, *where* $s_0 \in \mathbb{Z}_2$ *is a fixed value.*

Proof. Let us first consider the simplest possible case when the coefficient matrix $\mathbf{\Lambda}$ is the identity matrix. In this case, the bilinear form $s(n)$ has the following representation:

$$s(n) = \sum_{i=1}^{n} x_i y_i. \tag{4.1}$$

The following is true if $n = 1$:

$$\mathbb{P}[s(1) = 0] = \mathbb{P}[x_1 y_1 = 0] = \frac{3}{4} = \frac{1}{2} + \frac{1}{2^2},$$
$$\mathbb{P}[s(1) = 1] = \mathbb{P}[x_1 y_1 = 1] = \frac{1}{4} = \frac{1}{2} - \frac{1}{2^2}.$$

Let us define the deviation term:

$$\sigma(n) = \frac{\mathbb{P}[s(n) = 0] - \mathbb{P}[s(n) = 1]}{2}.$$

We can see that the deviation term $\sigma(1) = \frac{1}{2^2} = \frac{1}{2^{1+1}}$.

We now calculate the next iteration, i.e., for $n = 2$, we have

$$\mathbb{P}[s(2) = 0] = \frac{5}{8} = \frac{1}{2} + \frac{1}{2^3},$$
$$\mathbb{P}[s(2) = 1] = \frac{3}{8} = \frac{1}{2} - \frac{1}{2^3}.$$

Hence the deviation term is $\sigma(2) = \frac{1}{2^3} = \frac{1}{2^{2+1}}$.

Now we apply mathematical induction on n. Hence, for $n = k$, we have

$$\mathbb{P}[s(k) = 0] = \tfrac{1}{2} + \tfrac{1}{2^{k+1}},$$
$$\mathbb{P}[s(k) = 1] = \tfrac{1}{2} - \tfrac{1}{2^{k+1}}.$$

Then due to the following obvious identity

$$\mathbb{P}[s(k+1) = s_0] = \mathbb{P}[s(1) = 0] \cdot \mathbb{P}[s(k) = s_0] + \mathbb{P}[s(1) = 1] \cdot \mathbb{P}[s(k) = s_0 - 1],$$

we have

$$\mathbb{P}[s(k+1) = 0] = (\tfrac{1}{2} + \tfrac{1}{2^{k+1}}) \cdot \tfrac{3}{4} + (\tfrac{1}{2} - \tfrac{1}{2^{k+1}}) \cdot \tfrac{1}{4} = \tfrac{1}{2} + \tfrac{1}{2^{k+2}},$$
$$\mathbb{P}[s(k+1) = 1] = (\tfrac{1}{2} + \tfrac{1}{2^{k+1}}) \cdot \tfrac{1}{4} + (\tfrac{1}{2} - \tfrac{1}{2^{k+1}}) \cdot \tfrac{3}{4} = \tfrac{1}{2} - \tfrac{1}{2^{k+2}},$$

and hence we obtain the deviation term $\sigma(k+1) = \frac{1}{2^{(k+1)+1}}$, thus proving the general expressions for probabilities $\mathbb{P}[s(n) = 0]$ and $\mathbb{P}[s(n) = 1]$.

Since the identity matrix has full rank, by calculating the limit when n tends to positive infinity we get

$$\lim_{n \to +\infty} \mathbb{P}[s(n) = s_0] = \lim_{n \to +\infty} \left(\frac{1}{2} \pm \frac{1}{2^{n+1}} \right) = \frac{1}{2}.$$

More generally, let us now assume that the matrix $\mathbf{\Lambda}$ has full rank. Then the linear operator Λ maps the basis of $\mathbb{Z}_2^n$ to another basis. However, by denoting $\mathbf{z} = \mathbf{\Lambda y}$, we essentially obtain Eq (4.1) in another set of bases since the linear operator Λ is a one-to-one mapping. Hence the proven expressions for probabilities $\mathbb{P}[s(n) = 0]$ and $\mathbb{P}[s(n) = 1]$ also hold for this case as well.

Now assume that the coefficient matrix $\mathbf{\Lambda}$ has a rank $r = \operatorname{rank} \mathbf{\Lambda} < n$. Then by calculating $\mathbf{\Lambda e}_j = \mathbf{f}_j$, where $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n$ is the standard basis of $\mathbb{Z}_2^n$, we obtain a system of r linearly independent vectors and using the same technique as before for the $r \times r$ minor of matrix $\mathbf{\Lambda}$, we obtain the following expressions for the considered probabilities:

$$\mathbb{P}[s(n) = 0] = \tfrac{1}{2} + \tfrac{1}{2^{r+1}},$$
$$\mathbb{P}[s(n) = 1] = \tfrac{1}{2} - \tfrac{1}{2^{r+1}}.$$

Clearly, taking the limits of both probabilities proves the lemma, since $\lim\limits_{\substack{n \to +\infty \\ r \to n}} \mathbb{P}[s(n) = 0] = \frac{1}{2}$ and $\lim\limits_{\substack{n \to +\infty \\ r \to n}} \mathbb{P}[s(n) = 1] = \frac{1}{2}$. □

Note, that in the presented proof both, parameters n and r play their important roles in order to achieve the uniform distribution. For example, if we set $\lambda_{ij} = 1$ for all $i, j = 1, 2, \ldots, n$, we obtain a matrix $\mathbf{\Lambda}$ with rank $r = 1$. Then we have $\mathbb{P}[s(n) = 0] = \frac{3}{4}$ and $\mathbb{P}[s(n) = 1] = \frac{1}{4}$ regardless of the choice of n. In other words, if r does not tend to n, even if n tends to positive infinity, the null space may be large enough to gobble up most of the pairs $(\mathbf{x}, \mathbf{y})$.

Interestingly enough, in our setup the null space of the coefficient matrix $\mathbf{\Lambda}$ is non-trivial and hence $r < n$. However, since the public matrices $\mathbf{L}$ and $\mathbf{R}$ are generated at random to fit the presented templates, we can keep the dimension of the null space relatively small as needed.

