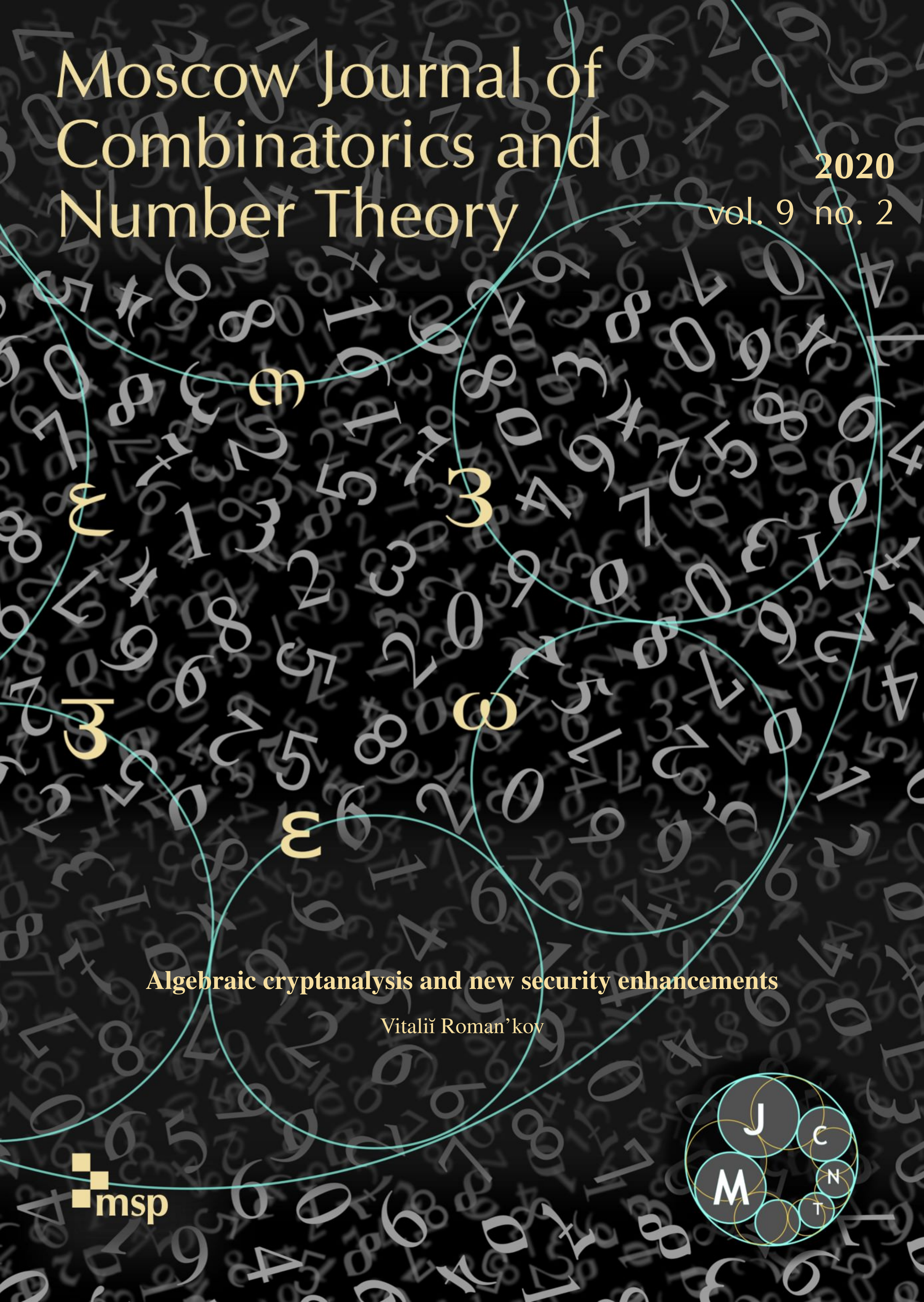

Moscow Journal of Combinatorics and Number Theory

2020
vol. 9 no. 2

Algebraic cryptanalysis and new security enhancements

Vitaliĭ Roman’kov

msp

Moscow Journal of Combinatorics and Number Theory
Vol. 9, No. 2, 2020
https://doi.org/10.2140/moscow.2020.9.123

Algebraic cryptanalysis and new security enhancements

Vitaliĭ Roman'kov

We briefly discuss linear decomposition and nonlinear decomposition attacks using polynomial-time deterministic algorithms that recover the secret shared keys from public data in many schemes of algebraic cryptography. We show that in this case, contrary to common opinion, typical computational security assumptions are not very relevant to the security of the schemes; i.e., one can break the schemes without solving the algorithmic problems on which the assumptions are based. Also we present another and in some points similar approach, which was established by Tsaban et al.

Before demonstrating the applicability of these two methods to two well-known noncommutative protocols, we cryptanalyze two new cryptographic schemes that have not yet been analyzed.

Further, we introduce a novel method of construction of systems resistant against attacks via linear algebra. In particular, we propose improved versions of the well-known Diffie–Hellman-type (DH) and Anshel–Anshel–Goldfeld (AAG) algebraic cryptographic key-exchange protocols.

1. Introduction

In [Roman'kov 2013a], the author introduced a method of *linear decomposition* applicable in algebraic cryptanalysis. This method was further developed in [Myasnikov and Roman'kov 2015]; see also [Roman'kov 2013b; 2018a; 2018b]. In [Roman'kov 2016], this method was supplemented by a *nonlinear decomposition* method; see also [Roman'kov 2018b]. These methods can be applied for obtaining secret keys without computing private parameters or solving algorithmic problems on which the protocols are based. These applications are called *linear* and *nonlinear decomposition attacks* respectively. They are deterministic, provable and polynomial-time. These methods were widely applied in cryptanalysis of dozens of protocols of algebraic cryptography; see [Roman'kov 2018b]. The linear decomposition attack can be applied to protocols based on matrix groups over arbitrary (finite or infinite) fields. The nonlinear decomposition attack is applicable to protocols based on groups that are not necessary matrix, or do not use matrix representations. See details in [Roman'kov 2016; 2018b].

B. Tsaban [2015] introduced a method for obtaining provable polynomial-time solutions of problems in noncommutative algebraic cryptography called the *linear span-method*, or simply the *span-method*; see also [Ben-Zvi et al. 2018]. This method is probabilistic and is a fundamental base for algebraic span cryptanalysis, a general approach for provable polynomial-time solutions of computational problems in groups of matrices over finite fields, and thus in all groups with efficient matrix representations over finite fields. This approach is widely applicable; in particular, it is applicable to the AAG protocol. Algebraic

This work was supported by the Mathematical Center in Akademgorodok under agreement no. 075-15-2019-1613 with the Ministry of Science and Higher Education of the Russian Federation.

MSC2010: primary 20F10; secondary 20F70, 94A60.

Keywords: postquantum cryptography, algebraic cryptanalysis, algebraic cryptography, marginal sets.

span cryptanalysis improves upon earlier approaches, such as Cheon–Jun's method [2003] and Tsaban's linear centralizer method [2015].

We will not describe these methods in detail, but we will give a couple of examples of how these methods can be applied. Some of these applications, namely to the DH and to the AAG protocols, were previously presented in the literature. We present them here because we propose improved versions of them. There are exactly two new applications: one of them to cryptanalysis of the ElGamal-type version of the cryptosystem MOR introduced in [Bhunia et al. 2019], and the other to the cryptosystem proposed in [Baba et al. 2011].

A different probabilistic attack on the braid group cryptosystems is the length-based attack. The length-based attack on AAG protocol was initially proposed by J. Hughes and A. Tannenbaum [2002]. A. D. Myasnikov and A. Ushakov [2007] showed that accurately designed length-based attack can successfully break a random instance of the simultaneous conjugacy search problem for certain parameter values and argued that the public/private information chosen uniformly random leads to weak keys. This attack can be applied to other groups too. See [Garber et al. 2006; Hofheinz and Steinwandt 2002; Hughes 2002; Myasnikov et al. 2005; 2006].

The presence of effective methods of linear algebra in algebraic cryptanalysis requires the development of tools to counter these methods. Section 7 presents such tools. Their use makes some well-known schemes protected against attacks by the linear algebra methods. As examples of such protection, we provide improved versions of the DH and AAG algebraic cryptographic key-exchange protocols.

Throughout we use the following notation:

- $\mathbb{Z}$, the set of integer numbers.
- $\mathbb{N}$, the set of nonnegative integer numbers.
- $\mathbb{S}_n$, the symmetric group of degree n.
- $g^h = hgh^{-1}$, conjugate.
- $[g, h] = ghg^{-1}h^{-1}$, commutator.

For a group G, we have:

- G', commutant (derived subgroup).
- $\mathrm{C}_G(A)$, centralizer of A in G.
- $\mathrm{Aut}(G)$, automorphism group.

2. Mathematical background for the linear algebra methods

Let $\mathbb{F}$ be a field and $\mathrm{M}(n, \mathbb{F})$ be the set of $n \times n$ matrices with entries in $\mathbb{F}$. For a set $S \subseteq \mathrm{M}(n, \mathbb{F})$, let $\mathrm{Alg}(S)$ be the algebra generated by S, that is, the smallest algebra $A \subseteq \mathrm{M}(n, \mathbb{F})$ that contains S as a subset. Every subalgebra of $\mathrm{M}(n, \mathbb{F})$ is also a vector space over the field $\mathbb{F}$. Let $\mathrm{GL}(n, \mathbb{F})$ be the group of invertible matrices in $\mathrm{M}(n, \mathbb{F})$. For a subgroup $G \leq \mathrm{GL}(n, \mathbb{F})$, we have $\mathrm{Alg}(G) = \mathrm{span}(G)$, where $\mathrm{span}(G)$ is the vector space spanned by G.

Proposition 2.1 [Ben-Zvi et al. 2018, Proposition 1]. *Let $G = \mathrm{gp}(g_1, \ldots, g_k) \leq \mathrm{GL}(n, \mathbb{F})$ be a group, and $d \leq n^2$ be the dimension of the vector space $\mathrm{Alg}(G)$. A basis for the vector space $\mathrm{Alg}(G)$ can be computed using $O(kd^2n^2)$ field operations.*

Lemma 2.2 (invertibility lemma [Tsaban 2015, Lemma 9]). *For a finite field $\mathbb{F}_q$ of order q, let $h_1, \dots, h_m \in \mathrm{M}(n, \mathbb{F}_q)$ such that some linear combination of these matrices is invertible. If $\alpha_1, \dots, \alpha_m$ are chosen uniformly and independently from $\mathbb{F}_q$, then the probability that the linear combination $\alpha_1 h_1 + \cdots + \alpha_m h_m$ is invertible is at least $1 - n/q$.*

Let V be a finite-dimensional vector space over a field $\mathbb{F}$ with basis $\mathcal{B} = \{v_1, \dots, v_r\}$. Let $\mathrm{End}(V)$ be the semigroup of endomorphisms of V. We assume that elements $v \in V$ are given as vectors relative to $\mathcal{B}$, and endomorphisms $a \in \mathrm{End}(V)$ are given by their matrices relative to $\mathcal{B}$. For an endomorphism $a \in \mathrm{End}(V)$ and an element $v \in V$ we denote by v^a the image of v under a. Also, for any subsets $W \subseteq V$ and $A \subseteq \mathrm{End}(V)$ we put $W^A = \{w^a : w \in W, a \in A\}$. We assume that elements of the field $\mathbb{F}$ are given in some constructive form and the "size" of the form is defined. Furthermore, we assume that the basic field operations in $\mathbb{F}$ are efficient; in particular they can be performed in polynomial time in the size of the elements. In other words, F is *constructive*. For an element $\alpha \in \mathbb{F}$ we write $|\alpha|$ for the size of α and put $|v| = \max\{|\alpha_i|\}$ for a vector $v = (\alpha_1, \dots, \alpha_r) \in V$, and $|a| = \max\{|\alpha_{ij}|\}$ for a matrix $a = (\alpha_{ij}) \in \mathrm{End}(V)$.

