Constructions in public-key cryptography over matrix groups

Dima Grigoriev
CNRS, IRMAR, Université de Rennes
Beaulieu, 35042, Rennes, France
dmitry.grigoryev@univ-rennes1.fr
http://name.math.univ-rennes1.fr/dimitri.grigoriev

Ilia Ponomarenko *
Petersburg Department of V.A.Steklov
Institute of Mathematics
Fontanka 27, St. Petersburg 191023, Russia
inp@pdmi.ras.ru
http://www.pdmi.ras.ru/~inp

Abstract

The purpose of the paper is to give new key agreement protocols (a multi-party extension of the protocol due to Anshel-Anshel-Goldfeld and a generalization of the Diffie-Hellman protocol from abelian to solvable groups). They as well as a number of homomorphic public-key cryptosystems rely on difficulty of the conjugacy and membership problems for subgroups of a given group. To support all of them we present a general technique to produce a family of instances being matrix groups (over finite commutative rings) which play a role for these schemes similar to the groups Z_n^* in the existing cryptographic constructions like RSA or discrete logarithm.

Introduction

One of the oldest cryptographical problems consists in constructing of a key agreement protocol. Roughly speaking it is a multi-party algorithm, defined by a sequence of steps,

*Partially supported by RFFI, grants, 03-01-00349, NSH-2251.2003.1. The paper was done during the stay of the author at the Mathematical Institute of the University of Rennes.

specifying the actions of two or more parties in order a shared secret becomes available to two or more parties. Probably the first such procedure based on abelian groups is due to Diffie-Hellman (see [8]). In fact, it concerns automorphisms of abelian (even cyclic) groups induced by taking to a power. Some generalizations of this protocol to non-abelian groups (in particular, the matrix groups over some rings) were suggested in [22] where security was based on an analog of the discrete logarithm problems in groups of inner automorphisms. Certain variations of the Diffie-Hellman systems over the braid groups were described in [14]; there several trapdoor one-way functions connected with the conjugacy and the taking root problems in the braid groups were proposed. A general scheme for constructing key agreement protocols based on algebraic structures was proposed in [1]. In principle, it enables one to construct such protocols for non-abelian groups and their automorphisms induced by conjugations. In this paper we generalize to the non-abelian case the Diffie-Hellman protocol, construct multi party procedure for the protocol [1], and analyze the security of both protocols realized in matrix groups over rings.

The question on finding probabilistic public-key cryptosystems in which the decryption function has a homomorphic property goes back to [23] (see also [7]). In such a cryptosystem the spaces of messages and of ciphertexts are algebraic structures G and H and the decryption function $D : G \to H$ is a homomorphism. A number of such cryptosystems is known for abelian groups, e.g. the quadratic residue cryptosystem [8] and its generalization for highest residues [21] (see also an overview in [11]). In most of them the security is based on the intractability of number-theoretical problems close to the integer factoring. Recently, several homomorphic cryptosystems were constructed for infinite (but finitely presented) groups, see [11, 12] and references there.

The third problem considered in this paper is how to produce instances for cryptosystems based on computations with matrix groups over rings. In contrast to numerous theoretical cryptosystems where there is a lot of efficient algorithms to generate integers with given properties (e.g., the pairs of two distinct large primes of the same bit size used in the quadratic residue cryptosystem), it is not clear a priory how to find efficiently matrix groups in which some problems (like membership or conjugacy) arising in cryptography are computationally difficult. We propose a general scheme for solving this problem and give a specialization of this scheme for matrix groups over finite commutative rings.

In Section 1 we study key agreement protocols between two parties (named usually Alice and Bob) and their extensions to several parties. The security of the Diffie-Hellman protocol relies on the difficulty of the following *transporter problem*: having an action $G \times V \to V$ of a group G on a set V for given $u, v \in V$ to find $g \in G$ (provided that it does exist) such that $(g, u) \mapsto v$. In case of V being a cyclic group of order n and G being a group acting on V by taking a power, one arrives to the discrete logarithm problem (usually, n is taken to be prime). The security of the key agreement protocol of [1] (see also Subsection 1.1) relies on the difficulty of the conjugacy problem with respect to a subgroup of G. In Subsection 1.1 we extend the construction of [1] to *multi-party* key agreement

protocol. Then in Subsection 1.2 we design another generalization of the Diffie-Hellman protocol to actions of groups G which satisfy a certain *identity*. Clearly, any abelian group satisfies the identity $aba^{-1}b^{-1} = 1$ and more generally, any solvable group with a fixed length of its derived series satisfies an appropriate commutator identity. The security of our protocol again relies on the difficulty of the transporter problem for a suitable action of G.

In Section 2 we consider homomorphic public-key cryptosystems (see e.g. [11]) in which the decrypting function (known to Alice) is a group homomorphism $f : G \to H$ where the groups H, G play the roles of the spaces of plain and ciphertext messages, respectively. Usually, the security of a homomorphic cryptosystem relies on the difficulty of the problem of the membership to a normal subgroup of G (here, the kernel of f). For example, in a homomorphic cryptosystem from [12] G was a subgroup of the modular group $\mathrm{SL}_2(\mathbb{Z})$ and the security of this cryptosystem relied on the difficulty of a certain membership problem to a subgroup of the modular group.

The crucial role in the classical cryptographic constructions (like RSA, discrete logarithm or quadratic residue [8]) plays the natural action of the group $\mathrm{Aut}(\mathbb{Z}_n^*)$ on the group $\mathbb{Z}_n^*$. So, varying n one gets a mass pool of instances for cryptographic primitives. This action is a special case of the natural action of the group $Aut_R(V)$ (viewed as a matrix group) on the free module V over the ring R. In this paper we propose a construction of a pool of matrix groups instances for cryptographic primitives (Subsection 3.2). The security of these instances relies on the difficulty of certain problems on matrix groups (e.g. the membership to a subgroup or the conjugacy with respect to a subgroup). For the complexity of such problems few results were established in case of matrix groups over fields [3, 13]; for matrix groups over arbitrary rings much less is known. We note also that matrix groups were mentioned in [6] as candidates for groups with a difficult conjugacy problem.

The common way in cryptography of producing a trapdoor and a cryptosystem, is to generate a private key departing from a pair of primes p, q, while their product $n = pq$ plays the role of a public key. In our scheme (see Subsection 3.1) as a private key we take a rooted tree (called a derivation tree) whose leaves being furnished with specially chosen (non-abelian, in general) groups. We assume that Alice has in possession such representations of these groups which allow her to solve efficiently a problem lying in the background of a cryptosystem (like membership or conjugacy). Internal vertices of the tree are endowed with certain operations on groups which allow one to assign recursively a group to each vertex of the tree starting with its leaves. At the end of the recursion a group is assigned to the root, and this group plays the role of a public key. This scheme is also modified to produce a "secret" homomorphism of matrix groups with a private key being a derivation tree for this homomorphism (similar to the derivation tree of the group). In Subsection 3.2 we give a realization of this general scheme in finite matrix groups.

The similarity of the common constructions in cryptography based on commutative groups (say. $\mathbb{Z}_n^*$) with our construction (relying on finite matrix groups) allows us to call the latter type of constructions the *non-commutative cryptography*.