This lemma can also be generalized to any power of 2. To shorten the paper, we do not present the full proof of the lemma but rather settle for the sketch of the proof of the following fact:

Lemma 2. *Define the bilinear form* $s(n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\lambda_{ij}x_iy_j = \mathbf{x}^T\mathbf{\Lambda}\mathbf{y}$*, where* $\lambda_{ij} \in \mathbb{Z}_{2^t}$ *are uniformly chosen at random, and* $x_i, y_j \in \mathbb{Z}_{2^t}$ *are uniformly distributed random variables. Let* $\mathbf{\Lambda}_2 = \mathbf{\Lambda} \bmod 2$ *and assume that in* $\mathbb{Z}_2^{n\times n}$*, we have* $\operatorname{rank}\mathbf{\Lambda}_2 = r$*. The limit* $\lim\limits_{\substack{n\to+\infty\\ r\to n}} \mathbb{P}[s(n) = s_0] = \frac{1}{2^t}$*, where* $s_0 \in \mathbb{Z}_{2^t}$ *is a fixed value.*

Sketch of proof. We start by dividing the elements of $\mathbb{Z}_{2^t}$ into $t+1$ disjoint sets of the form $\mathbb{S}_h = \left\{k\cdot 2^h | k = 1, 3, \ldots, 2^{t-h}-1\right\}$, where $h = 0, 1, \ldots, t-1$. Also set $\mathbb{S}_t = \{0\}$. Each element of $\mathbb{S}_h$ is equally likely and for any two indices $h_1 < h_2$, the elements of $\mathbb{S}_{h_1}$ are less probable than the elements of $\mathbb{S}_{h_2}$. We also define the deviation term as follows:

$$\sigma(t, n) = \frac{\mathbb{P}[s(n) = 0] - \mathbb{P}[s(n) = 2^{t-1}]}{2}.$$

Now we perform an iterative process by reducing matrix $\mathbf{\Lambda}$ modulo 2 and calculating probabilities $\mathbb{P}[s(n) = s_0]$ for $\mathbf{\Lambda}_2 = \mathbf{\Lambda} \bmod 2$ in $\mathbb{Z}_2$. Then we consider matrices $\mathbf{\Lambda}_4 = \mathbf{\Lambda} \bmod 4$, $\mathbf{\Lambda}_8 = \mathbf{\Lambda} \bmod 8$, and so on until we reach $\mathbf{\Lambda}$. We calculate probabilities $\mathbb{P}[s(n) = s_0]$ for all iterations in the appropriate rings. Then due to the identity:

$$\mathbb{P}[s(k+1) = s_0] = \sum_{j=0}^{2^t-1}\mathbb{P}[s(1) = j]\mathbb{P}[s(k) = s_0 - j],$$

which holds for all $k = 2, \ldots n$ we obtain the following expressions for probabilities:

$$\begin{gathered}\mathbb{P}[s(n) = 0] = \frac{1}{2^t} + \sum_{j=0}^{t-1}\left(\frac{1}{2^{t-j}}\cdot\sigma(j+1, n)\right),\\ \mathbb{P}[s(n) = s_h] = \frac{1}{2^t} + \left[\sum_{j=0}^{h-1}\left(\frac{1}{2^{t-j}}\cdot\sigma(j+1, n)\right)\right] - \frac{1}{2^{t-h}}\cdot\sigma(h+1, n).\end{gathered}$$

Note that while $\sigma(t, n)$ do vary depending on the generated coefficient matrix $\mathbf{\Lambda}$, all of these deviations are functions of n and the limit $\lim\limits_{n\to+\infty}\frac{\sigma(t_1,n)}{\sigma(t_2,n)} = 0$, if $t_1 < t_2$. Therefore, we have $\lim\limits_{\substack{n\to+\infty\\ r\to n}}\mathbb{P}[s(n) = s_0] = \frac{1}{2^t}$ for every $s_0 \in \mathbb{Z}_{2^t}$ as desired. □

Through experiments, we found that the deviation term can vary even when the rank $r = \operatorname{rank}\mathbf{\Lambda}_2$ is fixed at some value smaller than n. Moreover, even when we consider the case of $2^t = 4$ and express $\mathbf{\Lambda} = 2\mathbf{\Gamma} + \mathbf{\Lambda}_2$, where $\mathbf{\Gamma}$ is a binary matrix, the deviation term varies regardless of the rank of $\mathbf{\Gamma}$ in $\mathbb{Z}_2$. However, the deviation term only has a few possibilities and they all differ by some constant factor. For these reasons, the expressions of the probabilities are presented in terms of deviations. Also, no other ranks, e.g., $\operatorname{rank}\mathbf{\Gamma}$, were used to define the goal limit. Notably, these remarks do not change the correctness of the lemma since the main summand in all the expressions is $\frac{1}{2^t}$, which is independent of both n and r.

Using these lemmas, we prove Proposition 1.

Proof of Proposition 1. Due to the polynomial structure of the private key matrices $\mathbf{X}$ and $\mathbf{Y}$, the expression of the public key matrix $\mathbf{A}$ can be rewritten as follows:

$$\mathbf{A} = \left(\mathbf{X}\mathbf{W}\right)^{\mathbf{Y}} = \left(\left(\sum_{k=1}^{m}\alpha_{1k}\mathbf{L}^k\right)\mathbf{W}\right)^{\left(\sum_{j=l}^{m}\alpha_{2l}\mathbf{R}^l\right)}. \tag{4.2}$$

Then by expanding Eq (4.2) and focusing on the powers of generator a, we obtain the following result:

$$a'_{ij} = \alpha_1^T \mathbf{\Lambda} \alpha_2 + 2^{t-2}\gamma_{ij}, \tag{4.3}$$

where $a'_{ij} = \text{dlog}_a a_{ij}$ is a discrete logarithm of matrix $\mathbf{A}$ entry a_{ij}, α_1 and α_2 are two private vectors of coefficients to generate key matrices $\mathbf{X}$ and $\mathbf{Y}$, respectively, $\mathbf{\Lambda}$ is a matrix obtained from Eq (4.2) by calculating $\text{dlog}_a\left(\left(\mathbf{L}^k\mathbf{W}\right)^{\mathbf{R}^l}\right)$ for all possible pairs (k, l), and $\gamma_{ij} \in \mathbb{Z}_2$ takes into account the non-commutativity of the platform group $\mathbb{M}_{2^t}$.