Lemma 2.3 (principal lemma [Myasnikov and Roman'kov 2015, Lemma 3.1]). *There is an algorithm that for given finite subsets $W \subseteq V$ and $U \subseteq \mathrm{End}(V)$ finds a basis of the subspace* $\mathrm{span}(W^{\mathrm{sm}}(U))$ *in the form $\{w_1^{a_1}, \dots, w_t^{a_t}\}$, where $w_i \in W$ and $a_i \in \mathrm{sm}(U)$. Here* $\mathrm{sm}(U)$ *denotes the submonoid generated by U. Furthermore, the number of field operations used by the algorithm is polynomial in $r = \mathrm{dim}(V)$ and the cardinalities of W and U. The total estimate is $O(r^3|U|^2 + r|W|^2)$.*

3. Cryptanalysis of two schemes of Baba et al. by the linear algebra methods

In [Baba et al. 2011], S. Baba, S. Kotyada and R. Teja demonstrated how to define a supposedly one-way function FACTOR in a noncommutative group. As an example of a platform for implementing FACTOR, they proposed one of the groups, such as $\mathrm{GL}(n, \mathbb{F}_q)$, $\mathrm{UT}(n, \mathbb{F}_q)$ or braid groups B_n, $n \in \mathbb{N}$. Here $\mathbb{F}_q$ denotes a finite field of order q.

They believed that the function FACTOR was one-way, which means that the inverse to FACTOR is easy to compute, while the function itself is hard to compute. Shortly afterwards Stanek [2011] published an extension of the baby-step giant-step algorithm disproving this conjecture. Note that the baby-step giant-step methods are limited in practice because of memory requirements. In [Romsy 2011] a modification of Pollard's kangaroo algorithm was presented that solves the FACTOR problem requiring only negligible memory. Anyway these methods have very complicated implementations. We will show that the linear algebra approach is much simpler and more efficient. At the same time, this will be an example of using the methods presented.

Then, using the FACTOR function as a primitive, the authors of [Baba et al. 2011] defined a public key cryptosystem which is comparable to the classical ElGamal system based on the discrete logarithm problem. Recall, that the ElGamal system can be described as follows: Let G be a public finite cyclic group with generator g, and let $x \in \mathbb{Z}$ be Alice's private key. The element g^x is public. To send a message $m \in G$, Bob picks a random integer y and sends the ciphertext $c = (g^y, g^{xy}m)$ to Alice. To decrypt, Alice calculates $(g^y)^x = g^{xy}$ and inverts it to retrieve m. There are a couple of cryptosystems of ElGamal-type. See, for example, [Kahrobaei and Khan 2006; Fine et al. 2016]. The versions proposed in [Mahalanobis 2008; 2012] were analyzed in [Roman'kov and Obzor 2018]. See also cryptanalysis in [Roman'kov 2018b].

In [Baba et al. 2011], the authors also proposed a key exchange, analogous to the DH key exchange protocol in a noncommutative setting using FACTOR. Recall, that the classical DH protocol can be described as follows: Let G be a public finite cyclic group with generator g, and let $x \in \mathbb{Z}$ be Alice's private key and $y \in \mathbb{Z}$ be Bob's private key. Alice publishes g^x and Bob publishes g^y. Then each of them computes the exchanged key $g^{xy} = (g^x)^y = (g^y)^x$.

In this paper, we apply and compare two methods of algebraic cryptanalysis via linear algebra, namely, the linear decomposition method invented and developed by the author in [Roman'kov 2013a; 2013b; 2018b] and in [Myasnikov and Roman'kov 2015], and the span-method invented by B. Tsaban and developed with A. Ben-Zvi, and A. Kalka [Tsaban 2015; Ben-Zvi et al. 2018] to show the vulnerability of the cryptosystem and protocol proposed in [Baba et al. 2011].

3A. ***The ElGamal-type cryptosystem based on FACTOR [Baba et al. 2011].*** Let G be any group and let $g, h \in G$ be two noncommuting elements chosen by Alice. Let $\mathrm{gp}(g)$ and $\mathrm{gp}(h)$ be the cyclic subgroups generated by these elements, respectively. In order to define the FACTOR function one assume that $\mathrm{gp}(g) \cap \mathrm{gp}(h) = \{1\}$. Let $\varphi : \mathrm{gp}(g) \times \mathrm{gp}(h) \to G$ be a function defined by $\varphi(g^x, h^y) = g^x \cdot h^y$, where $x, y \in \mathbb{Z}$. Obviously, φ is injective. Then $\mathrm{FACTOR}(g^x h^y) = \varphi^{-1}(g^x h^y)$.

We suppose that Alice is the recipient of the messages and Bob is communicating with Alice. Let $m \in G$ be a message.

Algorithm.

- Alice picks arbitrary random integers $x, y \in \mathbb{Z}$ and sets a public key $(G, g, h, g^x h^y)$. Alice has a private key (g^x, h^y) for decryption.
- To send m, Bob picks arbitrary random private integers x', y' and sends the ciphertext
$$c = (g^{x+x'} h^{y+y'}, g^{x'} h^{y'} m)$$
to Alice.
- To decrypt the ciphertext, Alice uses her private key and calculates
$$(g^x)^{-1}(g^{x+x'} h^{y+y'})(h^y)^{-1} = g^{x'} h^{y'}.$$
Then she inverts it to retrieve m.

The authors of this scheme hoped that the security of the cryptosystem described above reduces to solving FACTOR problem in the underlying group. Below we will show that the system is vulnerable to linear algebra attacks.

3B. ***Cryptanalysis of the ElGamal-type cryptosystem based on FACTOR.*** We will show that any intruder can efficiently retrieve m.

First we will use the span-method.

Theorem 3.1. *Suppose that G is a finite group presented as a matrix group over a finite field $\mathbb{F}_q$ of order q; i.e., $G \leq \mathrm{M}(n, \mathbb{F}_q)$. Let $g, h \in G$ be two noncommuting elements such that $\mathrm{gp}(g) \cap \mathrm{gp}(h) = \{1\}$. Given $g^x h^y$, $g^{x+x'} h^{y+y'} \in G$, where $x, x', y, y' \in \mathbb{N}$, one can find in polynomial time (in the size of the public data) the element $g^{x'} h^{y'}$.*

Proof. Let $V = \mathrm{span}(\mathrm{gp}(g))$ be the linear subspace of $\mathrm{M}(n, \mathbb{F}_q)$ generated by all matrices of the form g^i, $i \in \mathbb{Z}$. Then $\dim(V) \leq n - 1$. Since g is the root of its characteristic polynomial of degree n, matrices $1, g, g^2, \ldots, g^n$ are linearly dependent. Obviously, if g^{k+1} lies in $\mathrm{span}(\{1, g, g^2, \ldots, g^k\})$, then $g^{k+t}, g^{1-t} \in \mathrm{span}(\{1, g^2, \ldots, g^k\})$ for every $t = 2, 3, \ldots$.

By the Gaussian elimination method, we can efficiently construct a basis for V. For example, we can take as a basis the maximum independent set of elements of the form $1, g, g^2, \ldots, g^k$, checking for each subsequent $l = 0, 1, \ldots$ whether or not g^{l+1} lies in $\mathrm{span}(\{1, g, g^2, \ldots, g^l\})$.

Consider the equation

$$f(g^x h^y)h = hf(g^x h^y) \sim fg^x h = hfg^x, \tag{1}$$

which is linear with respect to n^2 unknown entries of matrix f. We will seek f in the form

$$f = \sum_{i=0}^{k} \alpha_i g^i;$$

i.e., we seek a solution f in V. We know that there is a nondegenerate solution $f = g^{-x}$. By Proposition 2.1 we can efficiently construct a basis $e_1, \ldots, e_p$ of the subspace of all solutions of (1) in V. Then we apply the invertibility lemma, Lemma 2.2 to find an invertible solution f.

Let the element f be found. Then

$$f(g^{x+x'} h^{y+y'}) = g^{x'}(f(g^x h^{y'}))h^y = (g^{x'} h^{y'})f(g^x h^y)$$

and

$$(g^{x'} h^{y'})f(g^x h^y)(g^x h^y)^{-1} f^{-1} = g^{x'} h^{y'}. \qquad \square$$

Now we apply the result just obtained to the protocol under consideration.

Corollary 3.2. *We have*

$$(g^{x'} h^{y'})^{-1}(g^{x'} h^{y'} m) = m,$$

and the message m is thus computed.

Now we will show how, using the linear decomposition method, we can calculate the message m for an arbitrary constructive field by a deterministic algorithm.

Theorem 3.3. *Let $G \leq \mathrm{M}(n, \mathbb{F})$ be a matrix group over an arbitrary* (*constructive*) *field $\mathbb{F}$. Let $g, h \in G$ be two noncommuting elements such that $\mathrm{gp}(g) \cap \mathrm{gp}(h) = \{1\}$ and $m \in G$. Given the elements $g^x h^y$, $g^{x+x'} h^{y+y'}$, $g^{x+x'} h^{y+y'} m \in G$, where $x, x', y, y' \in \mathbb{Z}$, one can find in polynomial time* (*in the size of the public data*) *the element m.*

Proof. Let $V = \mathrm{span}(\mathrm{gp}(g) g^x h^y \, \mathrm{gp}(h))$ be the linear subspace of $\mathrm{M}(n, \mathbb{F})$ generated by all matrices of the form $g^i(g^x h^y)h^j$, $i, j \in \mathbb{Z}$. Then $\dim(V) \leq (n-1)^2$. In the notation of Lemma 2.3, $V = W^{\mathrm{sm}}(U)$, $W = \{g^x h^y\}$, $U = \mathrm{sm}(l(g^{\pm 1}, r(h^{\pm 1}))$, where for any $f \in G$, $l(f)$ means the endomorphism of $\mathrm{M}(n, \mathbb{F})$ corresponding to left-sided multiplication by f. Similarly $r(f)$ means the endomorphism of $\mathrm{M}(n, \mathbb{F})$ corresponding to right-sided multiplication by f.

By Lemma 2.3, we can efficiently obtain a basis $e_i = g^{u_i}(g^x h^y)h^{v_i}$, $u_i, v_i \in \mathbb{Z}$, $i = 1, \ldots, r$ of V.

Since, $g^{x+x'}h^{y+y'} \in V$, by Lemma 2.3 we can efficiently obtain an expression of the form

$$g^{x+x'}h^{y+y'} = \sum_{i=1}^{r} \alpha_i e_i, \quad \alpha_i \in \mathbb{F},\ i = 1, \dots, r. \tag{2}$$

The right side of (2) is equal to

$$g^x \Bigl(\sum_{i=1}^{r} \alpha_i g^{u_i} h^{v_i} \Bigr) h^y, \tag{3}$$

and it follows by (2) that

$$g^{x'}h^{y'} = \sum_{i=1}^{r} \alpha_i g^{u_i} h^{v_i}. \tag{4}$$

The message m is retrieved as above. □

Remark 3.4. Recall that the authors of [Baba et al. 2011] suggest as a platform for their cryptosystem one of the groups $\mathrm{GL}(n, \mathbb{F}_q)$, $\mathrm{UT}(n, \mathbb{F}_q)$, or braid groups B_n, $n \in \mathbb{N}$. In our cryptanalysis, we consider only matrix groups. Any group B_n admits a faithful matrix representation [Bigelow 2001; Krammer 2002]. The braid group B_n is linear via the so-called Lawrence–Krammer (LK) representation $B_n \to \mathrm{GL}(m, \mathbb{Z}[t^{\pm 1}, 1/2])$, where $m = n(n-1)/2$, which is injective. The LK representation can be computed by a polynomial-time algorithm. This representation is also invertible by (similar) polynomial-time algorithm; see [Krammer 2002; Cheon and Jun 2003].