1 Group-theoretical key agreement protocol

1.1 A multi-party protocol. The following group-theoretical variant of key agreement two-party protocol was proposed in [1]. Let G be a group, and to two parties A and B are assigned their subgroups

$$G_A = \langle a_1, \ldots, a_m \rangle, \quad G_B = \langle b_1, \ldots, b_n \rangle. \tag{1}$$

The group G and the elements a_i, b_j are publically known. The parties A and B choose secret elements $a \in G_A$ and $b \in G_B$ and transmit to each other the collections

$$X_B = \{a^{-1} b_j a\}_{j=1}^n, \quad X_A = \{b^{-1} a_i b\}_{i=1}^m,$$

respectively. Since A (resp. B) has a representation of the element a (resp. b) via generators $a_1, \ldots, a_m$ (resp. $b_1, \ldots, b_n$), then A (resp. B) can compute a representation of the element $b^{-1}ab$ (resp. $a^{-1}ba$) via elements of the set X_A (resp. X_B). Thus, A and B have a common key

$$a^{-1}(b^{-1}ab) = [a, b] = (a^{-1}ba)^{-1}b.$$

An obvious necessary condition for this protocol to be secure is that the set of all such commutators with $a \in G_A$, and $b \in G_B$ should contain at least two elements.

Let us describe a generalization of the group-theoretical key agreement protocol for s parties with $s \geq 2$ and a single public communicating channel. In contrast to the straightforward algorithm having a quadratic complexity, we give an algorithm the complexity of which is linear in s. Without loss of generality we assume that $s = 2^t$ for some $t \geq 1$, for otherwise in the recursive construction below we divide the parties into two unequal subsets which leads just to a slight changing the notation. As in the case $s = 2$ the groups $G_1, \ldots, G_s \subset G$ of the parties are given publically by their sets of generators. At the initial step the ith party chooses a secret key $a_i \in G_i$, $i = 1, \ldots, s$. Let S_1 and S_2 be disjoint $s/2$-subsets of the set $\{1, \ldots, s\}$. Then given $u = 1, 2$ the parties from S_u recursively construct the common key $K_u \in G$, such that for all $i \in S_u$ there exist integers $\varepsilon_{i,j} \in \{-1, +1\}$ and $1 \leq m_i \leq s/2$, $1 \leq j \leq m_i$, and certain elements $B_{i,1}, \ldots, B_{i,m_i} \in \langle \{a_l : \ l \in S_{u,i}\} \rangle$ with $S_{u,i} = S_u \setminus \{i\}$, for which we have

$$K_u = (B_{i,1}^{-1} a_i^{\varepsilon_{i,1}} B_{i,1}) \cdots (B_{i,m_i}^{-1} a_i^{\varepsilon_{i,m_i}} B_{i,m_i}).$$

By recursion we can assume that the ith party knows the elements $B_{i,j}^{-1} a B_{i,j}$ for all j and for all chosen generators a of the group G_i (and thereby, it knows $B_{i,j}^{-1} a_i B_{i,j}$), but does not

necessary know $B_{i,j}$. At this point the party $i \in S_u$ sends the elements $B_{i,j}^{-1} a B_{i,j}$ for all the chosen generators a of the group G_i to a certain party from the set $S_{u'}$ with $u' = 3 - u$ and asks for the elements $K_{u'}^{-1} B_{i,j}^{-1} a B_{i,j} K_{u'}$. Then for $u = 1$ the ith party computes the element

$$[K_1, K_2] = K_1^{-1}(K_2^{-1} K_1 K_2) = K_1^{-1}(K_2^{-1}(B_{i,1}^{-1} a_i^{\varepsilon_{i,1}} B_{i,1}) K_2) \cdots (K_2^{-1}(B_{i,m_i}^{-1} a_i^{\varepsilon_{i,m_i}} B_{i,m_i}) K_2).$$

Similarly, for $u = 2$ the ith party computes the element $[K_1, K_2] = (K_1^{-1} K_2 K_1)^{-1} K_2$. Thus, this element can be chosen as the common key for all parties. It is easy to see that the ith party computes the common key in $O(s|a_i|)$ operations in the group G, where $|a_i|$ denotes the length of the word a_i in the chosen generators of the group G_i.

1.2 A new protocol. In this subsection we define a new group-theoretical two party key agreement protocol that can be viewed as a non-commutative generalization of the Diffie-Hellman protocol (see [8]).

Let G be a group acting on a set X so that given $(x, g) \in X \times G$ the image x^g of x with respect to g can be efficiently computed. Two parties A and B going to choose a secret common key from X, fix publically subgroups G_A, G_B of the group G and two words

$$W_A(u_A, u_B) = u_A^{a_{1,1}} u_B^{b_{1,1}} \cdots u_A^{a_{1,m_1}}, \qquad W_B(u_A, u_B) = u_B^{b_{2,1}} u_A^{a_{2,1}} \cdots u_B^{b_{2,m_2}}$$

of the free group F_2 with two free generators u_A, u_B such that

(W1) $m_1, m_2 \in \mathbb{N}$, $a_{i,j}, b_{i,j} \in \mathbb{Z}$ for all i, j, and $a_{1,m_1} \neq 0$, $b_{2,m_2} \neq 0$,

(W2) $W_A(g_A, g_B) = W_B(g_A, g_B)$ for all $(g_A, g_B) \in G_A \times G_B$.

The protocol begins with the choice of a publically known element $x_0 \in X$ and the secret elements $g_A \in G_A$ by the party A and $g_B \in G_B$ by the party B. Then during the communications the party A performs the following:

- At step 0 set $K_A = x_0$.
- At steps $i = 1, \ldots, m_1 - 1$ send $K_A^{g_A{}^{a_{1,i}}}$ and receive $K_A := K_A^{g_A{}^{a_{1,i}} g_B{}^{b_{1,i}}}$.
- At step $i = m_1$ set $K_A := K_A^{g_A{}^{a_{1,m_1}}}$.

The communications of the party B are defined similarly. Thus, at the end of the communication process due to condition (W2) the parties A and B have the common key

$$K_A = x_0^{W_A(g_A, g_B)} = x_0^{W_B(g_A, g_B)} = K_B.$$

For $X = \mathbb{Z}_p^*$ with p being a prime, $G = G_A = G_B$ being the group $\mathbb{Z}_{p-1}^* \cong \mathrm{Aut}(\mathbb{Z}_p^*)$ and $W_A(u_A, u_B) = u_B u_A$, $W_B(u_A, u_B) = u_A u_B$ we come to the Diffie-Hellman protocol.

This scheme can be easily realized for a solvable group G with bounded length n of the derived series of G. For example, one can take $G_A = G_B = G$ and choose the words $W_A = W_{A,n}$ and $W_B = W_{B,n}$ by induction on n as follows. If $n = 1$, then the group G is abelian and so conditions (W1) and (W2) are satisfied for

$$W_{A,1}(u_A, u_B) = u_B u_A, \qquad W_{B,1}(u_A, u_B) = u_A u_B.$$

For $n \geq 2$ the commutator $[g, h] = g^{-1}h^{-1}gh$ with arbitrary $g, h \in G$ belongs to the derived subgroup $G' = [G, G]$ of G (the derived length of G' equals $n - 1$). Assume by induction that conditions (W1) and (W2) are satisfied for the words $W_{A,n-1}$ and $W_{B,n-1}$. Then a straightforward checking shows that these conditions are also satisfied, for example, for the words

$$W_{A,n} = W_{A,n-1}([u_B, u_A], [u_A^{-1}, u_B^{-1}]), \qquad W_{B,n} = W_{B,n-1}([u_B, u_A], [u_A^{-1}, u_B^{-1}]).$$

Indeed, property (W2) is obvious. Next, one can verify by induction on $n \geq 1$ that the length (the number of letters) of the word $W_{A,n}$ (as well as $W_{B,n}$) equals $2 \cdot 4^{n-1}$. This means that there are no reductions in these words which implies property (W1).