However, we can now apply Lemma 2 to the bilinear form $\alpha_1^T \mathbf{\Lambda} \alpha_2$ in Eq (4.3) to show that its value is uniformly distributed in $\mathbb{Z}_{2^{t-1}}$. The term $2^{t-2}\gamma_{ij}$ in no way affects this result, since $\mathbb{Z}_{2^{t-1}}$ is a group under addition. □

This, of course, means that due to Lemma 2, the uniform distribution is asymptotic. For this reason, in Section 6, we perform the statistical analysis of the presented KEP to find the minimal value of m for which the distribution of the entries of the public key matrix is indistinguishable from the uniform distribution. We aim to show the validity of the decisional assumption for our proposal analogues to the one defined for DH KEP.

4.2. The security game for the presented KEP

Due to the proven theoretical result, we can see that the distribution of every individual entry of the public key matrix is asymptotically uniform in the set $\langle a \rangle$ and the deviation from the desired distribution is negligible for a sufficiently large m, which, for now, is yet to be determined. Relying on this result, we define the following security game between the attacker $\mathcal{A}$ and the challenger C:

Security Game 3. *Let the platform group-defining parameter t and the matrix order m be fixed. Also, let* $\mathbf{W}, \mathbf{L}$*, and* $\mathbf{R}$ *be the public parameters, satisfying Templates 1–3 respectively. For the randomly chosen value of* $\delta \in \{0, 1\}$*, we define the following experiment:*

(1) The challenger C generates the private keys of both Alice and Bob as vectors α *and* β *of size* $2m$ *and calculates matrices* $\mathbf{X}, \mathbf{Y}, \mathbf{U}$ *and* $\mathbf{V}$ *as polynomials of* $\mathbf{L}$ *and* $\mathbf{R}$*.*

(2) C calculates the public keys $\mathbf{A} = \left(^{\mathbf{X}}\mathbf{W}\right)^{\mathbf{Y}}$ *and* $\mathbf{B} = \left(^{\mathbf{U}}\mathbf{W}\right)^{\mathbf{V}}$ *of Alice and Bob, respectively.*

(3) If $\delta = 0$*, C calculates the matrix* $\mathbf{K}_0 = \left(^{\mathbf{X}}\mathbf{B}\right)^{\mathbf{Y}}$*, which is the shared key.*

(4) If $\delta = 1$*, C generates a vector* $\gamma = (\gamma_{11}, \ldots, \gamma_{1m}, \gamma_{21}, \ldots \gamma_{2m})$ *and calculates matrices* $\mathbf{Z}_1, \mathbf{Z}_2$*, and* $\mathbf{K}_1$ *as follows:*

$$\begin{aligned}
\mathbf{Z}_1 &= \gamma_{11}\mathbf{L} + \gamma_{12}\mathbf{L}^2 + \ldots + \gamma_{1m}\mathbf{L}^m;\\
\mathbf{Z}_2 &= \gamma_{21}\mathbf{R} + \gamma_{22}\mathbf{R}^2 + \ldots + \gamma_{2m}\mathbf{R}^m;\\
\mathbf{K}_1 &= \left(^{\mathbf{Z}_1}\mathbf{W}\right)^{\mathbf{Z}_2}.
\end{aligned}$$

(5) C sends the triplet $(\mathbf{A}, \mathbf{B}, \mathbf{K}_\delta)$ *to the attacker* $\mathcal{A}$*.*

Relying on the received data $\mathcal{A}$ *outputs a value* $\delta_{\mathcal{A}}$*. He wins the game if* $\delta_{\mathcal{A}} = \delta$*.*

The defined security game is inspired by the so-called Diffie-Hellman decisional problem presented in [2] in the form of the Attack Game 10.6. Essentially the presented security game means that $\mathcal{A}$ cannot distinguish between the shared key $\mathbf{K}_0$ and a randomly generated matrix $\mathbf{K}_1$. This is the decisional assumption of our proposal and due to the closure of Templates 2 and 3 under

exponentiation, it also resembles the generalized decisional Diffie-Hellman assumption presented in [3], where sets of polynomials are used to restrain the domain of the arguments. However, the basic concept of the game is the same: to distinguish between the shared key and the imitation one.

From the statistical point of view, this means that the entries of both matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ have the same distribution. However, due to Proposition 1 and Lemma 2, this is exactly the case for our protocol.

Due to the classic result (see [2]), the decisional assumption also implies the following result:

Proposition 2. *Let the platform group-defining parameter t and the matrix order m be fixed. Also, let* $\mathbf{W}, \mathbf{L}$ *and* $\mathbf{R}$ *be the public parameters, satisfying Templates 1–3 respectively. Given Alice's and Bob's public keys* $\mathbf{A} = \left({}^{\mathbf{X}}\mathbf{W}\right)^{\mathbf{Y}}$ *and* $\mathbf{B} = \left({}^{\mathbf{U}}\mathbf{W}\right)^{\mathbf{V}}$*, respectively, it is hard to compute the shared key* $\mathbf{K} = \left({}^{\mathbf{X}}\mathbf{B}\right)^{\mathbf{Y}}$.

The latter proposition is the computational assumption for our proposal.

Algebraically $\mathcal{A}$ may try to obtain the private key matrices (or, alternatively, coefficients of polynomials) of one of the parties, say Alice. To put it simpler, his goal is to solve the following system of matrix equations:

$$\begin{cases} \mathbf{A} = \left({}^{\mathbf{X}}\mathbf{W}\right)^{\mathbf{Y}}, \\ \mathbf{XL} = \mathbf{LX}, \\ \mathbf{YR} = \mathbf{RY}. \end{cases} \tag{4.4}$$

Moreover, the solution pair $(\mathbf{X}, \mathbf{Y})$ has to satisfy Templates 2 and 3, respectively, since otherwise the protocol falls apart due to the properties of the platform group $\mathbb{M}_{2^t}$. Problem (4.4) is the analog of DLP for our proposal.