3C. ***The DH key exchange protocol based on FACTOR [Baba et al. 2011] as a particular case of the protocol in [Sidelnikov et al. 1993].*** Suppose Alice and Bob want to exchange keys. Suppose G, g, h are as in Section 3A.

Algorithm 1.

- Alice chooses a random pair of integers (x_1, y_1). Then Alice sends the element $g^{x_1}h^{y_1}$ to Bob.
- Bob picks up two random integers (x_2, y_2). Then Bob sends the element $g^{x_2}h^{y_2}$ to Alice.
- Alice computes $K_A = g^{x_1}(g^{x_2}h^{y_2})h^{y_1} = g^{x_1+x_2}h^{y_1+y_2}$.
- Bob computes $K_B = g^{x_2}(g^{x_1}h^{y_1})h^{y_2} = g^{x_1+x_2}h^{y_1+y_2}$.
- Now Alice and Bob have their exchanged secret key $K_1 = K_A = K_B$.

This algorithm is a particular case of the following algorithm of [Sidelnikov et al. 1993].

Let G be a group, A and B two of its commutative subgroups, and $g \in G$. This data is public.

Algorithm 2.

- Alice chooses a random pair of elements $(a, b) \in A \times B$. Then Alice sends the element agb to Bob.
- Bob picks up two random elements $(a', b') \in A \times B$. Then Bob sends the element $a'gb'$ to Alice.
- Alice computes $K_A = aa'gb'b$.
- Bob computes $K_B = a'agbb'$.
- Now Alice and Bob have their exchanged secret key $K_2 = K_A = K_B$.

3D. ***Cryptanalysis the DH key exchange protocols presented above.*** Now we will apply the linear decomposition method to reveal K.

Theorem 3.5. *Let $G \leq \mathrm{M}(n, \mathbb{F})$ be a matrix group over an arbitrary constructive field $\mathbb{F}$. Let $g \in G$ and let $A = \mathrm{gp}(a_1, \dots, a_m)$, $B = \mathrm{gp}(b_1, \dots, b_s)$ be two finitely generated subgroups of G. Given $agb, a'gb'$, where $a, a' \in A,\ b, b' \in B$, one can find in polynomial time (in the size of the public data) the element $aa'gbb'$.*

Proof. Let $V = \mathrm{span}(AgB)$ be the linear subspace of $\mathrm{M}(n, \mathbb{F})$ generated by all matrices of the form ugv, $u \in A,\ v \in B$. Then $\dim(V) \leq (n-1)^2$.

In the notation of Lemma 2.3, $V = W^{\mathrm{sm}}(U)$, where $W = \{g\},\ U = \mathrm{sm}(l(a_i^{\pm 1}), r(b_j \pm 1)),\ i = 1, \dots, m$, $j = 1, \dots, s$. Let $e_i = u_i g v_i\ \ i = 1, \dots, r$, be a basis of V that can be efficiently obtained by Lemma 2.3

Since, $agb \in V$, we can efficiently obtain an expression of the form

$$agb = \sum_{i=1}^{r} \alpha_i e_i, \quad \alpha_i \in \mathbb{F},\ i = 1, \dots, r. \tag{5}$$

Then

$$\sum_{i=1}^{r} \alpha_i u_i (a'gb') v_i = a' \left(\sum_{i=1}^{r} \alpha_i e_i \right) b' = aa'gbb', \tag{6}$$

completing the proof. □

Corollary 3.6. *Each of the keys K_1 and K_2 of Algorithms 1 and 2 can be efficiently calculated in polynomial time (from the size of the public data of the algorithms).*

The described cryptanalysis has many analogues, presented in [Roman'kov 2013a; 2013b; 2016; 2018a; 2018b; 2019a; Ben-Zvi et al. 2018; Tsaban 2015]. In [Roman'kov 2018a], a general scheme based on multiplications is presented. It corresponds to a number of cryptographic systems known in the literature, which are also vulnerable to attacks by the linear decomposition method. Note that Tsaban's span-method allows him to show the vulnerability of the well-known schemes of [Anshel et al. 1999], and the triple decomposition key exchange protocol of [Peker 2014].

4. Cryptanalysis of a new version of the MOR scheme

S. Bhunia, A. Mahalanobis, P. Shinde and A. Singh [Bhunia et al. 2019] studied the ElGamal-type version of the MOR cryptosystem with symplectic and orthogonal groups over finite fields $\mathbb{F}_q$ of odd characteristics. The MOR cryptosystem over $\mathrm{SL}(d, \mathbb{F}_q)$ was previously investigated by the second of these authors. In that case, the hardness of the MOR cryptosystem was found to be equivalent to the discrete logarithm problem in F_{q^d}. It is shown in [Bhunia et al. 2019] that the MOR cryptosystem over $\mathrm{Sp}(d, q)$ has the security of the discrete logarithm problem in $\mathbb{F}_{q^d}$. The MOR cryptosystem was also studied in [Paeng et al. 2001; Mahalanobis 2015] and was cryptanalyzed in [Monico 2016].

We are to show that the version of MOR in [Bhunia et al. 2019] is not entirely accurate. It should be supplemented with an additional assumption. The equivalence theorem there should be clarified too.

We also show that the proposed ElGamal-type version of MOR over any finitely generated matrix group $G \leq \mathrm{GL}(d, \mathbb{F}_q)$ is vulnerable with respect to the linear decomposition attack in any case when the automorphism φ can be naturally extended to a linear transformation of the linear space $\mathrm{span}(G)$

generated by G in $\mathrm{M}(d, \mathbb{F})$, for example, if φ is an inner automorphism. In fact, there exists an efficient algorithm to compute the original message by its ciphertext. It can be done for every constructive field, i.e., a field for which all operations are efficient, and the Gaussian elimination process is efficient too.

4A. ***The ElGamal version of the MOR cryptosystem [Bhunia et al. 2019].*** Let $G = \mathrm{gp}(g_1, g_2, \dots, g_n)$ be a (finite) public group and φ a nontrivial public automorphism of G.

Alice's keys are as follows:

- Private key: $t \in \mathbb{N}$.
- Public key: $\{\varphi(g_i) : i = 1, \dots, n\}$ and $\{\varphi^t(g_i) : i = 1, \dots, n\}$.

We suppose that Alice is the recipient of the messages and Bob is communicating with Alice. Let $m \in G$ be a message.

Algorithm.

- To send the message (plaintext) m Bob picks up a random integer r, then he computes $\{\varphi^r(g_i) : i = 1, \dots, n\}$ and $\varphi^{tr}(m)$. The ciphertext is $(\{\varphi^r(g_i) : i = 1, \dots, n\}, \varphi^{tr}(m))$.
- Since Alice knows t, she computes $\varphi^{tr}(g_i)$ from $\varphi^r(g_i)$ and then $\varphi^{-tr}(g_i),\ i = 1, \dots, n$. Finally, the message m can be computed by $\varphi^{-tr}(\varphi^{tr}(m)) = m$.

Remark 4.1. There is one obstacle to the implementation of the decryption process. To recover m, Alice should compute $\{\varphi^{-tr}(g_i) : i = 1, , n\}$ by $\{\varphi^{tr}(g_i) : i = 1, \dots, n\}$ or compute it by φ^r. It can be done if she knows φ^{-1}, i.e., $\{\varphi^{-1}(g_i) : i = 1, \dots, n\}$.

In the general case, the calculation of the inverse automorphism is not an obviously efficient process. We have to assume that Alice can do it, for example, because she knows $s \in \mathbb{N}$ such that $\varphi^s = \mathrm{id}$. It happens, in particular, if she knows the order s_1 of φ or the order s_2 of $\mathrm{Aut}(G)$. Then $\varphi^{-1} = \varphi^{s-1}$ ($s = s_1$ or $s = s_2$). Also Alice can know φ^{-1}.

Alice can simultaneously build φ and φ^{-1} during the setting of parameters of the protocol.

This obstacle manifests itself more significantly in the proof of the following theorem.

Theorem [Bhunia et al. 2019, Theorem 2.1]. *The difficulty in breaking the above MOR cryptosystem is equivalent to the DH problem in the group* $\mathrm{gp}(\varphi)$.

Proof. It is easy to see that if one can break the DH problem, then one can compute φ^{tr} from φ^t in the public key and φ^r in the ciphertext. This breaks the system.

On the other hand, observe that the plaintext is $m = \varphi^{-tr}(\varphi^{tr}(m))$. Assume that there is an oracle that can break the MOR cryptosystem, i.e., given φ, φ^t and a ciphertext (φ^r, f) will deliver $\varphi^{-tr}(f)$. Now we query the oracle n times with the public key and the ciphertexts $(\varphi^r(g_i), g_i)$ for $i = 1, \dots, n$. From the output, one can easily find $\varphi^{-tr}(g_i)$ for $i = 1, 2, \dots, n$. So we just witnessed that for $\varphi^r(g_i)$ and $\varphi^t(g_i)$ for $i = 1, \dots, n$, one can compute $\varphi^{-tr}(f)$ for every $f = f(g_1, \dots, g_n)$ using the oracle. This solves the DH problem. □

Remark 4.2. In the first part of the proof one computes φ^{tr}, but one needs φ^{-tr} to compute m in the protocol. However, it is not always easy to find the inverse. There are some cryptographic schemes based on the complexity of the problem of finding the inverse to a given automorphism.

4B. ***Cryptanalysis of the ElGamal version of the MOR cryptosystem.*** We propose the following cryptanalysis that works in the case of an arbitrary (constructive) field $\mathbb{F}$.

Suppose that the ElGamal-type system MOR is considered over a finitely generated matrix group $G \leq \mathrm{GL}(d, \mathbb{F})$. Then $G \subseteq \mathrm{M}(d, \mathbb{F})$. Let $G = \mathrm{gp}(g_1, \ldots, g_n)$. We suppose that φ can be naturally extended to a linear transformation of $V = \mathrm{span}(G)$ that is a linear subspace generated by G in $\mathrm{M}(d, \mathbb{F})$. It happens for example, if φ is an inner automorphism of G. Note, that the case of inner automorphism φ is considered in [Bhunia et al. 2019] as the most significant.

To reveal m using only open protocol data, we perform the following actions.

Step 1: Let V_i, $i \in \{1, \ldots, n\}$, be the subspace of V generated by all elements of the form $\varphi^k(g_i)$ for $k \in \mathbb{Z}$. There is a basis of V_i of the form $e_1(i) = \varphi^0(g_i) = g_i$, $e_2(i) = \varphi(g_i), \ldots, e_{l_i}(i) = \varphi^{l_i-1}(g_i)$. It can be efficiently constructed as follows.

Initially, we include $e_1(i) = g_i$ in the basis. Then we check whether $\varphi(g_i)$ belongs to the linear subspace generated by $e_1(i)$. If not, then we add $e_2(i) = \varphi(g_i)$ to the basis under construction. Suppose $e_1(i), \ldots, e_j(i)$ is a constructed part of the basis. Then we check whether $\varphi^j(g_i) = \varphi(e_j(i))$ belongs to the linear subspace generated by $e_1(i), \ldots, e_j(i)$. If not, then we add $e_{j+1}(i) = \varphi^j(g_i)$ to the basis under construction, and continue. If so, we stop the process and claim that the basis is constructed and $l_i = j$. Indeed, a linear presentation of $\varphi^j(g_i)$ via $e_1(i), \ldots, e_j(i)$ after applying φ gives a linear presentation of $\varphi^{j+1}(g_i)$ via $e_2(i), \ldots, e_j(i), \varphi^j(g_i)$, and so via $e_1(i), \ldots, e_j(i)$. This argument works for every $j + v$, $v \geq 1$. Similarly we can obtain the linear decomposition of each $\varphi^{-v}(g_i)$, $v \geq 1$.