More generally, to define $W_{A,n}$ and $W_{B,n}$ one can choose arbitrary words $W_1, W_2, W_3, W_4 \in W_Y$ where $Y = \{u_A, u_B\}$ and W_Y is the set of all words in the alphabet $Y^{\pm}$, and use $[W_1, W_2]$ and $[W_3, W_4]$ instead of $[u_A, u_B]$ and $[u_B^{-1}, u_A^{-1}]$, respectively. Certainly, to provide condition (1) one should guarantee that the words $W_{A,n-1}(u_A, u_B)$ (resp. $W_{B,n-1}(u_A, u_B)$) and W_2 (resp. W_4) must be terminated to u_A (resp. u_B). To avoid triviality we also should take $W_1, \ldots, W_4$ so that $W_{A,n}$ and $W_{B,n}$ would be nonidentity elements in the underlying free group.

Clearly, any realization of the above protocol is based on identities of the group G. In addition to commutator identities for solvable groups (see above) one can also use the identity $x^m = 1$ (that holds in any finite group the order of which is a divisor of m, and in the Burnside groups). In this case we can choose as the words W_A and W_B the prefix and the inverse of the suffix of the word $(u_A u_B)^m$, respectively, so that the prefix is terminated to u_A. In fact, as it was proved by B.Neumann any variety of groups can be given by a collection of identities such that the first of them is of the form $x^m = 1$ with m being a nonnegative integer, whereas the other ones are the elements of the commutant of the underlying free group (see [17]).

We complete the subsection by making two remarks on the above protocol. First, the set X must be of superpolynomial size, for otherwise the key agreement scheme can be broken in polynomial time by the known permutation group theory technique (see [16]). Second, the words W_A and W_B must be chosen so that the number of elements $W_A(g_A, g_B) = W_B(g_A, g_B)$ with $g_A, g_B \in G$ would be at least two.

1.3 On the security of the protocols. In the above protocols we assume that all groups are given explicitly, e.g. by sets of generators, so that the group operations can be performed efficiently. Then the security of the first protocol is based on the intractability of the following problem (see [24]).

Subgroup Conjugation Search Problem (SCSP). *Given a group G, subgroups H_1, H_2 of G, and two elements $f, g \in H_1$, find an element $h \in H_2$ such that $f = h^{-1}gh$, provided that at least one such h exists.*

As usually in the cryptography, an efficient algorithm solving SCSP would break the protocol (but to break the protocol it is not necessary to solve SCSP). Such an algorithm does exist for $G = \mathrm{GL}(n, \mathbb{F}_q)$ where n is a natural number, $\mathbb{F}_q$ is a finite field of the order q, and the subalgebra $A(H_2)$ of the full matrix algebra $\mathrm{Mat}_n(\mathbb{F}_q)$ generated by the group H_2 is such that

$$A(H_2) \cap G = H_2.$$

Then for arbitrary H_1 the problem SCSP can be solved in probabilistic polynomial time (in n and in $\log q$) by the linear algebra technique, provided that n is less than $q/2$. Indeed, in this case the solution of the linear system $hf - gh = 0$ with respect to $h \in A(H_2)$ is an element of H_2 with a great probability. (From [4] it follows that in this case the problem SCSP can be solved efficiently even by a deterministic algorithm.)

It seems that the problem SCSP remains difficult when G is restricted to subgroups of the group $\mathrm{GL}(V, R)$ of all invertible R-linear transformations of the free R-module V where R is a finite commutative ring. To see this we consider the Linear Transporter Problem on the intractability of which the second protocol is based.

Linear Transporter Problem (LTP). *Let R be a commutative ring, V be an R-module and $G \leq \mathrm{GL}(V, R)$. Given $u \in V$ and $v \in u^G = \{u^g : \ g \in G\}$ find $g \in G$ such that $v = u^g$.*

A special case of (LTP) is the Discrete Logarithm Problem. Indeed, take $V = \mathbb{Z}_p^*$ with p being a prime. Then V can be considered as an one-dimensional module over the ring $R = \mathrm{End}(V) \cong \mathbb{Z}_{p-1}$ (with respect to taking the power $v \mapsto v^n$ where $v \in V$, $n \in \mathbb{Z}_{p-1}$). Choosing u to be a generator of the group V we come to the Discrete Logarithm Problem.

Preserving the notation of LTP set $T(V) = \{T_v : x \mapsto x + v, \ v, x \in V\}$ to be the translation group of the R-module V. Then obviously

$$v = u^g \ \Leftrightarrow \ T_v = g^{-1} T_u g, \qquad u, v \in V, \quad g \in \mathrm{GL}(V, R).$$

So, the problem LTP is the special case of the problem SCSP with $G = \mathrm{AGL}(V, R)$, $H_1 = T(V)$ and $H_2 = \mathrm{GL}(V, R)$. (Here $\mathrm{AGL}(V, R) = T(V)\,\mathrm{GL}(V, R)$ is the group of all affine transformations of V.) This shows that SCSP is at least as hard as LTP. In particular, this construction gives us a family of groups for which the problem SCSP

turns to be at least as hard as the Discrete Logarithm Problem. A general technique to construct groups of this kind will be given in Section 3.

2 Homomorphic cryptosystems over groups

A homomorphic cryptosystem is a probabilistic public-key scheme (in the sense of [8]) in which the spaces of plaintext messages and ciphertexts are groups H_k and G_k, respectively, depending on a security parameter k and such that its decryption function

$$f_k : G_k \to H_k \tag{2}$$

is an epimorphism for all k. Usually, in a homomorphic cryptosystem the public key includes generator sets X_k and Y_k of the groups G_k and H_k, and some set $R_k \subset X_k$ such that $Y_k \subset f_k(R_k) = \{f_k(g) : \ g \in R_k\}$. Besides, it is assumed that there are publically known $k^{O(1)}$-time algorithms to solve the following problems:

(H1) given two elements a, b of G_k (resp. H_k) find the element ab^{-1},

(H2) given $y \in Y_k$ find an element of the set $R_k \cap f_k^{-1}(y)$,

(H3) generate a random element of the group $\ker(f_k)$

where sizes of all elements are assumed to be at most k. Under these assumptions the encryption can be performed in time $k^{O(1)}$ as follows. First, given a message $h = y_1 \cdots y_m \in H_k$ with $y_i \in Y_k$ and m being a natural number at most $k^{O(1)}$, Bob computes in time polynomial in k an element $r = r_1 \cdots r_m \in G_k$ such that $r_i \in R_k$ and $f_k(r_i) = y_i$ for all i. Second, Bob mixes r with random elements $g_1, \cdots, g_{m+1} \in G_k$ belonging to the kernel of the homomorphism f_k and outputs the element $g = g_1 r_1 g_2 \cdots g_m r_m g_{m+1}$ as the ciphertext of h. Alice being able to compute f_k efficiently performs the decoding as follows:

$$f_k(g) = f_k(g_1 r_1 g_2 \cdots g_m r_m g_{m+1}) = f_k(r_1) \cdots f_k(r_m) = y_1 \cdots y_m = h.$$

The key point of such a system is to choose a presentation of the group G_k and the epimorphism f_k in order to provide the inverse of f_k to be a trapdoor function. The exact definition of homomorphic public-key cryptosystems and a survey of constructions can be found in [11, 12].