Evidently, the polynomial construction of private matrices $\mathbf{X}$ and $\mathbf{Y}$ ensures that the last two equations in system (4.4) are satisfied. However, since $\mathbb{M}_{2^t}$ cannot be decomposed into smaller groups and since we use elements from both cyclic subgroups $\langle a \rangle$ and $\langle ba \rangle$ to define the base matrix $\mathbf{W}$, the discrete logarithm mapping is not applicable to the MPF equation of system (4.4). Furthermore, the non-commutativity of $\mathbb{M}_{2^t}$ presents an additional obstacle for solving the above system directly. Hence, we present two approaches the attacker can use to solve system (4.4) indirectly.

First, let us consider the linearization technique. Since the MPF equation in system (4.4) uses a power of 2 as a modulo, the simplest task is trying to solve a sub-problem of system (4.4) defined in $\mathbb{Z}_2$. To do so, $\mathcal{A}$ defines a mapping φ as follows:

$$\varphi : \mathbb{M}_{2^t} \mapsto \mathbb{Z}_2 : \varphi(b^{\alpha}a^{\omega}) = \omega \bmod 2,$$

i.e., φ is essentially the parity mapping, which completely ignores the generator b. $\mathcal{A}$ also defines Φ as the matrix analog of φ by entry-wise application of φ to $\mathbf{W}$. Then by denoting $\mathbf{X}_2 = \mathbf{X} \bmod 2$ and $\mathbf{Y}_2 = \mathbf{Y} \bmod 2$, $\mathcal{A}$ transforms the first equation in the system (4.4) to the following form:

$$\mathbf{X}_2\Phi(\mathbf{W})\mathbf{Y}_2 = \Phi(\mathbf{A}).$$

Expanding $\mathbf{X}_2$ and $\mathbf{Y}_2$ as polynomials of $\mathbf{L}_2 = \mathbf{L} \bmod 2$ and $\mathbf{R}_2 = \mathbf{R} \bmod 2$, the attacker obtains

$$\left(\alpha_{11}\mathbf{L}_2 + \alpha_{12}\mathbf{L}_2^2 + \ldots + \alpha_{1m}\mathbf{L}_2^m\right) \cdot \Phi(\mathbf{W}) \cdot \left(\alpha_{21}\mathbf{R}_2 + \alpha_{22}\mathbf{R}_2^2 + \ldots + \alpha_{2m}\mathbf{R}_2^m\right) = \Phi(\mathbf{A}). \tag{4.5}$$

$\mathcal{A}$ now introduces m^2 variables of the form $\gamma_k = \alpha_{1i}\alpha_{2j}$ and hence obtains a system of linear equations (SLE) with respect to the defined γ's. However, all the matrices in Eq (4.5) are singular,

and hence the obtained $m^2 \times m^2$ system is underdefined and produces a set of solutions much too large to perform a total scan. Experimental results for the platform group $\mathbb{M}_{16}$ have shown that the maximum rank of the obtained SLE is $(m-1)^2$, i.e., at least $(2m+1)$ variables are free [24]. Furthermore, for randomly chosen matrices $\mathbf{L}_2, \mathbf{R}_2$, and $\Phi(\mathbf{W})$, the maximum rank of SLE is a rare event obtained roughly 6% of the time. Hence the probability of determining α's from γ's is at most $\frac{1}{2^{2m+1}}$. This makes the linearization technique inefficient for the sub-problem defined by Eq (4.5) and hence for the initial problem defined by system (4.4) as well if the public parameters are appropriately chosen. However, this choice can be studied separately since it may be performed once and can be fixed afterward.

Another approach may involve a faithful representation of the elements of $\mathbb{M}_{2^t}$ as matrices. This technique was previously used to break protocols constructed using braid groups. In our case, the faithful representation of $\mathbb{M}_{2^t}$ exists and we denote it by $R(b^\alpha a^\omega)$. For $t = 4$, it can be found in [25]. However, as mentioned previously, MPF itself is defined in terms of a group operation and a multiplication by a scalar. Therefore, the latter action (exponentiation in our case) is applied to the element regardless of its representation. Assume that $\mathcal{A}$ uses $R()$ to the elements of $\mathbb{M}_{2^t}$ to view them as matrices. He then obtains a tensor $\mathbf{W}' = R(\mathbf{W})$, where $w'_{ij} = R(w_{ij})$ are matrices of the appropriate order. However, the faithful representation does not affect the scalars contained in the power matrices and hence the public key is computed as follows:

$$R(\mathbf{A}) = \left({}^{\mathbf{X}}\mathbf{W}'\right)^{\mathbf{Y}},$$

i.e., despite commuting entries of $R(w_{ij})$, the non-commutativity of $\mathbb{M}_{2^t}$ is preserved since otherwise the representation would not be faithful. However, the exponentiation is still applied to matrices $R(w_{ij})$ rather than its individual entries, i.e., we have

$$R(a_{ij}) = \prod_{k=1}^{m} \prod_{l=1}^{m} \left(w'_{ij}\right)^{x_{ik}y_{lj}}.$$

We now see that $\mathcal{A}$ gains nothing by using this approach.

For now we limit ourselves to the discussed approaches leaving other possible ways (which may or may not be found) to recover the private key of the user for future work.

Let us, however, mention the following fact: the discrete logarithm mapping can be applied to the expression $\left({}^{\mathbf{X}}\mathbf{B}\right)^{\mathbf{Y}}$ and to matrices $\mathbf{K}_0$ or $\mathbf{K}_1$, whichever one is given. This is due to the loss of the non-commutativity factor by that point. However, the methods of search for the private key matrices are similar to the ones given above since Templates 2 and 3 cannot be ignored. Future work may involve an exploration of the question of whether it is possible to maintain the non-commutativity of $\mathbb{M}_{2^t}$ while also establishing a shared key.

5. Comparison of our proposal to other schemes

As mentioned previously, our proposal is inspired by the DH KEP and RSA algorithms. However, as opposed to those classic protocols, the security of our proposal relies on a problem related to multivariate quadratic (MQ) equations. It has been previously shown in [26] that the MQ problem is hard to solve using Grobner bases if there are more than 80 equations and variables. Furthermore, schemes like [27, 28] also use multivariate quadratic equations defined in some field. Despite the

fact that, in our case, there are no linear terms in the MQ problem we think that the security of our proposal could be close to those schemes since the non-commutativity of the platform group $\mathbb{M}_{2^t}$ brings an additional randomness factor into the obtained multivariate system of equations. However, more investigations may be needed to approve or disapprove our claim.