Step 2: For each $i = 1, \ldots, n$, we have constructed a basis $e_1(i), \ldots, e_{l_i}(i)$ of V_i, where $e_{j+1}(i) = \varphi^j(g_i)$, $j = 0, \ldots, l_i - 1$. Each subspace V_i is φ-invariant. In the general case, $l_i \leq d^2$.

In [Bhunia et al. 2019], the authors single out as the main the case of inner automorphism φ. They write:

> The purpose of this section is to show that for a secure MOR cryptosystem over the classical Chevalley and twisted orthogonal groups, we have to look at automorphisms that act by conjugation like the inner automorphisms. There are other automorphisms that also act by conjugation, like the diagonal automorphism and the graph automorphism for odd-order orthogonal groups. Then we argue what is the hardness of our security assumptions.

Then they note that by the Dieudonné theorem, $\varphi = \sigma\iota\eta\gamma\theta$, where σ is a central automorphism, ι is an inner automorphism, η is a diagonal automorphism, γ is a graph automorphism, and θ is a field automorphism.

Then they continue:

> The group of central automorphisms is too small and the field automorphisms reduce to a discrete logarithm in the field F_q. So there is no benefit of using these in a MOR cryptosystem. Also there are not many graph automorphisms in classical Chevalley and twisted orthogonal groups other than special linear groups and odd-order orthogonal groups. In the odd-order orthogonal groups, these automorphisms act by conjugation.
>
> Recall that our automorphisms are presented as actions on generators. It is clear [Mahalanobis 2012, Section 7] that if we can recover the conjugating matrix from the action on the generators, the security is a discrete logarithm problem in $\mathbb{F}_{q^d}$, or else the security is a discrete logarithm problem in $F_{q^{d^2}}$.

In our cryptanalysis, we assume that φ can be naturally extended to an automorphism of the linear space V. This happens if φ is an inner or field automorphism or is induced by an inner automorphism of $\mathrm{GL}(d, \mathbb{F})$.

We return to the above-introduced subspaces V_i, $i = 1, \ldots, n$. For a fixed V_i, denote by φ_i the linear map of V_i induced by φ. The matrix $A(\varphi_i)$ in the basis $E_i = \{e_1(i), \ldots, e_{l_i}(i)\}$ has the form

$$A(\varphi_i) = \begin{pmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & \cdots & \cdots & 0 & 1 & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \\ \alpha_1 & \alpha_2 & \cdots & \cdots & \cdots & \alpha_{l_i} \end{pmatrix},$$

where $\varphi(e_{l_i}(i)) = \sum_{k=1}^{l_i} \alpha_k e_k(i),\ \alpha_k \in \mathbb{F}$.

In this way, we can efficiently compute for each i the value $\varphi^{-1}(g_i)$ corresponding to the first row of $A(\varphi_i)^{-1}$. So we can compute φ^{-1}.

Now we know the matrices $A(\varphi_i)^{\pm 1}$, $A(\varphi_i)^{\pm r}$, $A(\varphi_i)^{\pm t}$, $i = 1, \ldots, n$, and we need to calculate r or t. Then we can calculate φ_i^{-rt} and restore m. We can provide sufficient calculations using only one or more of the matrices above.

In [Menezes and Vanstone 1992], it was shown how the discrete logarithm problem in some special class of matrices can be reduced to the discrete logarithm problem in some extensions of the underlying field. In [Menezes and Wu 1997], these results were extended to show how the discrete logarithm problem in every group $\mathrm{GL}(d, \mathbb{F})$ can be reduced in probabilistic polynomial time to the similar problem in small extensions of $\mathbb{F}$. The case of a finitely generated nilpotent group is considered in [Roman'kov 2019b].

We see that matrix groups over finite fields offer no significant advantage for the implementation of cryptographic protocols whose security is based on the difficulty of computing discrete logarithms.

The described cryptanalysis has many analogues, presented in [Roman'kov 2013a; Myasnikov and Roman'kov 2015]. In [Roman'kov 2018a], a general scheme based on multiplications is presented. It corresponds to a number of cryptographic systems known in the literature, which are also vulnerable to attacks by the linear decomposition method. The nonlinear decomposition method was invented in [Roman'kov 2016]. The nonlinear method can be applied when the group chosen as the platform for a cryptographic scheme is not linear or the least degree of their representability by matrices is too big for efficient computations. See details in [Roman'kov 2018b].

A protection against linear algebra attacks was recently invented in [Roman'kov 2019a]. It is described in the case of the cryptographic scheme of [Anshel et al. 1999] but can be applied to the DH and some other schemes too. See [Roman'kov 2019c; 2019d]. Further, we'll present this protection in more detail. This version is improved with respect to [Anshel et al. 1999].

5. Cryptanalysis of the Ko et al. and Anshel–Anshel–Goldfeld classical protocols of algebraic cryptography

5A. ***Noncommutative analogues of the DH protocol.*** In algebraic cryptography, the following noncommutative analogues of the DH protocol are considered:

- An analogue with conjugations [Ko et al. 2000]: for a group G and an element $g \in G$, determine by two elements $g^a = aga^{-1}$ and $g^b = bgb^{-1}$, where $a, b \in G$, $ab = ba$, the element $g^{ab} = abga^{-1}b^{-1} = bagb^{-1}a^{-1}$.
- An analogue with twoside multiplication: for a group G and an element $g \in G$, determine by two elements of the form aga' and bgb', where $a, b \in G$, $ab = ba$, $a'b' = b'a'$, the element $abga'b' = bagb'a'$.
- An analogue with automorphisms: for a group G and an element $g \in G$, determine by two elements of the form $\alpha(g)$ and $\beta(g)$, where $\alpha, \beta \in \mathrm{Aut}(G)$, $\alpha\beta = \beta\alpha$, the element $\alpha(\beta(g)) = \beta(\alpha(g))$.

The linear decomposition method under certain natural conditions into the group G (first of all, this is the existence of an effective embedding in a finite-dimensional linear space) effectively solves each of these problems.

The case of two-sided multiplication in its slightly weak form was analyzed in Section 3C. Now we consider the case with conjugations. We will demonstrate two attacks, the first based on the linear decomposition, and the second based on the nonlinear decomposition.

5B. ***The Ko et al. protocol [2000].*** Let $G \leq \mathrm{M}(n, \mathbb{F})$ be a public matrix group over an arbitrary (constructive) field $\mathbb{F}$, and let g be a public element of G. Suppose that $A = \mathrm{gp}(a_1, \dots, a_k)$ and $B = \mathrm{gp}(b_1, \dots, b_l)$ are two pointwise commuting public subgroups of G.

Alice's keys are as follows:

- Private key: $a \in A$.
- Public key: $g^a = aga^{-1}$.

Bob's keys are as follows:

- Private key: $b \in B$.
- Public key: $g^b = bgb^{-1}$.

Algorithm.

- Alice sends g^a to Bob.
- Bob sends g^b to Alice.
- Since Alice knows a, she computes $(g^b)^a = g^{ab}$ from g^b.
- Since Bob knows b he computes $(g^a)^b = g^{ba}$.
- Now both, Alice and Bob, know a secret key $K = g^{ab}$, because $ab = ba$.

5C. ***Cryptanalysis of the Ko et al. protocol.*** We will apply the linear and nonlinear decomposition attacks.

Linear decomposition attack. Let $V = \mathrm{span}(g^A)$ be the linear subspace of $\mathrm{M}(n, \mathbb{F})$ generated by all matrices of the form g^c, $c \in A$. Then $\dim(V) \leq n^2$.

Let $e_1, e_2, \dots, e_r$ be a basis of V that can be efficiently obtained; see [Roman'kov 2013a; 2018b; Myasnikov and Roman'kov 2015]. Let $e_i = g^{c_i}$, $c_i \in A$, $i = 1, \dots, r$.

Since, $g^a \in V$, we can efficiently obtain a presentation of the form

$$g^a = \sum_{i=1}^{r} \alpha_i e_i, \quad \alpha_i \in \mathbb{F}, \; i = 1, \dots, r. \tag{7}$$

Then

$$\sum_{i=1}^{r} \alpha_i g^{c_i} g^b g^{-c_i} = \left(\sum_{i=1}^{r} \alpha_i e_i\right)^b = K. \tag{8}$$

The exchanged key is recovered without computing the private parameters a and b. We did not solve the underlined search conjugacy problem (to find a by g^a or to find b by g^b).

Nonlinear decomposition attack. All assumptions and algorithms are the same as above except the assumption that G is a linear group. In addition we suppose that every subgroup of G is finitely generated and the membership problem for G is efficiently decidable. For example, G is a finitely generated nilpotent or more generally polycyclic group. See [Roman'kov 2016; 2018b] for details.

Let g^A be subgroup of G generated by all elements of the form $g^c, c \in A$. Let $g_i = g^{c_i}$, $c_i \in A$, $i = 1, \ldots, r$, be a finite generating set of g^A. We suppose that this generating set can be efficiently constructed; see [Roman'kov 2016; 2018b] again.

Since, $g^a \in g^A$, we can efficiently obtain a presentation of the form

$$g^a = \prod_{i=1}^{s} g_{i_j}^{\epsilon_i}, \quad i_j \in \{1, \ldots, r\},\ \epsilon_i \in \{\pm 1\},\ i = 1, \ldots, s. \tag{9}$$

Then

$$\prod_{i=1}^{s} c_{i_j} (g^b)^{\epsilon_i} c_{i_j}^{-1} = \left(\prod_{i=1}^{s} g_{i_j}^{\epsilon_i}\right)^b = K. \tag{10}$$

The exchanged key is recovered without computing the private parameters a and b. We did not solve the underlined search conjugacy problem (to find a by g^a or to find b by g^b).

5D. ***The Anshel–Anshel–Goldfeld protocol [Anshel et al. 1999].*** M. Anshel, I. Anshel and D. Goldfeld [Anshel et al. 1999], see also [Myasnikov et al. 2008; 2011; Roman'kov 2012], proposed a group-based key exchange protocol that we call the AAG protocol. It works as follows.

Suppose two correspondents Alice and Bob want to exchange a key. They agree about a group G given by a finite set of generators that is used as the platform. It is supposed that G is equipped with an efficient normal form of its elements and the main group operations can be computed efficiently. All the information about G, the normal form and efficient algorithms to compute products of elements, its inversions and normal forms, is public. In particular, the word problem is efficiently solvable for G.

To exchange a key the correspondents act as follows.

Alice fixes a positive integer k and chooses a tuple of elements $\bar{a} = (a_1, \ldots, a_k)$. Bob fixes a positive integer l and chooses a tuple of elements $\bar{b} = (b_1, \ldots, b_l)$. These two tuples are public.

Algorithm.

- Alice picks a private group word $u = u(x_1, \ldots, x_k)$; then she computes $u_0 = u(a_1, \ldots, a_k)$ and sends the tuple $\bar{b}^{u_0} = (b_1^{u_0}, \ldots, b_l^{u_0})$ to Bob.
- Bob picks a private group word $v = v(y_1, \ldots, y_l)$; then he computes $v_0 = v(b_1, \ldots, b_k)$ and sends the tuple $\bar{a}^{v_0} = (a_1^{v_0}, \ldots, a_k^{v_0})$ to Alice.
- Alice computes
$$u(a_1^{v_0}, \ldots, a_k^{v_0}) u_0^{-1} = u_0^{v_0} u_0^{-1} = [v_0, u_0].$$

- Bob computes
$$v_0 v(b_1^{u_0}, \dots, b_l^{u_0})^{-1} = v_0(v_0^{u_0})^{-1} = [v_0, u_0].$$
Now the commutator
$$K = [v_0, u_0]$$
is the secret exchanged key.