One way to implement the general concept of a homomorphic cryptosystem is to take G_k to be a subgroup of a certain group F such that the group operations in F can be performed in time polynomial in the size of operands. In the cryptosystems from [11] and [12] the group F was taken as a free product of abelian groups and a modular group, respectively. In these cryptosystems the restriction of the mapping f_k to the set R_k was

known publically and one can produce efficiently random $k^{O(1)}$-size elements of the group $\ker(f_k)$. In fact, the security of these cryptosystems was based on the difficulty of the membership problem (see below) for special subgroups of the group G_k.

Suppose first that the order of the group H_k is at most $k^{O(1)}$ (e.g. such an assumption was used in [11]). Then using the generator set Y_k of H_k one can list all the elements $h_1, \ldots, h_m$ of this group in time $k^{O(1)}$ and then find within the same time a set $\{g_1, \ldots, g_m\}$ of distinct representatives of right cosets of $G_k = \ker(f_k)$ in G_k (one can set $g_i = f_k^{-1}(h_i)$ for all i). Now if an adversary Charlie could recognize efficiently the elements of G belonging to G_k, then he would efficiently compute $f_k(g)$ for all $g \in G_k$ due to the formulae

$$f_k(g) = f_k(g_i) \ \Leftrightarrow \ gg_i^{-1} \in G_k$$

where $i \in \{1, \ldots, m\}$. Thus, in this case the security of our cryptosystem is based on the intractability of the following problem:

Membership Testing (MT). *Given a group F and its subgroup G test whether a given $g \in F$ belongs to G.*

Suppose now the order of $H = H_k$ to be arbitrary. Then a quite natural way to break the cryptosystem is to find an expression of any $g \in G_k$ in the terms of generators belonging to the set X_σ (the attack of this kind was considered in [12]). Indeed, if Charlie could find efficiently for any element $g \in G_k$ an expression $g = x_1 \cdots x_m$ where $x_i \in X_k^{\pm}$ for all i, then he would efficiently compute $f_k(g)$ due to the formulae

$$f_k(g) = f_k(x_1) \cdots f_k(x_m) = f(x_1) \cdots f(x_m)$$

(we recall that the bijection $f : X_k \to Y_k$ is given publically). Thus, in this case we come to the presentation problem (see [12]). The MT problem and the presentation problem are closely related to each other (but generally could be not polynomial-time equivalent) and one can combine them in the following well-known problem of computational group theory (see [3]).

Constructive Membership Testing (CMT). *Given a group F and its subgroup G generated by a set X find an expression of a given $g \in F$ as a word in X, or determine that $g \notin G$.*

Last two decades a great attention was paid to CMT with different presentations of the group G. For example, if F is a subgroup of the symmetric group of degree $n \geq 1$, then the CMT can be solved in time $n^{O(1)}$ by the sift algorithm (see e.g. [16]). In the case of groups $F = \mathrm{GL}(n, \mathbb{F})$ where $\mathbb{F}$ is an algebraic number field, there exists an effective Las Vegas algorithm solving CMT [3]. However, for $n = 1$ and $\mathbb{F}$ being a finite field, CMT is nothing else but the the Discrete Logarithm Problem. In [3] it was conjectured that CMT is difficult whenever the group G either involves a large abelian group as a quotient

of a normal subgroup or has nonabelian composition factors which require large degree permutation representations. Finally, the problem becomes much more difficult if we take $F = \mathrm{GL}(n, R)$ to be the group of $n \times n$ invertible matrices over a ring R. In this case the problem is undecidable for $n = 4$ and $R = \mathbb{Z}$ (see [19]).

3 Cryptographical generation of groups

3.1 A general scheme. We begin with a general scheme to construct a vast family of groups and homomorphisms supporting both key agreement protocols of Section 1 and homomorphic cryptosystems of Section 2. Let $\mathcal{G}$ be a class of groups closed with respect to a set $\mathcal{O}$ of group-theoretical operations of different arities (like direct or wreath products). For an integer $s \geq 1$ we denote by $\mathcal{O}_s$ a set of all operations of arity s belonging to $\mathcal{O}$. For a set $\mathcal{G}_0 \subset \mathcal{G}$ (playing the role of a starting family of groups in the construction) we define recursively a class $\mathcal{P}(\mathcal{G}_0, \mathcal{O})$ of pairs (G, T) where $G \in \mathcal{G}$ and T is a rooted labeled tree, as follows:

Base of recursion: any pair (G, T) with $G \in \mathcal{G}_0$ and T being the one-point tree with root labeled by G, belongs to $\mathcal{P}(\mathcal{G}_0, \mathcal{O})$.

Recursive step: given pairs $(G_1, T_1), \ldots, (G_s, T_s) \in \mathcal{P}(\mathcal{G}_0, \mathcal{O})$ and an operation $o \in \mathcal{O}_s$, the class $\mathcal{P}(\mathcal{G}_0, \mathcal{O})$ contains the pair (G, T) where $G = o(G_1, \ldots, G_s)$ and T is the tree obtained from $T_1, \ldots, T_s$ by adding a new root labeled by o and its sons being the roots of $T_1, \ldots, T_s$.

Let $(G, T) \in \mathcal{P}(\mathcal{G}_0, \mathcal{O})$. Then obviously $G \in \mathcal{G}$ and the *derivation tree* T of G provides the constructive proof for this membership. The group G is uniquely determined by T and we call it the *group associated with* T. The fact, that a derivation tree is an ordinary rooted tree the leaves and the internal vertices of which are labeled by elements of $\mathcal{G}_0$ and $\mathcal{O}$, respectively, enables us to choose a random derivation tree of a fixed size.

Suppose from now on that all the groups of $\mathcal{G}$ are given in a certain way (e.g., one can take as $\mathcal{G}$ a class of matrix groups given by generator sets). We assume also that for each operation $o \in \mathcal{O}_s$ and groups $G_1, \ldots, G_s \in \mathcal{G}$, the size $L(G)$ of the presentation of the group $G = o(G_1, \ldots, G_s)$ is at most $O(L)$ where $L = \sum_{i=1}^{s} L(G_i)$ and the group G can be constructed from $G_1, \ldots, G_s$ in time $L^{O(1)}$.

Remark 3.1 *Thus the set of generators of G is assumed to be efficiently constructed; for instance, in the case of semidirect products (including both direct and wreath products considered below), this set is obtained by means of union of the generator sets of the operands.*

Let us define a size $L(T)$ of a derivation tree T to be the sum of the sizes of all labels of T; thus $L(T)$ includes the sizes of the groups assigned to the leaves of T together with

the number of edges of T. Then the size of any pair $(G,T) \in \mathcal{P}(\mathcal{G}_0, \mathcal{O})$ is $O(L(T))$, and the knowledge of T enables us to find G in time polynomial in $L(T)$.