6. Statistical analysis of the proposed KEP

In this section, we perform the statistical experiments to support the theoretic results presented above. Due to the asymptotic nature of the uniform distribution, here we aim to explore the dependence of the considered distribution on the public parameters m and t. To put it simply, since all of the results regarding the uniform distribution of entries were proven using limits, we explore how fast we can actually obtain a distribution statistically indistinguishable from the uniform. Furthermore, we perform a two-sample chi-squared test for the obtained samples to show that $\mathbf{K}_0$ and $\mathbf{K}_1$ are computationally indistinguishable to establish the validity of the decisional assumption defined by Security Game 3.

We consider the public parameters of the system: the group-defining parameter t and the matrix size m. For every fixed value of t, our goal is to find the minimum value of m such that for all the performed statistical tests, the appropriate null hypothesis would not be rejected at the 0.95 confidence level. We execute a simulation of Security Game 3 by repeating the described experiment n times with an exception of performing both steps 3 and 4 regardless of the value of δ. Hence we perform a total of n iterations and obtain two samples for the true shared key $\mathbf{K}_0$ and the imitation key $\mathbf{K}_1$. For a specific value of m, the base matrix $\mathbf{W}$ is kept constant throughout all the iterations. In this paper, we set the standard value for the parameter n as 200 although a larger value could be used as well. During each iteration, we append the frequencies of each element in two different arrays: the first array is appended based on generator a values and the second array is based on the matrix coordinates and the generator a value. In other words, in the first array, we consider the distribution of the matrix entries in $\langle a \rangle$ ignoring their position. In the second array, we consider the distribution of a specific matrix entry. Both of these approaches are important since the attacker may consider individual positions of the key matrix as well as the matrix as a whole. Therefore, it is also important to ensure that each individual entry of the key matrix follows the uniform distribution.

We use Pearson chi-squared (Chi-Sq.) goodness of fit to test the obtained sample for uniform distribution. To compare the shared key and the imitation key samples, we use the two-sample chi-squared test. Also, we inspect the data visually. For better visual comparison (especially when the parameter t grows), we present the graphical data using polygonal chains. That way it is more convenient to view the results for both samples in a single graph.

Let us denote by $KEP(m, t, n)$ the considered algorithm with parameters m, t and n. As an example, let us present the distribution of the shared key and imitation key matrix entries after performing $KEP(8, 4, 200)$.

The presented information in Figure 1 and all of the subsequent figures should be interpreted as follows: for each element of the form a^k, we obtain the relative frequency of that element in the shared key $\mathbf{K}_0$ and the imitation key $\mathbf{K}_1$ ignoring its position. let us denote these frequencies by $\text{freq}_{\mathbf{K}_0}(k)$ and $\text{freq}_{\mathbf{K}_1}(k)$, respectively. We plot the points $(k, \text{freq}_{\mathbf{K}_0}(k))$ in blue and $(k, \text{freq}_{\mathbf{K}_1}(k))$ in red and for better visual interpretation (especially for larger values of t), we connect these points using line segments. Ideally we should obtain a perfect straight line parallel to the x-axis if the distribution of elements of

$\mathbb{M}_{2^t}$ is uniform and the relative frequency should equal 2^{1-t}.

We can see from Figure 1 that even powers of the generator a dominate the odd ones. This coincides with the theoretical proof presented in the previous section, i.e., the asymptotic nature of the uniform distribution is clearly visible. Hence the matrix size $m = 8$ is too small if the case of platform group $\mathbb{M}_{16}$ (note that $t = 4$). For similar-looking graphs, we do not consider individual position distributions, since the overall distribution is far from uniform. Note also that even for the imitation key $\mathbf{K}_1$, the distribution is not too close to being uniform.

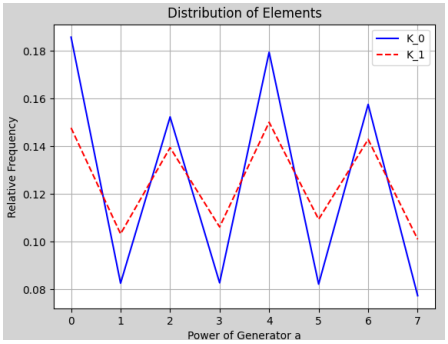
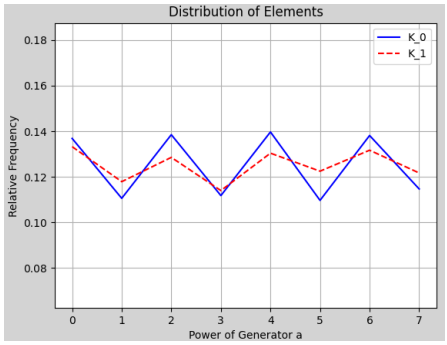
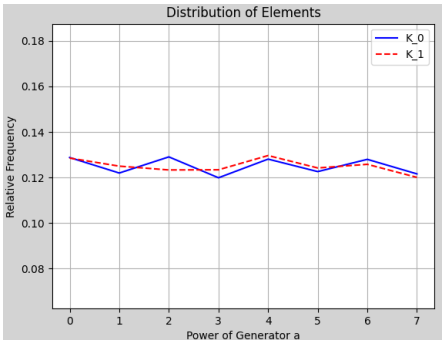


Figure 1. Statistical analysis for $KEP(8, 4, 200)$.

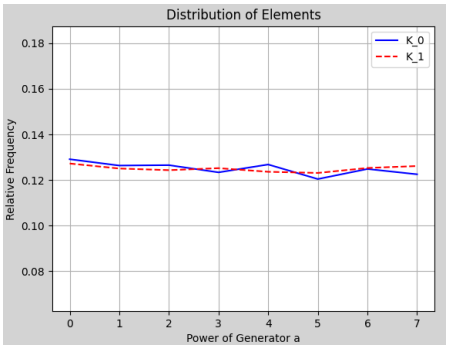
Let us now explore the asymptotic nature of the uniform distribution. We fix the parameter $t = 4$ and increase the matrix size. Several obtained graphs are displayed in Figure 2.