5E. ***Cryptanalysis of the Anshel–Anshel–Goldfeld protocol.*** The AAG protocol was analyzed by Tsaban in [Tsaban 2015; Ben-Zvi et al. 2018]. We will give his analysis for the reader's convenience because we are going to present an improvement of AAG to make it resistant to such sort of attacks.

The commutator key-exchange protocol uses the Artin braid group B_n, $n \in \mathbb{N}$, as its platform group. It was shown in [Tsaban 2015] that the problem of computing the exchanged key reduces, polynomially, to the same problem in matrix groups over finite fields. Now let G be a matrix group and two sets $\{a_1, \dots, a_k\}$ and $\{b_1, \dots, b_l\}$ be as in the protocol. Let $A = \mathrm{gp}(a_1, \dots, a_k)$ and $B = \mathrm{gp}(b_1, \dots, b_l)$ be subgroups generated by these sets respectively. Also denote by $\mathrm{Alg}(A)$ and $\mathrm{Alg}(B)$ the subalgebras (and so vector spaces) generated by A and B respectively.

The linear span-method by Tsaban works as follows:

(1) Compute bases for the vector spaces of $\mathrm{Alg}(A)$ and $\mathrm{Alg}(B)$.

(2) Solve the following homogeneous system of linear equations in the unknown matrix $x \in \mathrm{Alg}(A)$:
$$b_i \cdot x = x \cdot b_i^{u_0}, \quad i = 1, \dots, l,$$
a system of linear equations on the coefficients determining the matrix x, as a linear combination of the basis of the space $\mathrm{Alg}(A)$.

(3) Fix a basis for the solution space, and pick random solutions until the picked solution x_0 is invertible.

(4) Solve the following homogeneous system of linear equations in the unknown matrix $y \in \mathrm{Alg}(B)$:
$$a_j \cdot y = y \cdot a_j^{v_0}, \quad j = 1, \dots, k,$$
a system of linear equations on the coefficients determining y, as a linear combination of the basis of the space $\mathrm{Alg}(A)$.

(5) Fix a basis for the solution space, and pick random solutions until the picked solution y_0 is invertible.

(6) Output:
$$[x_0, y_0].$$

It is easy to prove, see [Ben-Zvi et al. 2018], that the output is correct, i.e., $[x_0, y_0] = [u_0, v_0]$. That steps (3) and (5) terminate quickly follows from the invertibility lemma, Lemma 2.2.

Remark 5.1. The linear span-method by Tsaban et al. described above is efficiently applicable to schemes based on the intractability of the conjugacy search problem for matrix groups over finite fields. It cannot be directly applied to schemes that use abstract groups or matrices over infinite fields groups as the platforms.

6. Marginal subsets

In this section we introduce a new concept that can be effectively used to improve some cryptographic schemes, including algebraic cryptography protocols like AAG and DH. This concept formally generalizes the well-known concept of the marginal subgroup, but it is worth noting that this generalization is very different from the original concept.

The marginal subgroup is determined by the word, and the marginal subset is determined by the word and its chosen value. The set of all marginal subsets is not closed under group-theoretic operations. A marginal subset can be very wild.

Let F be a free group on a countably infinite set $\{x_1, x_2, \dots\}$ and let W be a nonempty subset of F. If $w = w(x_1, \dots, x_n) \in W$ and $g_1, \dots, g_n$ are elements of a group G, we define the *value* of the word w at $(g_1, \dots, g_n)$ to be $w(g_1, \dots, g_n)$. The subgroup of G generated by all values in G of words in W is called the *verbal subgroup* of G determined by W,

$$W(G) = \mathrm{gp}(w(g_1, \dots, g_n) : g_i \in G, w \in W).$$

If W is a nonempty set of words in $x_1, x_2, \dots$ and G is any group, a normal subgroup N is said to be *W-marginal* in G if

$$w(g_1, \dots, g_n) = w(u_1 g_1, \dots, u_n g_n)$$

for all $w(x_1, \dots, x_n) \in W,\ g_i \in G,\ u_i \in N,\ 1 \leq i \leq n$. This is equivalent to the requirement $g_i = f_i \pmod N),\ 1 \leq i \leq n$, always implies that $w(g_1, \dots, g_n) = w(f_1, \dots, f_n)$.

In particular, for $n \in \mathbb{N}$, any group word $w = w(x_1, \dots, x_n)$ and any group G, a normal subgroup N is said to be w-marginal in G if

$$w(g_1, \dots, g_n) = w(u_1 g_1, \dots, u_n g_n)$$

for all $g_i \in G,\ u_i \in N,\ 1 \leq i \leq n$. This is equivalent to the requirement $g_i = f_i \pmod N),\ 1 \leq i \leq n$, always implies that $w(g_1, \dots, g_n) = w(f_1, \dots, f_n)$.

Since every set of W-marginal subgroups of G generate a normal subgroup that is also marginal, there is the maximal W-marginal (in particular w-marginal) subgroup of G denoted by $W^*(G)$ (in particular $w^*(G)$). See [Robinson 1982] for more details about verbal and marginal subgroups.

We introduce a new notion that significantly extends the marginality property. For simplicity we give this notion for the case when W consists of a single word w. This notion can be easily extended to any set W.

Definition 6.1. For $n \in \mathbb{N}$, let $w = w(x_1, \dots, x_n)$ be a group word, G be a group and $\bar{g} = (g_1, \dots, g_n)$ be a tuple of elements of G. We say that a tuple $\bar{c} = (c_1, \dots, c_n) \in G^n$ is a *marginal tuple* determined by w and $\bar{g}$ if

$$w(c_1 g_1, \dots, c_n g_n) = w(g_1, \dots, g_n).$$

We will write $\bar{c} \perp w(\bar{g})$ in this case. A set $\overline{C} \subseteq G^n$ is said to be *marginal* with respect to w and $\bar{g}$, and write $\overline{C} \perp w(\bar{g})$, if $\bar{c} \perp w(\bar{g})$ for every tuple $\bar{c} \in \overline{C}$.

Remark 6.2. Let G be a group, $w = w(x_1, \dots, x_n)$ be a word and $\bar{g} = (g_1, \dots, g_n)$ be a tuple of elements of G. Then the following marginality properties are true for G, w and $\bar{g}$:

(1) Each subset of a marginal set is marginal.

(2) The direct power $(w^*)^n$ is marginal.

(3) The component c_i of any marginal tuple $\bar{c}$ can be any element of the group G if w is independent of x_i.

(4) The set $C_i,\ i = 1, \dots, n$, consisting of all i-th components of all $\bar{c} \in \bar{C}$, is generally not closed with respect to group operations. For example, if g_i occurs in $w(\bar{g})$ all times in the form g_i^2 then any element $h \in G$ such that $h^2 = 1$ and $hg_i = g_i h$ can be the i-th component of a marginal tuple $\bar{c}$ with trivial other components. But the product of two such elements h cannot be an involution and so this product is out of C_i in the general case.

(5) There are many ways to construct a marginal set. Obviously, we can even construct a nonrecursive marginal set in the case of the infinite group G. Below we present a very simple and efficient algorithm for constructing a marginal set using the word w.

A method for constructing the marginal set $\bar{C}$, $\bar{C} \perp w$, based on w. As we noted in Remark 6.2, the marginal set $\bar{C}$, $\bar{C} \perp w$, is generally not closed under group operations. This set can be chosen as very wild; for example, it can be computable, but not recursive. We are to develop various methods for creating such sets. We also note that the proposed idea can be established as an improvement of many other cryptographic schemes based on the insolubility of the problem of finding conjugacy in groups to make these schemes resistant to attacks by the linear algebra methods.

Now we give a very simple and efficient algorithm for constructing the marginal set $\bar{C}$ using the word w. This method is universal because it does not depend on the structure of G.

Let $w = w(a_1, \dots, a_k) = a_1 a_2 \cdots a_k,\ a_i \in G,\ i = 1, \dots, k$, be any expression in the straight form of a fixed element $f \in G$. It is possible that $a_i = a_j$ or $a_i = a_j^{-1}$ for $i \neq j$. Also this expression can be nonreduced. Consider the equation

$$x_1 a_1 x_2 a_2 \cdots x_k a_k = f. \tag{11}$$

Every solution of (11) can be included in a marginal set $\bar{C}$, $\bar{C} \perp w$. We can fix i and choose any values $x_j = c_j,\ j \neq i,\ c_j \in G$. Then we obtain the solution of (11) by setting

$$x_i = a_{i-1}^{-1} c_{i-1}^{-1} \cdots a_1^{-1} c_1^{-1} f a_k^{-1} c_k^{-1} \cdots a_{i+1}^{-1} c_{i+1}^{-1}. \tag{12}$$

We can also generate a solution of (11) using a sequence of the following random elementary inserts. Suppose we have a solution $(c_1, \dots, c_k)$ of (11). For any i and any random element $d \in G$ we can change c_i to $c_i' = c_i a_i d a_i^{-1}$ and c_{i+1} to $c_{i+1}' = d c_{i+1}$. Then we get a new solution of (11). Continuing this process with random i and d, we get a series of new solutions of (11).

Remark 6.3. In the case when $G \leq \mathrm{M}(n, \mathbb{F})$ is a matrix group over $\mathbb{F}$, the notion of a marginal set can be naturally generalized to any ring-word (even to any algebra-word). Let R be a free associative algebra on a countably infinite set $\{x_1, x_2, \dots\}$ over a field $\mathbb{F}$, and let W be a nonempty subset of R. If $w = w(x_1, \dots, x_n) \in W$ and $u_1, \dots, u_n$ are elements of $M = \mathrm{M}(n, \mathbb{F})$, we define the *value* of the word w at $(u_1, \dots, u_n)$ to be $w(u_1, \dots, u_n)$. Let $\bar{g} = (g_1, \dots, g_n)$ be a tuple of elements of G. We say that a tuple $\bar{c} = (c_1, \dots, c_n) \in M^n$ is a *marginal tuple* determined by w and $\bar{g}$ if

$$w(c_1 g_1, \dots, c_n g_n) = w(g_1, \dots, g_n).$$

Other generalizations when we use the ring T instead of the group G or use more general operations instead of multiplication on the left side are also possible.

7. Improved versions of the AAG and Ko et al. cryptographic protocols

Suppose two correspondents Alice and Bob want to exchange a key. They agree about a group G given by a finite set of generators that is used as the platform. It is supposed that G is equipped with an efficient normal form of its elements and the main group operations can be computed efficiently. All the information about G, the normal form and efficient algorithms to compute products of elements, their inversions and normal forms, is public. In particular, the word problem is efficiently solvable for G.

7A. ***An improved version of the AAG key exchange protocol.*** To exchange a key the correspondents act as follows.

Alice fixes a positive integer k and chooses a tuple of elements $\bar{a} = (a_1, \dots, a_k)$. Then she picks up a private group word $u = u(x_1, \dots, x_k)$ and computes $u(\bar{a}) = u(a_1, \dots, a_k)$. Also she finds a marginal set $\bar{C} \subseteq G^k$, $\bar{C} \perp u(\bar{a})$.

Bob fixes a positive integer l and chooses a tuple of elements $\bar{b} = (b_1, \dots, b_l)$. Then he picks up a private group word $v = v(y_1, \dots, y_l)$ and computes $v(\bar{b}) = v(b_1, \dots, b_l)$. Also he finds a marginal set $\bar{D} \subseteq G^l$, $\bar{D} \perp v(\bar{b})$.