One of the problems arising in constructions of group-theoretical public-key cryptosystems is to find an efficient algorithm to produce a random group (or a collection of groups) belonging to a special class $\mathcal{G}$ and with a given size L of the presentation. Such a group G must be equipped with a private key providing an efficient solution of a certain computational problem for G that is supposedly difficult in the class $\mathcal{G}$ without knowledge of a private key. Our approach to the above problem is to choose an appropriate class $\mathcal{G}_0$ of groups, a set $\mathcal{O}$ of group-theoretical operations, and then to generate instances for the cryptosystem in question as follows:

Step 1: given a security parameter L choose randomly groups $G_1, \ldots, G_t \in \mathcal{G}_0$, such that $\sum_{i=1}^{t} L(G_i) = O(L)$;

Step 2: choose randomly a rooted labeled tree T of size $O(L)$ and with t leaves being labeled by $G_1, \ldots, G_t$;

Step 3: compute the group G associated with T (i.e. $(G,T) \in \mathcal{P}(\mathcal{G}_0, \mathcal{O})$);

Step 4: output the group G as a public key and the labeled tree T as a secret key.

Denote by $\mathcal{G}^*$ the class of groups G such that $(G,T) \in \mathcal{P}(\mathcal{G}_0, \mathcal{O})$ for some labeled tree T. Then the secrecy of the key T is based on the intractability of the following problem: given $G \in \mathcal{G}^*$ find a derivation tree T associated with G. A special case of this problem will be considered in Section 3.3.

For a homomorphic cryptosystem the above scheme is not sufficient because together with the group G we have to provide a group H and a secret homomorphism $f : G \to H$. To this end suppose that each group $G \in \mathcal{G}_0$ is equipped with a set $M(G)$ of homomorphisms $f : G \to H$ with $H \in \mathcal{G}_0$. We also assume that the following property holds:

Compatibility. For any operation $o \in \mathcal{O}_s$ and groups $G_i, H_i \in \mathcal{G}^*$, $i = 1, \ldots, s$, one can efficiently construct monomorphisms $\eta_i : G_i \to G$ and $\xi_i : H_i \to H$ where $G = o(G_1, \ldots, G_s)$ and $H = o(H_1, \ldots, H_s)$ such that given epimorphisms $f_i : G_i \to H_i$ there exists an efficiently computed homomorphism $f : G \to H$ for which the equality $f \circ \eta_i = \xi_i \circ f_i$ holds for all i.

The constructed homomorphism is denoted by $o(f_1, \ldots, f_s)$. In this notation the set $M(\mathcal{G}_0, \mathcal{O})$ of instances f for a homomorphic cryptosystem can be defined recursively as follows:

Base of recursion: $M(G) \subset M(\mathcal{G}_0, \mathcal{O})$ for all $G \in \mathcal{G}_0$.

Recursion step: $o(f_1, \ldots, f_s) \in M(\mathcal{G}_0, \mathcal{O})$ for all $o \in \mathcal{O}_s$ and $f_1, \ldots, f_s \in M(\mathcal{G}_0, \mathcal{O})$.

We observe, that in the process of constructing the homomorphism $f : G \to H$ we also

produce the derivation trees of the groups G and H. Obviously, these trees are isomorphic as unlabelled trees. We associate with f its derivation tree $\overrightarrow{T}$ which is constructed in a similar way as the derivation tree T of G. In fact, $\overrightarrow{T}$ is obtained from T by changing the labels of its leaves: a leaf of T with a label G_0 gets the label $f_0 \in M(G_0)$ corresponding to the choice of the homomorphism in the base of recursion.

Concerning a presentation of constructed homomorphisms we need to guarantee that properties (H1), (H2) and (H3) hold. To this end we assume that they hold for homomorphisms belonging to $M(G)$ for all $G \in \mathcal{G}_0$ and that they are preserved by operations from $\mathcal{O}$.

A realization of the exposed general schemes in finite matrix groups will be considered in the next subsection.

3.2 Generating matrix groups. Let us define the classes $\mathcal{G}_0, \mathcal{G}$ of groups and the set $\mathcal{O}$ of operations. First, we set

$$\mathcal{G} = \bigcup_{n,R} \{G : \ G \text{ is a subgroup of } \mathrm{GL}(n, R)\}$$

where n and R run over natural numbers and finite commutative rings, respectively. Thus, any $G \in \mathcal{G}$ is a group of $n \times n$ invertible matrices with entries belonging to R for some $n \in \mathbb{N}$ and some finite commutative ring R. We recall that any such ring is a direct sum of local commutative rings and each of the latter can be described via appropriate Galois ring: the Galois ring $\mathrm{GR}(p^m, r)$ of characteristic p^m and rank r is $\mathbb{Z}_{p^m}[x]/(f)$ where $f \in \mathbb{Z}_{p^m}[x]$ is a monic polynomial of degree r whose image in $\mathbb{Z}_p[x]$ is irreducible (see [18]). We note that $\mathrm{GR}(p^m, r)$ is a local ring whose radical $\mathrm{Rad}(\mathrm{GR}(p^m, r))$ equals to (p).

Proposition 3.2 [18, 26] *Let R be a finite commutative local ring of characteristic p^m and $\mathbb{F} = \mathrm{GF}(p^r)$ the residue field of R. Then*

(1) *$R^\times = \mathcal{T} \times (1_R + \mathrm{Rad}(R))$ where $\mathcal{T}$ is a cyclic group isomorphic to $\mathbb{F}^\times$,*

(2) *the subring R_0 of R generated by $\mathcal{T}$ is a Galois ring $\mathrm{GR}(p^m, r)$,*

(3) *R is a homomorphic image of the ring $R_0[X_1, \ldots, X_t]$ where t is the minimal size of a generator set of the radical of R.*■

Proposition 3.3 [18] *Let p be a prime and m, r be natural numbers. Then*

(1) *there exists the unique up to isomorphism Galois ring $\mathrm{GR}(p^m, r)$ of characteristic p^m and rank r,*

(2) *each element x of $\mathrm{GR}(p^m, r)$ is uniquely represented in the form $x = \sum_{i=0}^{m-1} t_i p^i$ where $t_i \in \mathcal{T} \cup \{0\}$ for all i,*

(3) *given* $\overline{\sigma} \in \mathrm{Aut}(\mathbb{F})$ *the mapping* $x \mapsto \sum_{i=0}^{m-1} t_i^{\sigma} p^i$ *where* σ *is the automorphism of the group* $\mathcal{T}$ *induced by* $\overline{\sigma}$ *(see statement (1) of Proposition 3.2), is an automorphism of* $\mathrm{GR}(p^m, r)$.■

To construct a pool of finite commutative rings R one can start with the ring $R = \mathbb{Z}_m$ (as the recursion base) and to extend it repeatedly, for example, by taking of:

(R1) the group ring $R[G]$ for a finite commutative group G,

(R2) the quotient ring $R[X]/(\lambda)$ for a univariate polynomial $\lambda \in R[X]$.

In particular, construction (R2) produces all the Galois rings. We also remark that since the factorization of the characteristic of the resulting ring is not given, the decomposition in local summands is not presented explicitly.