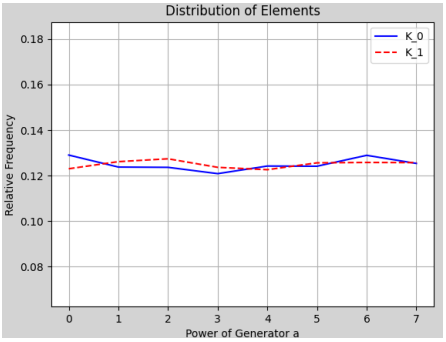
(a) $KEP(10, 4, 200)$.



(b) $KEP(12, 4, 200)$.



(c) $KEP(14, 4, 200)$.



(d) $KEP(16, 4, 200)$.

Figure 2. Statistical analysis for uniformity of shared (blue) and imitation (red) keys.

We can clearly see from Figure 2 that the pattern exists by looking at the presented frequencies: the previously observed parity effect diminishes when the parameter m increases, which coincides with the results of Section 4.2. However, we also see from Figures 2a and 2b that the matrix size is still too small to achieve the uniform distribution of the shared key entries. However, we see from Figures 2c and 2d that the obtained distributions are similar and can be considered uniform.

To sum up the obtained results in Table 1, we present the p-values for the goodness of fit tests as well as the p-value of the two-sampled chi-squared test.

Table 1. The statistical analysis results for the platform group $\mathbb{M}_{16}$.

m	$p(\chi^2_{\mathbf{K}_0})$	$p(\chi^2_{\mathbf{K}_1})$	$p(\chi^2_{\mathbf{K}_0,\mathbf{K}_1})$
8	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$
10	$< 10^{-3}$	0.333	$< 10^{-3}$
12	$< 10^{-3}$	0.334	0.008
14	0.010	0.817	0.108
16	0.422	0.883	0.598

We can see from the presented table that we cannot reject the null hypothesis at a 95% significance level for values of $m \geq 14$. However, since the p-value for $m = 14$ is close to the bottom edge, we prefer to choose the value of $m = 16$ for further experiments.

We now continue experimenting with the system parameter values $t = 4$ and $m = 16$. For $KEP(16, 4, 200)$, we perform Pearson's chi-squared test for each individual position of $\mathbf{K}_0$ and define the boolean matrix $\mathbf{P} = \{p_{ij}\}$, with its entries

$$p_{ij} = \begin{cases} 0, & \text{if p-value } p\left(\chi^2_{\mathbf{k}_{0ij}}\right) < 0.05, \\ 1, & \text{otherwise,} \end{cases} \tag{6.1}$$

where $\mathbf{k}_{0ij}$ is a 200-element sample of the possible values of the (i, j)-th position of matrix $\mathbf{K}_0$.

We also define the accuracy for matrix entries following the uniform distribution as:

$$ACC(\mathbf{P}) = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} p_{ij}. \tag{6.2}$$

We execute $KEP(16, 4, 200)$ 25 more times and perform the goodness-of-fit tests. The obtained results show that only 1 experiment out of 25 does not follow the uniform distribution for $\mathbf{K}_0$ values. However, the analysis of the imitation key $\mathbf{K}_1$ has also 1 experiment that rejects the null hypothesis. We define the accuracy for $\mathbf{K}_1$ by Eq (6.2) with $p\left(\chi^2_{\mathbf{k}_{0ij}}\right)$ replaced by $p\left(\chi^2_{\mathbf{k}_{1ij}}\right)$ in Eq (6.1).

For more clarity, let us present the results of four executions of $KEP(16, 4, 200)$. We also present the accuracy of uniformity and the two-sample chi-squared test p-value.

We can see from Table 2 that the null hypothesis for the uniform distribution was not rejected for both shared and imitation keys in all of the presented experiments. Moreover, due to the high accuracy value, we can see that for each individual positions the null hypothesis was not rejected for the most part. The takeaway is that looking at matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ as a whole and at the individual positions, we get different results. However, both of these ways are important for statistical analysis, since the

presented results show that individual positions can be treated as independent random variables if the overall uniform distribution is achieved.

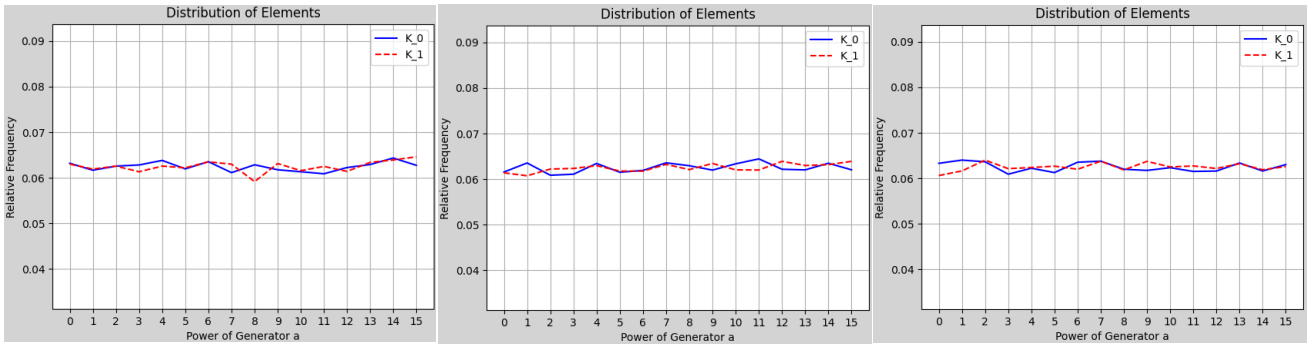
Table 2. Results of four experiments for the platform group $\mathbb{M}_{16}$.

Index	Shared key $\mathbf{K}_0$		Imitation key $\mathbf{K}_1$		Two-sample Chi-Sq.
	Chi-Sq.	ACC	Chi-Sq.	ACC	
1	0.322	0.94	0.506	0.94	0.121
2	0.175	0.94	0.182	0.91	0.470
3	0.055	0.93	0.661	0.95	0.583
4	0.275	0.95	0.547	0.96	0.529

All in all, relying on the p-values for the two-sample chi-squared test presented in Table 2, we can see that there is no significant difference between the distribution of the entries of the shared key $\mathbf{K}_0$ and the imitation key $\mathbf{K}_1$.