Alice publishes elements $a_1, \dots, a_k$ as $a_{\pi(1)}, \dots, a_{\pi(k)}$, where $\pi \in \mathbb{S}_k$ is a random permutation. The same permutation is applied to the corresponding tuples $\bar{c} \in \bar{C}$.

Bob acts in the similar way.

Virtual and hidden elements. Alice can also introduce a virtual element h that is not used in the expression for $u(\bar{a})$. Then she add a new random component to any $\bar{c} \in \bar{C}$, $\bar{C} \perp w$. She can add many such components with aim to hide the length of the word u, or to hide equality (12), or choose some element h with huge centralizer as well as with small centralizer, to make solution of the problem more difficult for an intruder. Bob acts similarly.

Also Alice can hide some elements a_i as follows. Let $a_i = a_j$ and the corresponding components $c_i = c_j$ for all $\bar{c} \in \bar{C}$. Then Alice does not publish a_j and removes the j-component from every $\bar{c}$. Bob acts similarly.

These two operations are recommended. After these operations the parameters k and l can be changed to k' and l' respectively.

Alice publishes elements $a_1, \dots, a_{k'}$ as $a_{\pi(1)}, \dots, a_{\pi(k')}$, where $\pi \in \mathbb{S}_{k'}$ is a random permutation. The same permutation is applied to the corresponding tuples $\bar{c} \in \bar{C}$.

Bob acts in the similar way.

Alice publishes elements $a_1, \dots, a_{k'}$ as $a_{\pi(1)}, \dots, a_{\pi(k')}$, where $\pi \in \mathbb{S}_{k'}$ is a random permutation. The same permutation is applied to the corresponding tuples $\bar{c} \in \bar{C}$, and they are published.

Bob acts in the similar way.

Algorithm.

- Alice picks a private tuple $\bar{d} = (d_1, \dots, d_{l'}) \in \bar{D}$ and computes $\overline{db} = (d_1b_1, \dots, d_{l'}b_{l'})$. Then she sends the tuple $\overline{db}^{u(\bar{a})} = ((d_1b_1)^{u(\bar{a})}, \dots, (d_{l'}b_{l'})^{u(\bar{a})})$ to Bob.
- Bob picks a private tuple $\bar{c} = (c_1, \dots, c_{k'}) \in \bar{C}$ and computes $\overline{ca} = (c_1a_1, \dots, c_{k'}a_{k'})$. Then he sends the tuple $\overline{ca}^{v(\bar{b})} = ((c_1a_1)^{v(\bar{b})}, \dots, (c_{k'}a_{k'})^{v(\bar{b})})$ to Alice.
- Alice computes

$$u((c_1a_1)^{v(\bar{b})}, \dots, (c_ka_k)^{v(\bar{b})}) = u(\bar{a})^{-1}u(c_1a_1, \dots, c_ka_k)^{v(\bar{b})} = [u(\bar{a}), v(\bar{b})].$$

- Bob computes similarly
$$v((d_1b_1)^{u(\bar{a})}, \dots, (d_lb_l)^{u(\bar{a})})^{-1}v(\bar{b}) = (v(d_1b_1, \dots, d_lb_l)^{u(\bar{a})})^{-1}v(\bar{b}) = [u(\bar{a}), v(\bar{b})].$$
Now the commutator
$$K = [u(\bar{a}), v(\bar{b})]$$
is the secret exchanged key.

Definition 7.1. The conjugacy-membership problem is solvable for G with respect to $\bar{C} \subseteq^k$ if there is an algorithm that decides for any two tuples $\bar{a} = (a_1, \dots, a_k)$ and $\bar{f} = (f_1, \dots, f_k)$ of elements of G whether or not there exists an element $y \in G$ such that $(f_1^y a_1^{-1}, \dots, f_k^y a_k^{-1}) \in \bar{C}$. In short, is there an element $y \in G$ such that $\bar{f}^y \bar{a}^{-1} \in \bar{C}$? The corresponding problem, which is a mixture of conjugacy and membership problems, is the question of the existence of an algorithm that finds a solution, if such a solution exists.

The proposed version of the AAG protocol is based on intractability of the mixed conjugacy-membership search problem when $\bar{C}$ is a marginal set, $\bar{C} \perp u(a_1, \dots, a_k)$, for the unknown word $u(x_1, \dots, x_n)$ (or similarly when $\bar{D}$ is a marginal set, $\bar{D} \perp v(b_1, \dots, b_l)$). Indeed, suppose that an intruder finds $\bar{c}' \in \bar{C}$ and $y \in G$ such that $\overline{ca}^{v(\bar{b})} = \overline{c'a}^y$, and similarly he finds $\bar{d}' \in \bar{D}$ and $x \in G$ such that $\overline{db}^{u(\bar{a})} = \overline{d'b}^x$. Then $[x, y] = [u(\bar{a}), v(\bar{b})]$ as in the original version.

There are other problems that should probably be addressed first. The presence of virtual and hidden elements does not allow us to calculate the lengths of u and v. We also note that each solution of (11) is also a solution to each equation of the form $a_ia_{i+1} \cdots a_ka_1 \cdots a_{i-1} = f$, $i = 2, \dots, k$, and possibly some other equations. Therefore, the open data does not allow us to unambiguously restore $f^{v(\bar{b})}$, even if the attacker knows the length of v and all the letters $v(\bar{b})$ with their multiplicity.

7B. ***An improved version of the Ko et al. key exchange protocol.*** To exchange a key the correspondents act as follows.

Let G be a group. Alice and Bob agree about a public element $g \in G$. Let A and B be two finitely generated elementwise commuting subgroups of G. This data is public.

Alice fixes a positive integer k and chooses a tuple of elements $\bar{f} = (f_1, \dots, f_k)$ such that $g \in \mathrm{gp}(f_1, \dots, f_k)$. Then she picks a private group word $u = u(x_1, \dots, x_k)$ such that $g = u(\bar{f})$. Also she finds a marginal set $\bar{C} \subseteq G^k$, $\bar{C} \perp u(\bar{f})$. Alice publishes $\bar{C}$.

Bob fixes a positive integer l and chooses a tuple of elements $\bar{f}' = (f_1', \dots, f_l')$ such that $g \in \mathrm{gp}(f_1', \dots, f_l')$. Then he picks a private group word $v = v(x_1, \dots, x_l)$ such that $g = v(\bar{f}')$. Also he finds a marginal set $\bar{D} \subseteq G^l$, $\bar{D} \perp v(\bar{f}')$. Bob publishes $\bar{D}$.

If G is a matrix group, the words u and v can be ring-words.

Algorithm.

- Alice chooses a private tuple $\bar{h} = (h_1, \dots, h_k) \in \mathrm{C}_G(B)^k$ and computes $\tilde{f} = (f_1h_1, \dots, f_kh_k)$. Then she publishes $\tilde{f}$.
- Bob chooses a private tuple $\bar{h}' = (h_1', \dots, h_l') \in \mathrm{C}_G(A)^l$ and computes $\tilde{f}' = (f_1'h_1', \dots, f_l'h_l')$. Then he publishes $\tilde{f}'$.
- Alice picks a random tuple $\bar{d} = (d_1, \dots, d_l) \in \bar{D}$ and computes $\bar{d}\tilde{f}' = (d_1\tilde{f}_1', \dots, d_l\tilde{f}_l')$. She also chooses a random private element $a \in A$. Then she sends $(\bar{d}\tilde{f}')^a = ((d_1\tilde{f}_1')^a, \dots, (d_l\tilde{f}_l')^a)$ to Bob.

- Bob picks a random tuple $\bar{c} = (c_1, \dots, c_k) \in \overline{C}$ and computes $\bar{c}\tilde{f} = (c_1\tilde{f}_1, \dots, c_k\tilde{f}_k)$. He chooses a random private element $b \in B$. Then he sends $(\bar{c}\tilde{f})^b = ((c_1\tilde{f}_1)^b, \dots, (c_k\tilde{f}_k)^b)$ to Alice.
- Alice computes
$$(\bar{c}\tilde{f})^b\bar{h}^{-1} = ((c_1\tilde{f}_1)^b h_1^{-1}, \dots, (c_k\tilde{f}_k)^b h_k^{-1}) = ((c_1f_1)^b, \dots, (c_kf_k)^b) = (\bar{c}\bar{f})^b.$$
- Alice computes
$$u((\bar{c}\bar{f})^b) = u(\bar{c}\bar{f})^b = u(\bar{f})^b = g^b.$$
- Bob computes
$$(\bar{d}\tilde{f}')^a(\bar{h}')^{-1} = ((d_1\tilde{f}_1')^a(h_1')^{-1}, \dots, (d_l\tilde{f}_l')^a(h_l')^{-1}) = ((d_1f_1')^a, \dots, (d_lf_l')^a) = (\bar{d}\bar{f}')^a.$$
- Bob computes
$$v((\bar{d}\bar{f}')^a) = v(\bar{d}\bar{f}')^a = v(\bar{f}')^a = g^a.$$
- Alice computes $K_A = (g^b)^a = g^{ab}$.
- Bob computes $K_B = (g^a)^b = g^{ab}$, and
$$K = K_A = K_B = g^{ab}$$
is the secret exchanged key.

Remark 7.2. Alice publishes instead of $f_1, \dots, f_k$ changed elements $\tilde{f}_1, \dots, \tilde{f}_k$. This is done in order to make it difficult for a potential cracker to select the expression $u(f_1, \dots, f_k)$. Since each element h_i lies in $\mathrm{C}_G(B)$, the element $b \in B$ acts on h_i trivially. Alice may exclude $h_i^b = h_i$ from $c_i\tilde{f}^b$ and get $c_if_i^b$. Some of the elements $f_1, \dots, f_k$ are virtual. This means that the value $u(f_1, \dots, f_k)$ does not depend on them. Therefore, the choice in the marginal set $\overline{C}$ of the corresponding components can be carried out randomly. It is also possible that for $i \neq j$ we have $f_i = f_j$. Then both of these elements are published, and the corresponding elements h_i, c_i and h_j, c_j are chosen independently. If an element f_i occurs several times in the expression $u(f_1, \dots, f_k)$, then it is published once. The elements h_i and c_i corresponding to it are also selected once.

All of the above also holds true for Bob to select parameters.

We show a toy example of the just-considered improved version of the key exchange protocol with simple parameters.

Example 7.3. First we will give a symbolic description of the protocol.

Let $G = \mathrm{GL}(6, \mathbb{Z})$, and let $A, B \leq G$ be two elementwise permutable subgroups of G given by their generating sets $\{a_1, \dots, a_k\}$ and $\{b_1, \dots, b_l\}$ respectively, and $g \in G$. This data is public. Suppose that
$$u(x_1, x_2, x_3) = [x_1, x_2]x_2^{-1} + x_2x_3 - x_1$$
is a ring word in which the variables x_1 and x_2 take invertible values.

We choose a pair of elements f_1 and f_2 of G so that the element
$$f_3 = f_2^{-1}g - f_2^{-1}[f_1, f_2]f_2^{-1} + f_2^{-1}f_1$$
is invertible.

Then

$$g = u(f_1, f_2, f_3) = [f_1, f_2]f_2^{-1} + f_2 f_3 - f_1.$$

For $c_1, c_2, c_3 \in \mathrm{M}(6, \mathbb{Z})$, the equality

$$u(c_1 f_1, c_2 f_2, c_3 f_3) = u(f_1, f_2, f_3)$$

is true if and only if

$$c_3 = f_2^{-1} c_2^{-1} (g - [c_1 f_1, c_2 f_2] f_2^{-1} c_2^{-1} + c_1 f_1) f_3^{-1}. \tag{13}$$

The formula (13) describes the full marginal set $\widetilde{C} \perp u(f_1, f_2, f_3)$. Then Alice constructs an infinite marginal set $\overline{C}_3 = \{(c_1(i), c_2(i), c_3(i)) : i = 1, 2, \ldots\}$, choosing the elements $c_1(i)$ and $c_2(i)$ in G and calculating $c_3(i)$ according to (13).