We define a set $\mathcal{G}_0 \subset \mathcal{G}$ to be a class of classical simple (including abelian) subgroups G of the groups $\mathrm{GL}(n, \mathbb{F})$ where $n \in \mathbb{N}$ and $\mathbb{F}$ is a finite field. Any such group $G \in \mathcal{G}_0$ is given by a set of generators; for an abelian group of a prime order p one can use, e.g. its two-dimensional representation

$$\mathbb{Z}_p^+ \to \mathrm{GL}(2, p), \quad x \mapsto \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} \tag{3}$$

In fact, it is not necessary that $\mathcal{G}_0$ contains all classical groups; one can form $\mathcal{G}_0$ from the group of special types, e.g. $\mathrm{PSL}(n, \mathbb{F})$ or something like that. Since the elements of $\mathcal{G}_0$ are parametrized by the tuples of natural numbers, one can efficiently choose a random group $G \in \mathcal{G}_0$ with a given size $L(G)$ of presentation.

The choice of the set $\mathcal{O}$ of operations was inspired by the Aschbacher theorem [2] on classifying maximal subgroups of classical groups. Let us describe these operations.

Changing the underlying ring. Let R be a finite commutative ring and R' be an extension of R. Then the natural monomorphism

$$\varphi : \mathrm{GL}(n, R) \to \mathrm{GL}(n, R')$$

gives an unary operation in $\mathcal{G}$ taking $G \in \mathcal{G}$ to $\varphi(G)$. This operation can be performed efficiently whenever e.g. the embedding R to R' is given explicitly and the number $d = [(R')^+ : R^+]$ is small. As possible constructions of extensions we suggest the extension of $\mathbb{Z}_m$ to $\mathbb{Z}_{m'}$ where m is a divisor of m', and the ones described above in (R1) and (R2).

Conversely, any embedding of the ring R' into the ring $\mathrm{Mat}(d, R)$ induces the natural monomorphism

$$\varphi' : \mathrm{GL}(n, R') \to \mathrm{GL}(nd, R)$$

taking a matrix of $\mathrm{GL}(n, R')$ to the block matrix of $\mathrm{GL}(nd, R)$ with d^2 blocks of size n. As possible constructions of embeddings we suggest the natural embedding of a field of

the order q^d to $\mathrm{Mat}(d,q)$, or the direct sum of d copies of R to $\mathrm{Mat}(d,R)$, or the natural embeddings arising from constructions (R1) and (R2). This produces another unary operation in $\mathcal{G}$ taking $G \in \mathcal{G}$ to $\varphi'(G)$. In order not to blow up the size of representation one should assume that d is small.

In both cases the isomorphism type of the group G (as an abstract group) is not changed, but the operations change it as a linear group.

Direct products. Suppose that groups $G_1, \ldots, G_s \in \mathcal{G}$ are such that $G_i \leq \mathrm{GL}(n_i, R)$ where $n_i \in \mathbb{N}$ and R is a finite commutative ring. Then

$$G = G_1 \otimes \cdots \otimes G_s \leq \mathrm{GL}(n, R)$$

where $n = \prod_{i=1}^{s} n_i$, and we obtain an s-ary operation in $\mathcal{G}$. A set of generators for the group G can be efficiently constructed from the generating sets for $G_1, \ldots, G_s$ by means of the Kronecker product of the corresponding matrices. When R is a field the group G is irreducible iff so are the groups $G_1, \ldots, G_s$. (A matrix group G is called irreducible if the underlying linear space contains no nontrivial G-invariant subspaces.)

Similarly, if $m = n_i$, $G_i \cap G_i' = \{I_m\}$ and G_i' normalizes G_i for all $i = 1, \ldots, s$ where G_i' is the group generated by G_j, $j \neq i$, then $G_1 \times \cdots \times G_s$ is a subgroup of $\mathrm{GL}(m, R)$ which gives one more s-ary operation.

Wreath products. The *wreath product* $G \wr \Gamma$ of a group G and a permutation group $\Gamma \leq \mathrm{Sym}(m)$ is defined to be the semidirect product of the m-fold direct product $G^m = G \times \cdots \times G$ by the group Γ acting on G^m via coordinatewise permutations. If $G \leq \mathrm{GL}(n, R)$, then the group $G \wr \Gamma$ has two natural linear representations obtained from the natural monomorphisms

$$G^m \to \mathrm{GL}(nm, R), \qquad G^m \to \mathrm{GL}(n^m, R),$$

the first of which is induced by the m-fold direct sum of the underlying R-module, whereas the second one is induced by the m-fold tensor product of it. The images of the group $G \wr \Gamma$ under these two monomorphisms are called the *imprimitive* and the *product actions* of the wreath product, respectively. Thus, we obtain two more efficiently computable unary operations in $\mathcal{G}$ for each permutation group Γ. For our purpose it is enough to set Γ to be the symmetric group $\mathrm{Sym}(m)$. (More elaborated way could be based on the fact that any transitive group is obtained from the action of a group on the set of right cosets of some subgroup by means of right multiplications.) In the case of R being a field the resulting groups are always irreducible whenever G is irreducible and Γ is transitive.

Conjugations. An obvious unary operation in $\mathcal{G}$ consists in the conjugation of a group $G \subset \mathrm{GL}(n, R)$ by means of a randomly chosen matrix from $\mathrm{GL}(n, R)$. Such an operation enables us to hide the form of a generator set of the group G.

Let $\mathcal{O}$ be the set of the above operations and $\mathcal{G}^* \subset \mathcal{G}$ be the set of all groups G such that $(G, T) \in \mathcal{P}(\mathcal{G}_0, \mathcal{O})$ for some rooted labeled tree T (see Subsection 3.1). In the following statement we consider the specializations of the problems MT (see Section 2) and LTP (see Subsection 1.3) for the class $\mathcal{G}^*$. In both cases we suppose that the group $G \in \mathcal{G}^*$ is given by a set of generators. If $G \leq \mathrm{GL}(n, R)$ for a certain $n \in \mathbb{N}$ and for a finite commutative ring R, then in the case of LTP we set V to be the standard free R-module of dimension n on which the group $\mathrm{GL}(n, R)$ acts, whereas for MT problem we set $F = \mathrm{GL}(n, R)$.

Lemma 3.4 *Let $G \in \mathcal{G}^*$. Then given a derivation tree of G the problems MT and LTP can be solved in time polynomial in $L(G)$.*

Proof. Let T be a derivation tree of G. Then the labels of the leaves of T are the groups $G_1, \ldots, G_t \in \mathcal{G}_0$. Due to the choice of $\mathcal{G}_0$ the problems MT and LTP can be solved for the group G_i in time polynomial in $L(G_i)$ for $i = 1, \ldots, t$. (Indeed, any nonabelian classical matrix group is given together with a suitable matrix representation which can be used for testing membership; for an abelian group representation (3) provides a trivial membership testing algorithm).