Let us now consider the twice larger group $\mathbb{M}_{32}$, i.e., $t = 5$. We execute $KEP(16, 5, 200)$, $KEP(18, 5, 200)$ and $KEP(20, 5, 200)$, and present the obtained results for $\mathbf{K}_0$ and $\mathbf{K}_1$.

We can see from Figure 3 that the presented distributions are almost identical. These observations are also backed up by the statistical tests. In Table 3, we present a short summary of the obtained results.



(a) $KEP(16, 5, 200)$. **(b)** $KEP(18, 5, 200)$. **(c)** $KEP(20, 5, 200)$.

Figure 3. Statistical analysis for uniformity of shared and imitation keys.

Table 3. The statistical analysis results for the platform group $\mathbb{M}_{32}$.

m	$p\left(\chi^2_{\mathbf{K}_0}\right)$	$p\left(\chi^2_{\mathbf{K}_1}\right)$	$p\left(\chi^2_{\mathbf{K}_0,\mathbf{K}_1}\right)$
16	0.030	0.622	0.097
18	0.240	0.803	0.714
20	0.355	0.360	0.569

Relying on the obtained results based on the p-value of the shared key $\mathbf{K}_0$ set, we settle on the matrix size $m = 20$ for further analysis as the p-value 0.355 is the largest of the presented three. Note, however, that a value of $m \geq 18$ may also be used based on the results presented in Table 3. Similarly as before, let us present the results obtained for four executions of $KEP(20, 5, 200)$ in Table 4.

Table 4. Results of four experiments for the platform group $\mathbb{M}_{16}$.

Index	Shared key $\mathbf{K}_0$		Imitation key $\mathbf{K}_1$		Two-sample Chi-Sq.
	Chi-Sq.	ACC	Chi-Sq.	ACC	
1	0.185	0.96	0.844	0.96	0.593
2	0.952	0.96	0.157	0.94	0.978
3	0.478	0.94	0.094	0.95	0.461
4	0.589	0.96	0.126	0.94	0.524

We can see from Table 4 that the obtained results are similar to the ones presented in Table 2. Hence we see that the value of matrix size m does not change drastically as the size of the platform group $\mathbb{M}_{2^t}$ doubles. Therefore, we see that for practical implementation of our proposal, we can settle on a balance of both security parameters to achieve the uniform distribution of both $\mathbf{K}_0$ and $\mathbf{K}_1$ while also keeping them computationally indistinguishable. However, more experiments may be necessary to determine the reasoning behind the chaotic changes of the p-values.

After performing additional experiments for different values of parameters t and m, we present the lower bound of m for each individual key, i.e., for every t, we find a minimal value of m such that the uniformity test is passed for the entries of $\mathbf{K}_0$ and $\mathbf{K}_1$.

The results presented in Table 5 should be interpreted as follows: for a fixed value of t, the order of the square matrices should be no less than the presented value of m for the entries of the considered key to pass the uniformity test. For example, if $t = 7$, then the shared key passes the uniformity test if the matrices are at least 26×26. However, the imitation key passes the uniformity test significantly earlier, i.e., for 16×16 matrices. This shows that, for practical implementation, the pair $(t, m) = (7, 26)$ may be considered a reasonable choice. Of course, the value of m could be rounded up to, say, 30 or 32 depending on the available memory space. Moreover, the linearization technique presented in Section 4.2 should also be kept in mind when choosing the value of m.

Table 5. Lower bounds of matrix order m.

t	4	5	6	7
$\mathbf{K}_0$	16	20	24	26
$\mathbf{K}_1$	14	14	14	16

7. Conclusions and future works

In this paper, we have revisited the previously proposed KEP based on the MPFs defined over the non-commuting platform group. We formalized the definition of the MPF mapping and emphasized its significant difference from the classical matrix multiplication. Furthermore, we have demonstrated that the definition of MPF was previously misinterpreted.

Due to the non-commuting platform group, the discrete logarithm mapping cannot be applied either directly or indirectly to the system (4.4) and hence the algebraic analysis of the decisional problem becomes more complex. In fact, it was shown in [16] that this problem is NP-complete. Therefore, there is some indication that our proposal can withstand quantum attacks.

We have also shown that neither linearization nor faithful matrix representation can be used to obtain any kind of advantage in winning the Security Game 3 if the parameters of our KEP are appropriately

chosen. However, we think that the linearization technique should be kept in mind when generating the publicly known matrices $\mathbf{W}, \mathbf{L}$, and $\mathbf{R}$. As of now, we do not know if there are any other approaches the attacker can use to achieve a significant advantage. This is grounds for future work in the security of our protocol.

The proposed protocol is anonymous and hence is vulnerable to man-in-the-middle attack. This can be fixed by adding an identification step. We can consider this in the future. Also, we intend to study if the proposed protocol provides a forward secrecy property.

Statistical analysis of Security Game 3 has shown that there is a dependence between the group-defining parameter t and the matrix size m. However, we see that the value of m does not change drastically as the platform group $\mathbb{M}_{2^t}$ doubles in size. On the other hand, the results presented in Table 5 show that the value of m to pass the uniformity test for the imitation key $\mathbf{K}_1$ is smaller than in the case of the true key $\mathbf{K}_0$. Furthermore, based on the results presented in [29], we can see that both matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ look random to an attacker. Therefore, the statistical analysis in no way can help the attacker to distinguish between the shared and imitation keys.

Based on the obtained results, we claim that it is safe to use the proposed KEP in internet communications providing a secure shared key for two protocol parties. This key is used to ensure confidentiality using secure symmetric encryption methods, e.g., AES.

Author contributions

Aleksejus Mihalkovich: conceptualization, formal analysis, investigation, methodology, project administration, supervision, validation, writing-original draft, writing-review and editing; Jokubas Zitkevicius: data curation, investigation, software, visualization, writing-original draft; Eligijus Sakalauskas: formal analysis, methodology, supervision, validation.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare no conflict of interest.