Then Alice randomly chooses the elements $h_1, h_2, h_3 \in \mathrm{C}_G(B)$ (where $\mathrm{C}_G(B)$ denotes the centralizer of B in G) and calculates the elements $\tilde{f_i} = f_i h_i$ for $i = 1, 2, 3$. She also chooses a number $k \geq 3$ and the random virtual elements $\tilde{f_4}, \ldots, \tilde{f_k} \in G$. For each $i = 4, 5, \ldots$, she takes the random elements $c_4(i), \ldots, c_k(i) \in G$ and publishes the constructed marginal set $\overline{C} = \{(c_1(i), c_2(i), c_3(i), c_4(i), \ldots, c_k(i)) : i = 1, 2, \ldots\}$. In practice, she also applies a random permutation to the indices of the tuple $(\tilde{f_1}, \ldots, \tilde{f_k})$ and to each of the corresponding tuples from $\overline{C}$, so as not to show which ones are virtual. To simplify the recording, we do not do this hereinafter.

In continuation of the algorithm Bob picks a random element $b \in B$, chooses randomly $\bar{c}(q) \in \overline{C}$, calculates and publishes the elements

$$(c_i(q)\tilde{f_i})^b \quad \text{for } i = 1, \ldots, k.$$

Alice calculates

$$(c_i(q)\tilde{f_i})^b h_i^{-1} = (c_i(q) f_i)^b \quad \text{for } i = 1, 2, 3. \tag{14}$$

Then she obtains

$$u((c_1(q)f_1)^b, (c_2(q)f_2)^b, (c_3(q)f_3)^b) = u(c_1(q)f_1, c_2(q)f_2, c_3(q)f_3)^b = u(f_1, f_2, f_3)^b = g^b. \tag{15}$$

Also Alice randomly chooses an element $a \in A$ and computes the key: $(g^b)^a = g^{ab}$.

Bob acts the same way.

Next, we will give numerical values for the protocol parameters in our example. We set

$$A = \mathrm{gp}(t_{13}, t_{31}, t_{35}, t_{53}), \qquad B = \mathrm{gp}(t_{24}, t_{42}, t_{46}, t_{64}),$$

where $t_{ij} = e + e_{ij}$, $i \neq j$, is a transvection and, for each pair ij, e_{ij} is an elementary matrix that differs from the zero matrix by one element 1 that stands in the ij-position. Obviously, A and B are elementwise permutable. We also set

$$g = e + e_{12} + e_{23} + e_{34} + e_{45} + e_{56}, \qquad f_1 = t_{23}, \qquad f_2 = t_{34}.$$

Then

$$f_3 = e + e_{12} + 2e_{23} + e_{34} + e_{45} + e_{56} - e_{24} - e_{35},$$
$$f_3^{-1} = e - e_{12} - 2e_{23} - e_{34} - e_{45} - e_{56} + 2e_{13} + 3e_{24} + 2e_{35} + e_{46} - 3e_{14} - 5e_{25} - 2e_{36} + 5e_{15} + 5e_{26} - 5e_{16}.$$

Alice picks

$$h_1 = t_{31}, \qquad h_2 = t_{13} t_{53}^{-1}, \qquad h_3 = t_{15}^2 t_{53}.$$

Then

$$\begin{aligned}\tilde{f_1} &= f_1h_1 = e + e_{23} + e_{21} + e_{31},\\ \tilde{f_2} &= f_2h_2 = e + e_{34} + e_{13} - e_{53},\\ \tilde{f_3} &= f_3h_3 = e - e_{33} + e_{12} + 2e_{23} + e_{34} + e_{45} + e_{56} - 2e_{13} - e_{24} - e_{35} + 2e_{15} + e_{43} + e_{53}.\end{aligned}$$

Bob chooses randomly $\bar{c}(q_0) \in \overline{C}$. For example he takes

$$\begin{aligned}&c_1(q_0) = e - e_{34} + e_{21},\\ &c_2(q_0) = t_{23},\\ &c_3(q_0) = e - 2e_{22} + 6e_{23} - e_{34} + e_{35} - 10e_{24} + 16e_{25} - e_{36} - 16e_{26} + 2e_{21},,\\ &c_i(q_0) \quad \text{for } i = 4, \dots, k\end{aligned}$$

(we do not specify these virtual elements).

Then Bob picks $b = t_{24} \in B$, he calculates the elements

$$\begin{aligned}&(c_1(q_0)\tilde{f_1})^b = e + e_{23} - e_{34} + 2e_{21} + e_{31},\\ &(c_2(q_0)\tilde{f_2})^b = e + e_{34} + e_{13} - e_{53},\\ &(c_3(q_0)\tilde{f_3})^b = e + e_{12} + e_{23} + e_{45} + e_{56} - 2e_{13} - 3e_{24} - e_{35} + 5e_{25} - e_{14} + 2e_{15} + 2e_{21} + e_{43} + e_{53},\\ &(c_i(q_0)\tilde{f_i})^b \quad \text{for } i = 4, \dots, k.\end{aligned}$$

Then he publishes

$$(c_1(q_0)\tilde{f_1})^b, \dots, (c_k(q_0)\tilde{f_k})^b).$$

Suppose that Alice picks $a = t_{35} \in A$.

Alice calculates

$$\begin{aligned}&(c_1(q_0)f_1)^b = (c_1(q_0)\tilde{f_1})^b h_1^{-1} = e + e_{23} - e_{34} + e_{21},\\ &(c_2(q_0)f_2)^b = (c_2(q_0)\tilde{f_2})^b h_2^{-1} = e + e_{34},\\ &(c_3(q_0)f_3)^b = (c_3(q_0)\tilde{f_3})^b h_3^{-1} = e + e_{12} + 4e_{23} + e_{45} + e_{56} - 3e_{24} - e_{35} - e_{14} + e_{25} + 2e_{21}.\end{aligned}$$

By (15) she obtains that

$$g^b = e + e_{12} + e_{23} + e_{34} + e_{45} + e_{56} - e_{14} + e_{25}.$$

Then

$$(g^b)^a = g^{ab} = e + e_{12} + e_{23} + e_{34} + e_{45} + e_{56} + e_{36} - e_{14} \tag{16}$$

is the exchanged key.

Bob takes a ring word

$$v(x_1, x_2, x_3, x_4)) = x_1x_2 - x_3 + x_4$$

and elements

$$\begin{aligned}&f_1' = e + e_{24} - e_{21}, \quad && f_3' = e - e_{23} - e_{45} - e_{56} + e_{24} - e_{21} - e_{32},\\ &f_2' = e + e_{34}, && f_4' = e + e_{12} - e_{32}.\end{aligned}$$

Then

$$g = v(f_1', \dots, f_4') = f_1' f_2' - f_3' + f_4'.$$

Bob picks the following random elements in $C_G(A)$:

$$h_1' = e + e_{24}, \quad h_2' = e - e_{42}, \quad h_3' = e + e_{46} + e_{42}, \quad h_4' = e + 2e_{24}.$$

Then he computes

$$\begin{aligned}
\tilde{f}_1' &= f_1' h_1' = e - e_{21} + 2e_{24},\\
\tilde{f}_2' &= f_2' h_2' = e + e_{34} - e_{42} - e_{32},\\
\tilde{f}_3' &= f_3' h_3' = e - e_{23} - e_{45} - e_{56} + e_{24} - e_{21} - e_{32} + e_{46} + e_{42} + e_{26} + e_{22},\\
\tilde{f}_4' &= f_4' h_4' = e + e_{12} - e_{32} + 2e_{24} + 2e_{14} - 2e_{34}.
\end{aligned}$$

The full marginal set $\widetilde{D} \perp v(f_1', \dots, f_4')$ is described by

$$d_4 = (g - d_1 f_1' d_2 f_2' + d_3 f_3')(f_4')^{-1}.$$

Then one has

$$v(d_1 f_1', \dots, d_4 f_4') = v(f_1', \dots, f_4').$$

Bob constructs an infinite marginal set $\overline{D}_4 = \{(d_1(i), \dots, d_4(i)) : i = 1, 2, \dots\}$.

Bob chooses a number $l \geq 4$ and the random virtual elements $f_5', \dots, f_l'$. For each $i = 5, 6, \dots$ he takes the random elements $d_5(i), \dots, d_l(i)$ and publishes the constructed marginal set

$$\overline{D} = \{d_1(i), \dots, d_4(i), d_5(i), \dots, d_l(i) : i = 1, 2, \dots\}.$$

In practice, he also applies a random permutation to the indices of the tuple $(\tilde{f}_1', \dots, \tilde{f}_l')$ and to each of the corresponding tuples from $\overline{D}$, so as not to show which ones are virtual. To simplify the recording, we do not do this hereinafter.

Alice chooses $\bar{d}(p_0) \in \overline{D}$:

$$\begin{aligned}
&d_1(p_0) = e + e_{32}, d_2(p_0) = e - e_{23}, d_3(p_0) = e - e_{45} + e_{13},\\
&d_4(p_0) = e + e_{22} + e_{33} + e_{23} - e_{45} + e_{13} + e_{24} + e_{46} - e_{32} + e_{31}, d_i(p_0)
\end{aligned}$$

for $i = 5, \dots, l$. She computes

$$\begin{aligned}
(d_1(p_0)\tilde{f}_1')^a &= e - 2e_{34} + 2e_{24} - e_{21} + e_{32} - e_{31},\\
(d_2(p_0)\tilde{f}_2')^a &= e + e_{22} - e_{23} + e_{34} - e_{24} + e_{25} - e_{32} - e_{42},\\
(d_3(p_0)\tilde{f}_3')^a &= e + e_{22} - e_{23} - e_{45} - e_{56} + e_{24} + e_{46} + e_{25} - e_{36} + e_{26} - e_{21} - e_{32} + e_{42},\\
(d_4(p_0)\tilde{f}_4')^a &= e + e_{22} + e_{33} + e_{23} - e_{45} + e_{13} + e_{24} - e_{35} + e_{46} - e_{25} - e_{15} - e_{32} + e_{31},
\end{aligned}$$

and $(d_i(p_0)\tilde{f}_i')^a$ for $i = 5, \dots, l$.

Then she publishes

$$((d_1(p_0)\tilde{f}_1')^a, \dots, (d_l(p_0)\tilde{f}_l')^a).$$

Bob computes

$$
\begin{aligned}
(d_1(p_0)f_1')^a &= (d_1(p_0)\tilde{f}_1')^a(h_1')^{-1} = e+e_{34}+e_{24}-e_{21}+e_{32}-e_{31},\\
(d_2(p_0)f_2')^a &= (d_2(p_0)\tilde{f}_2')^a(h_2')^{-1} = e-e_{23}+e_{34}-e_{24}+e_{25},\\
(d_3(p_0)f_3')^a &= (d_3(p_0)\tilde{f}_3')^a(h_3')^{-1} = e-e_{12}-e_{23}-2e_{45}-e_{56}+e_{13}+e_{24}+e_{46}+e_{25}-e_{36}-e_{15}-e_{21}-e_{32},\\
(d_4(p_0)f_4')^a &= (d_4(p_0)\tilde{f}_4')^a(h_4')^{-1} = e+e_{33}+e_{23}-e_{45}+e_{13}+e_{24}+e_{46}-e_{35}-e_{25}-e_{15}-2e_{32}+e_{31}.
\end{aligned}
$$

Now he obtains

$$v((d_1(p_0)f_1')^a, \dots, (d_4(p_0)f_4')^a) = v(d_1(p_0)f_1', \dots, d_4(p_0)f_4')^a = g^a, \tag{17}$$

and computes $(g^a)^b = g^{ab}$; see (16).