Since $L(G) = L(T)^{O(1)}$, it suffices to verify that by means of the construction of the tree T the problems can be reduced in time $L(T)^{O(1)}$ to the corresponding problems for $G_1, \ldots, G_t$. For this purpose let us consider, for instance, the reduction in the case of the primitive wreath product $G = H \wr \Gamma$ with $H \leq \mathrm{GL}(n, R)$ and $\Gamma = \mathrm{Sym}(m)$ (other operations from $\mathcal{O}$ on groups are treated in a similar way). Then $G \leq \mathrm{GL}(n^m, R)$ and since T is given, we know the decomposition

$$V = U \otimes \cdots \otimes U \quad (m \text{ times})$$

where V and U are the standard R-modules for groups $\mathrm{GL}(n^m, R)$ and $\mathrm{GL}(n, R)$, respectively. Any element $g \in G$ can be represented as a pair $(h, k) \in H^m \times \mathrm{Sym}(m)$ such that

$$(u_1, \ldots, u_m)^g = (u_{i_1}^{h_{i_1}}, \ldots, u_{i_m}^{h_{i_m}}) \tag{4}$$

where $h = (h_1, \ldots, h_m)$ and $i_j = j^{k^{-1}}$ for $j = 1, \ldots, m$. Now the permutation k can be efficiently computed from the elements of the form $(0_R, \ldots, 1_R, \ldots, 0_R)^g$ (with 1_R being the unique nonzero component in a certain place). So, the element $h = gg_k^{-1}$ also can be found efficiently where g_k is the element of $\mathrm{GL}(V) = \mathrm{GL}(n^m, R)$ corresponding to k (this element acts on V exactly by permuting coordinates according to k). In particular, this provides a polynomial time reduction of the MT problem for G to the corresponding problem for H.

Next, proceeding to the LTP problem let $v \in u^G$ for some $u, v \in V$. Denote by D the bipartite graph with parts being the multisets $\{u_1, \ldots, u_m\}$ and $\{v_1, \ldots, v_m\}$ and the

edges being the pairs (u_i, v_j) for which $v_i \in (u_j)^H$. Then from (4) it follows that there is a one to one correspondence between the matchings $\{(u_i, v_{j_i}) :\ i = 1, \dots, m\}$ of the graph D and the set $\{k \in \Gamma :\ v = u^g$ with $g = (h, k) \in G$ for some $h \in H^m\}$. Since the problem of finding a matching of a bipartite graph can be solved efficiently, we see that the LTP problem for G is polynomial time reducible to the corresponding problem for H.■

A natural way to apply our construction to the key agreement protocol is to choose a random group $G \in \mathcal{G}^*$ of a prescribed size and then choose random subgroups G_A and G_B of G (see (1)). These groups can be specified by sets of generators constructed as follows:

Step 1. Let $\mathcal{L}$ be the set of leaves of the derivation tree T of the group G. For each $l \in \mathcal{L}$ take random subsets $X_A(l)$ and $X_B(l)$ of the group H_l associated with l.

Step 2. Construct the trees T_A and T_B obtained from T only relabelling of the leaves: the leaf $l \in \mathcal{L}$ is labeled by the group $H_A(l) = \langle X_A(l) \rangle$ in T_A and by the group $H_B(l) = \langle X_B(l) \rangle$ in T_B respectively.

Step 3. Set G_A and G_B to be the groups the derivation trees of which are T_A and T_B (the generator sets X_A of G_A and X_B of G_B are obtained in accordance with Remark 3.1 on constructing the generators).

Thus, the constructing of the groups G_A and G_B is performed simultaneously with the constructing the group G. (In fact, all we need, is the embedding of each group assigned to a leaf of the derivation tree of the group G into G.) In this way it is possible to control some properties of the groups, for instance, to avoid the situation when G_A centralizes G_B (then the common key coincides with 1_G and so is not secure).

Applying our construction to design homomorphic cryptosystems is more delicate. First of all we define the set $M(G)$ for each group $G \leq \mathrm{GL}(n, R)$ for some $n \in \mathbb{N}$ and some finite commutative ring R (note that this covers the case $G \in \mathcal{G}_0$ and also allows one to produce homomorphisms in one more way: replacing $\mathcal{G}_0$ by a bigger subclass of $\mathcal{G}$). Namely, any automorphism $\sigma \in \mathrm{Aut}(R)$ induces a homomorphism

$$f_\sigma : G \to G^\sigma,\ A \mapsto A^\sigma$$

where the matrix $A^\sigma \in \mathrm{GL}(n, R)$ is obtained from the matrix $A \in \mathrm{GL}(n, R)$ by entry-wise applying of σ. To choose σ we observe that $R = \oplus_{i \in I} R_i$ where each R_i is a finite local commutative ring. Any automorphism of the residue field of the ring R_i can be lifted to the automorphism of this ring (statement (3) of Proposition 3.3). In the representation of the Galois ring as a quotient ring of a ring of polynomials this lifting can be done efficiently. Taking any collection $\{\sigma_i\}_{i \in I}$ one can construct the automorphism $\sigma \in \mathrm{Aut}(R)$ such that $\sigma|_{R_i} = \sigma_i$ for all i. The set of such automorphisms we denote by $\mathrm{Aut}_0(R)$ (in the case of R being a field this group coincides with $\mathrm{Aut}(R)$). Set

$$M(G) = f_0 \cup \{f_\sigma :\ \sigma \in \mathrm{Aut}_0(R)\} \tag{5}$$

where f_0 is a trivial homomorphism taking any element of G to the identity matrix of $\mathrm{GL}(n, R)$. We assume that each $f \in M(G)$ is given by the images of generators of the group G and hence conditions (H1), (H2) hold trivially, while condition (H3) is obvious for f_σ and follows from the choice of presentation of G (by generators) for f_0. Then assuming that the ring R is given explicitly, one can choose a random element of $M(G)$ in time polynomial in $L(G)$.

To provide the recursive step in constructing a homomorphism $f \in M(\mathcal{G}_0, \mathcal{O})$ it suffices to verify the compatibility property for the set $\mathcal{O}$ of the operations (see Subsection 3.1) and to verify that the operations preserve properties (H2) and (H3) (see Section 2). However, the compatibility property is obviously fulfilled if the required monomorphisms η_i for f are chosen to be

- identical in case of the operation *changing the underlying ring*,
- the embedding $g_i \mapsto 1_{G_1} \otimes \cdots \otimes g_i \otimes \cdots \otimes 1_{G_s}$ in case of the operation *direct product*,
- the embedding $g \mapsto (g, \ldots g; 1_\Gamma)$ in case of the operation *wreath product*,
- the isomorphism $g \mapsto a^{-1}ga$ in case of the operation of *conjugation* by a.

(The monomorphisms ξ_i are defined in a similar way.)

Concerning properties (H2) and (H3) we note that they are obvious for the operations changing the underlying ring and conjugation. In the case of the direct product it suffices to note that a generator $1_{G_1} \otimes \cdots \otimes g_i \otimes \cdots \otimes 1_{G_s}$ of the group G (where g_i is a generator of G_i) is mapped under f to $1_{G_1} \otimes \cdots \otimes f_i(g_i) \otimes \cdots \otimes 1_{G_s}$, and $\ker(f) = \ker(f_1) \otimes \cdots \otimes \ker(f_s)$. In the case of the wreath product generators $(1_{G_1}, \ldots, g_i, \ldots, 1_{G_s}; 1_\Gamma)$ and $(1_{G_1}, \ldots, 1_{G_s}; \gamma)$ (where γ is a generator of Γ) of the group G are mapped under f to $(1_{G_1}, \ldots, f_i(g_i), \ldots, 1_{G_s}; 1_\Gamma)$ and to $(1_{G_1}, \ldots, 1_{G_s}; \gamma)$, respectively, and finally $\ker(f) = (\ker(f_1), \cdots, \ker(f_s); 1_\Gamma)$.

Thus, in all the cases, the resulting homomorphism is efficiently computable. The above discussion shows that the following statement holds.