References

1. W. Diffie, M. Hellman, New directions in cryptography, *IEEE T. Inform. Theory*, **22** (1976), 644–654. http://dx.doi.org/10.1109/TIT.1976.1055638
2. D. Boneh, V. Shoup, *A graduate course in applied cryptography*, 0.6 Eds., 2023, unpublished work. Available from: `https://toc.cryptobook.us/book.pdf`.
3. E. Bresson, Y. Lakhnech, L. Mazaré, B. Warinschi, A generalization of DDH with applications to protocol analysis and computational soundness, In: *Advances in cryptology-CRYPTO 2007*, Berlin: Springer, 2007, 482–499. http://dx.doi.org/10.1007/978-3-540-74143-5_27

4. R. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Commun. ACM*, **21** (1978), 120–126. http://dx.doi.org/10.1145/359340.359342
5. P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Rev.*, **41** (1999), 303–332. http://dx.doi.org/10.1137/S0036144598347011
6. *Submission requirements and evaluation criteria for the post-quantum cryptography standardization process*, NIST Computer Security Resource Center, 2016. Available from: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.
7. *Post-quantum cryptography, selected algorithms 2022*, NIST Computer Security Resource Center, 2022. Available from: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.
8. *Post-quantum cryptography, round 4 submissions*, NIST Computer Security Resource Center, 2022. Available from: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions.
9. K. Ko, S. Lee, J. Cheon, J. Han, J. Kang, C. Park, New public-key cryptosystem using braid groups, In: *Advances in cryptology-CRYPTO 2000*, Berlin: Springer, 2000, 166–183. http://dx.doi.org/10.1007/3-540-44598-6_10
10. V. Shpilrain, A. Ushakov, The conjugacy search problem in public key cryptography: unnecessary and insufficient, *AAECC*, **17** (2006), 285–289. http://dx.doi.org/10.1007/s00200-006-0009-6
11. E. Sakalauskas, N. Listopadskis, P. Tvarijonas, Key agreement protocol (KAP) based on matrix power function, *Proceedings of Sixth International Conference on Information Research and Applications*, 2008, 92–96.
12. A. Mihalkovich, E. Sakalauskas, Asymmetric cipher based on MPF and its security parameters evaluation, *Lietuvos Matematikos Rinkinys*, **53** (2012), 72–77. http://dx.doi.org/10.15388/LMR.A.2012.13
13. J. Liu, H. Zhang, J. Jia, A linear algebra attack on the non-commuting cryptography class based on matrix power function, In: *Information security and cryptology*, Cham: Springer, 2017, 343–354. http://dx.doi.org/10.1007/978-3-319-54705-3_21
14. E. Sakalauskas, A. Mihalkovich, Improved asymmetric cipher based on matrix power function resistant to linear algebra attack, *Informatica*, **28** (2017), 517–524. http://dx.doi.org/10.15388/Informatica.2017.142
15. A. Mihalkovich, M. Levinskas, Investigation of matrix power asymmetric cipher resistant to linear algebra attack, In: *Information and software technologies*, Cham: Springer, 2019, 197–208. http://dx.doi.org/10.1007/978-3-030-30275-7_16
16. A. Mihalkovich, E. Sakalauskas, K. Luksys, Key exchange protocol defined over a non-commuting group based on an NP-complete decisional problem, *Symmetry*, **12** (2020), 1389. http://dx.doi.org/10.3390/sym12091389
17. A. Mihalkovich, K. Luksys, E. Sakalauskas, Sigma identification protocol construction based on MPF defined over non-commuting platform group, *Mathematics*, **10** (2022), 2649. http://dx.doi.org/10.3390/math10152649

18. M. Durcheva, TrES: tropical encryption scheme based on double key exchange, *European Journal of Information Technologies and Computer Science*, **2** (2022), 11–17. http://dx.doi.org/10.24018/compute.2022.2.4.70

19. X. Jiang, H. Huang, G. Pan, Cryptanalysis of tropical encryption scheme based on double key exchange, *Journal of Cyber Security and Mobility*, **12** (2023), 205–220. http://dx.doi.org/10.13052/jcsm2245-1439.1224

20. *Modular maximal-cyclic group*, Groupprops, 2023, Available from: `https://groupprops.subwiki.org/wiki/Modular_maximal-cyclic_group`.

21. H. Grundman, T. Smith, Automatic realizability of Galois groups of order 16, *Proc. Amer. Math. Soc.*, **124** (1996), 2631–2640. http://dx.doi.org/10.1090/S0002-9939-96-03345-X

22. H. Grundman, T. Smith, Realizability and automatic realizability of Galois groups of order 32, *Open Math.*, **8** (2010), 244–260. https://doi.org/10.2478/s11533-009-0072-x

23. H. Grundman, T. Smith, Galois realizability of groups of order 64, *Centr. Eur. J. Math.*, **8** (2010), 846–854. http://dx.doi.org/10.2478/s11533-010-0052-1

24. A. Mihalkovich, E. Sakalauskas, M. Levinskas, Key exchange protocol based on the matrix power function defined over $\mathbb{M}_{16}$, In: *Intelligent computing*, Cham: Springer, 2022, 511–531. http://dx.doi.org/10.1007/978-3-031-10467-1_32

25. *Faithful irreducible representation of M16*, Groupprops, 2023, Available from: `https://groupprops.subwiki.org/wiki/Faithful_irreducible_representation_of_M16`.

26. J. Faugère, A. Joux, Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases, In: *Advances in cryptology-CRYPTO 2003*, Berlin: Springer, 2003, 44–60. http://dx.doi.org/10.1007/978-3-540-45146-4_3

27. A. Kipnis, J. Patarin, L. Goubin, Unbalanced oil and vinegar signature schemes, *Advances in Cryptology-EUROCRYPT'99*, Berlin: Springer, 1999, 206–222. http://dx.doi.org/10.1007/3-540-48910-X_15

28. R. Benadjila, T. Feneuil, M. Rivain, MQ on my mind: post-quantum signatures from the non-structured multivariate quadratic problem, *Proceedings of IEEE 9th European Symposium on Security and Privacy*, 2024, 468–485. http://dx.doi.org/10.1109/EuroSP60621.2024.00032

29. A. Mihalkovich, J. Zitkevicius, On the decisional problem based on matrix power function defined over non-commutative group, *Mathematical Models in Engineering*, **10** (2024), 1–9. http://dx.doi.org/10.21595/mme.2024.24071

AIMS Press