References

[Anshel et al. 1999] I. Anshel, M. Anshel, and D. Goldfeld, "An algebraic method for public-key cryptography", *Math. Res. Lett.* **6**:3-4 (1999), 287–291. MR Zbl

[Baba et al. 2011] S. Baba, S. Kotyada, and R. Teja, "A non-Abelian factorization problem and an associated cryptosystem", report 2011/048, Cryptology ePrint Archive, 2011, available at https://eprint.iacr.org/2011/048.pdf.

[Ben-Zvi et al. 2018] A. Ben-Zvi, A. Kalka, and B. Tsaban, "Cryptanalysis via algebraic spans", pp. 255–274 in *Advances in cryptology—CRYPTO 2018, Part I*, edited by H. Shacham and A. Boldyreva, Lecture Notes in Comput. Sci. **10991**, Springer, 2018. MR Zbl

[Bhunia et al. 2019] S. Bhunia, A. Mahalanobis, P. Shinde, and A. Singh, "The MOR cryptosystem in classical groups with a Gaussian elimination algorithm for symplectic and orthogonal groups", in *Modern cryptography*, edited by M. Domb, IntechOpen, 2019.

[Bigelow 2001] S. J. Bigelow, "Braid groups are linear", *J. Amer. Math. Soc.* **14**:2 (2001), 471–486. MR Zbl

[Cheon and Jun 2003] J. H. Cheon and B. Jun, "A polynomial time algorithm for the braid Diffie–Hellman conjugacy problem", pp. 212–225 in *Advances in cryptology—CRYPTO 2003*, edited by D. Boneh, Lecture Notes in Comput. Sci. **2729**, Springer, 2003. MR Zbl

[Fine et al. 2016] B. Fine, A. I. S. Moldenhauer, and G. Rosenberger, "Cryptographic protocols based on Nielsen transformations", *J. Comput. and Comm.* **4**:12 (2016), 63–107.

[Garber et al. 2006] D. Garber, S. Kaplan, M. Teicher, B. Tsaban, and U. Vishne, "Length-based conjugacy search in the braid group", pp. 75–87 in *Algebraic methods in cryptography*, edited by L. Gerritzen et al., Contemp. Math. **418**, Amer. Math. Soc., Providence, RI, 2006. MR Zbl

[Hofheinz and Steinwandt 2002] D. Hofheinz and R. Steinwandt, "A practical attack on some braid group based cryptographic primitives", pp. 187–198 in *Public key cryptography—PKC 2003*, edited by Y. G. Desmedt, Lecture Notes in Comput. Sci. **2567**, Springer, 2002. MR

[Hughes 2002] J. Hughes, "A linear algebraic attack on the AAFG1 braid group cryptosystem", pp. 176–189 in *ACISP 2002: Information security and privacy* (Melbourne, 2002), edited by L. Batten and J. Seberry, Lecture Notes in Computer Science **2384**, Springer, 2002. Zbl

[Hughes and Tannenbaum 2002] J. Hughes and A. Tannenbaum, "Length-based attacks for certain group based encryption rewriting systems", report 2003/102, Cryptology ePrint Archive, 2002, available at https://eprint.iacr.org/2003/102.pdf.

[Kahrobaei and Khan 2006] D. Kahrobaei and B. Khan, "NIS05-6: a non-commutative generalization of ElGamal key exchange using polycyclic groups", pp. 1–5 in *IEEE Globecom 2006*, 2006.

[Ko et al. 2000] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J.-s. Kang, and C. Park, "New public-key cryptosystem using braid groups", pp. 166–183 in *Advances in cryptology—CRYPTO 2000* (Santa Barbara, CA 2000), edited by M. Bellare, Lecture Notes in Comput. Sci. **1880**, Springer, 2000. MR Zbl

[Krammer 2002] D. Krammer, "Braid groups are linear", *Ann. of Math.* (2) **155**:1 (2002), 131–156. MR Zbl

[Mahalanobis 2008] A. Mahalanobis, "A simple generalization of the ElGamal cryptosystem to non-abelian groups", *Comm. Algebra* **36**:10 (2008), 3878–3889. MR Zbl

[Mahalanobis 2012] A. Mahalanobis, "A simple generalization of the ElGamal cryptosystem to non-abelian groups II", *Comm. Algebra* **40**:9 (2012), 3583–3596. MR Zbl

[Mahalanobis 2015] A. Mahalanobis, "The MOR cryptosystem and extra-special p-groups", *J. Discrete Math. Sci. Cryptogr.* **18**:3 (2015), 201–208. MR Zbl

[Menezes and Vanstone 1992] A. J. Menezes and S. A. Vanstone, "A note on cyclic groups, finite fields, and the discrete logarithm problem", *Appl. Algebra Engrg. Comm. Comput.* **3**:1 (1992), 67–74. MR Zbl

[Menezes and Wu 1997] A. J. Menezes and Y.-H. Wu, "The discrete logarithm problem in $\mathrm{GL}(n, q)$", *Ars Combin.* **47** (1997), 23–32. MR Zbl

[Monico 2016] C. Monico, "Cryptanalysis of a matrix-based MOR system", *Comm. Algebra* **44**:1 (2016), 218–227. MR Zbl

[Myasnikov and Roman'kov 2015] A. Myasnikov and V. Roman'kov, "A linear decomposition attack", *Groups Complex. Cryptol.* **7**:1 (2015), 81–94. MR Zbl

[Myasnikov and Ushakov 2007] A. D. Myasnikov and A. Ushakov, "Length based attack and braid groups: cryptanalysis of Anshel–Anshel–Goldfeld key exchange protocol", pp. 76–88 in *Public key cryptography—PKC 2007*, edited by T. Okamoto and X. Wang, Lecture Notes in Comput. Sci. **4450**, Springer, 2007. MR Zbl

[Myasnikov et al. 2005] A. Myasnikov, V. Shpilrain, and A. Ushakov, "A practical attack on a braid group based cryptographic protocol", pp. 86–96 in *Advances in cryptology—CRYPTO 2005*, edited by V. Shoup, Lecture Notes in Comput. Sci. **3621**, Springer, 2005. MR

[Myasnikov et al. 2006] A. Myasnikov, V. Shpilrain, and A. Ushakov, "Random subgroups of braid groups: an approach to cryptanalysis of a braid group based cryptographic protocol", pp. 302–314 in *Public key cryptography—PKC 2006*, edited by M. Yung et al., Lecture Notes in Comput. Sci. **3958**, Springer, 2006. MR

[Myasnikov et al. 2008] A. Myasnikov, V. Shpilrain, and A. Ushakov, *Group-based cryptography*, Birkhäuser, Basel, 2008. MR

[Myasnikov et al. 2011] A. Myasnikov, V. Shpilrain, and A. Ushakov, *Non-commutative cryptography and complexity of group-theoretic problems*, Mathematical Surveys and Monographs **177**, Amer. Math. Soc., Providence, RI, 2011. MR

[Paeng et al. 2001] S.-H. Paeng, K.-C. Ha, J. H. Kim, S. Chee, and C. Park, "New public key cryptosystem using finite nonabelian groups", pp. 470–485 in *Advances in cryptology—CRYPTO 2001* (Santa Barbara, CA, 2001), edited by J. Kilian, Lecture Notes in Comput. Sci. **2139**, Springer, 2001. MR Zbl

[Peker 2014] Y. K. Peker, "A new key agreement scheme based on the triple decomposition problem", *Int. J. Netw. Secur.* **16**:6 (2014), 426–4360.

[Robinson 1982] D. J. S. Robinson, *A course in the theory of groups*, Graduate Texts in Mathematics **80**, Springer, 1982. MR Zbl

[Roman'kov 2012] V. A. Roman'kov, *Introduction to cryptography*, Forum, Moscow, 2012. In Russian.

[Roman'kov 2013a] V. A. Roman'kov, *Algebraic cryptography*, Omsk State University, 2013. In Russian.

[Roman'kov 2013b] V. A. Roman'kov, "Cryptanalysis of some schemes applying automorphisms", *Prikl. Diskr. Mat.* **21** (2013), 35–51. In Russian.

[Roman'kov 2016] V. Roman'kov, "A nonlinear decomposition attack", *Groups Complex. Cryptol.* **8**:2 (2016), 197–207. MR

[Roman'kov 2018a] V. Roman'kov, "Two general schemes of algebraic cryptography", *Groups Complex. Cryptol.* **10**:2 (2018), 83–98. MR

[Roman'kov 2018b] V. A. Roman'kov, *Essays in algebra and cryptology: algebraic cryptanalysis*, Omsk State University, 2018. In Russian.

[Roman'kov 2019a] V. Roman'kov, "An improved version of the AAG cryptographic protocol", *Groups Complex. Cryptol.* **11**:1 (2019), 35–41. MR

[Roman'kov 2019b] V. A. Roman'kov, "Discrete logarithm for nilpotent groups and cryptanalysis of polylinear cryptographic system", *Prikl. Diskr. Mat. Suppl.* **12** (2019), 154–160. In Russian.

[Roman’kov 2019c] V. A. Roman’kov, “Efficient methods of algebraic cryptanalysis and protection against them”, *Prikl. Diskr. Mat. Suppl.* **12** (2019), 154–160. In Russian.

[Roman’kov 2019d] V. A. Roman’kov, “Linear algebra methods in cryptanalysis and protection against them”, *Herald of Omsk University* **24**:3 (2019), 21–30. In Russian.

[Roman’kov and Obzor 2018] V. A. Roman’kov and A. A. Obzor, “A nonlinear decomposition method in analysis of some encryption schemes using group automorphisms”, *Prikl. Diskr. Mat.* **41** (2018), 38–45. In Russian.

[Romsy 2011] M. Romsy, “Adaption of Pollard’s kangaroo algorithm to the FACTOR problem”, report 2011/483, Cryptology ePrint Archive, 2011, available at https://eprint.iacr.org/2011/483.pdf.

[Sidelnikov et al. 1993] V. M. Sidelnikov, M. A. Cherepnev, and V. V. Yashchenko, “Public key distribution systems based on noncommutative semigroups”, *Dokl. Akad. Nauk* **332**:5 (1993), 566–567. In Russian. MR

[Stanek 2011] M. Stanek, “Extending baby-step giant-step algorithm for FACTOR problem”, report 2011/059, Cryptology ePrint Archive, 2011, available at https://eprint.iacr.org/2011/059.pdf.

[Tsaban 2015] B. Tsaban, “Polynomial-time solutions of computational problems in noncommutative-algebraic cryptography”, *J. Cryptology* **28**:3 (2015), 601–622. MR Zbl

Received 9 Nov 2019. Revised 2 Mar 2020.

VITALIĬ ROMAN’KOV:

romankov48@mail.ru
Mathematical Center, Sobolev Institute of Mathematics of the Siberian Branch of the Russian Academy of Sciences,
Novosibirsk, Russia

Moscow Journal of Combinatorics and Number Theory

msp.org/moscow

Cover design: Blake Knoll, Alex Scorpan and Silvio Levy

See inside back cover or msp.org/moscow for submission instructions.

The subscription price for 2020 is US $310/year for the electronic version, and $365/year (+$20, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues and changes of subscriber address should be sent to MSP.

Moscow Journal of Combinatorics and Number Theory (ISSN 2640-7361 electronic, 2220-5438 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840 is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

MJCNT peer review and production are managed by EditFlow® from MSP.

PUBLISHED BY
mathematical sciences publishers
nonprofit scientific publishing
http://msp.org/

Moscow Journal of Combinatorics and Number Theory

vol. 9 no. 2 2020