Lemma 3.5 *Let $f : G \to H$ be a homomorphism constructed in the above way where $G, H \in \mathcal{G}^*$. Then given a derivation tree $\overrightarrow{T}$ of f (see the end of Subsection 3.1) one can find $f(g)$ for $g \in G$ in time polynomial in $L(G)$ and in the size of g.*■

3.3 Secure generation. Let us fix the classes $\mathcal{G}_0, \mathcal{G}, \mathcal{G}^*$, the set $\mathcal{O}$ of operations and the sets $M(G)$ for $G \in \mathcal{G}_0$ as in Subsection 3.2. Then due to Lemmas 3.4 (resp. Lemma 3.5) one can construct groups $G \in \mathcal{G}^*$ (resp. homomorphisms $f \in M(\mathcal{G}_0, \mathcal{O})$) to realize key agreement protocols (resp. homomorphic cryptosystems) in which the group G (resp. the homomorphism f) and the derivation tree T of G (resp. $\overrightarrow{T}$ of f) play the roles of public

and secret keys, correspondingly. The security of such systems is based on the difficulty of the following problem.

Decomposition Problem. *Given a group $G \in \mathcal{G}^*$ (resp. a homomorphism $f \in M(\mathcal{G}_0, \mathcal{O})$) find a derivation tree T of G (resp. $\overrightarrow{T}$ of f).*

This problem arises in connection with a computational version of the above mentioned Aschbacher's theorem. A number of practical algorithms (without complexity bounds) for Decomposition Problem are known (see [15]), but in general this problem seems to be difficult. Indeed, suppose that $R = \mathbb{Z}_m$ where $m = pq$ with p and q being two different primes. Denote by G_p the cyclic matrix group of the order p in $\mathrm{GL}(2, p)$ (see (3)). Similarly, the group G_q is defined. Then $G_p, G_q \in \mathcal{G}_0$ and

$$G = G'_p \times G'_q \leq \mathrm{GL}(2, R)$$

where G'_p and G'_q are the images of the groups G_p and G_q with respect to the natural embeddings $\mathrm{GL}(2, p)$ and $\mathrm{GL}(2, q)$ into $\mathrm{GL}(2, R)$. Thus, the group G can be constructed in two steps: first one constructs the groups G'_p and G'_q (the operation of changing the underlying ring), and then one sets $G = G'_1 \times G'_2$ (the operation of the direct product). This implies that $G \in \mathcal{G}^*$. This shows that the integer factoring problem is a special case of the Decomposition Problem.

Another strategy of an adversary Charlie could be to avoid solving the Decomposition Problem and to try to solve the problems like LTP, SCSP or CMT directly. To prevent such an attack one can choose the leaves of a derivation tree of the group G to be the groups of the size exponential with respect to $L(G)$. Then from the construction it follows that these groups will appear as the composition factors of G. However, for the groups with large composition factors all the problems like LTP, SCSP or CMT seem to be difficult (cf. Subsection 1.3 and Section 2).

We mention one more attack of Charlie for the case of a homomorphic cryptosystem. Suppose we construct in the above way the homomorphism $f : G \to H$ with $G, H \in \mathcal{G}^*$. We call the homomorphism *linear* if it induces the ring homomorphism $f' : A(G) \to A(H)$ where $A(G)$ (resp. $A(H)$) is the subring of the underlying full matrix ring generated by G (resp. H). For a linear homomorphism the corresponding homomorphic cryptosystem can be easily broken whenever $G \leq \mathrm{GL}(n, R)$ where $R = \mathbb{Z}_n$ for some $n \in \mathbb{N}$ or R is a finite field (or, more generally, a direct sum of Galois rings). Indeed, in this case Charlie can find $f(g)$ for $g \in G$ as follows. Take random generators $g_1, \ldots, g_s$ of the group G and find a decomposition $g = \sum_{i=1}^{s} c_i g_i$ with $c_i \in R$ just involving linear algebra. Then $f(g) = \sum_{i=1}^{s} c_i f(g_i)$ due to the linearity of f. To prevent this attack one can take some initial homomorphisms at the leaves of the derivation tree to be elements of the group $\mathrm{Aut}_0(R)$ (see (5)). Then the constructed homomorphism is not linear in general (e.g. if $g \in \mathrm{GL}(n, \mathbb{F})$ with $\mathbb{F}$ being a field, and $\sigma \in \mathrm{Aut}(\mathbb{F})$, then generally $(ag)^\sigma \neq ag^\sigma$).

We conjecture that two-party key agreement protocol and homomorphic public-key cryptosystem based on the constructed class of matrix groups over finite commutative rings are secure (as mass problems). If the latter was true, then one could construct encrypted simulation of a boolean circuit of the logarithmic depth (the details can be found in [11]).

Final remarks

One of the main problems in constructing homomorphic public-key cryptosystems consists in finding appropriate trapdoor functions. However, in the natural presentations of homomorphisms of algebraic structures the problem of breaking such a system is reduced to some variants of the CMT problem. On the other hand, there is the following result for matrix groups over finite fields.

Theorem 3.6 [13, Theorem 6.1] *Given* $K = \langle X \rangle \leq \mathrm{GL}(d, p^e)$ *where* $X \subset \mathrm{GL}(d, p^e)$, *there is a Las Vegas algorithm that given any* $g \in \mathrm{GL}(d, p^e)$, *decides whether* $g \in K$, *and if* $g \in K$, *then the algorithm produces a straight-line program with the input* X, *yielding* g. *The algorithm uses an oracle to compute discrete logarithms in fields of characteristic* p *with sizes up to* p^{ed}. *In case when all of those composition factors of Lie type in characteristic* p *are constructively recognizable with a Discrete Log oracle* [1], *the running time is a polynomial in the input length* $|X|d^2e \log p$, *plus the time required for polynomially many calls to the Discrete Log oracle.*■

This theorem shows that having an oracle for the Discrete Logarithm, the membership problem can be solved in probabilistic polynomial time for matrix groups over finite fields. This means that at least for homomorphic public-key cryptosystems over such groups there is a little hope to find a trapdoor function different from functions the difficulty of inversion of which is based on the intractability of the Discrete Logarithm. However, only a little is known on the computational complexity of the membership problem for matrix groups over rings. So, constructions over such groups seem to be more perspective from the point of view of algebraic (non-commutative) cryptography.

References

[1] I. Anshel, M. Anshel, D. Goldfeld, *An algebraic method for public-key cryptography*, Mathematical Research Letters, **6** (1999), 287–291.

[1] The current list of groups of Lie type recognizable with a Discrete Log oracle is given in [13]; this list includes the groups of series A, B, C, D.

[2] M. Aschbacher, *On the maximal subgroups of the finite classical groups*, Invent. Math. 76 (1984) 469–514.

[3] R. Beals, L. Babai, *Las Vegas algorithms for matrix groups*, Proc. 34th IEEE FOCS, 1993, 427–436.

[4] A. Chistov, G. Ivanyos, M. Karpinski, *Polynomial time algorithms for modules over finite dimensional algebras*, Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation (Kihei, HI), 68–74, ACM, New York, 1997.

[5] Do Long Van, A. Jeyanthi, R. Siromoney, K. G. Subramanian, *Public key cryptosystem based on word problems*, ICOMIDC Symp. Math. of Computation, Ho Chi Minh City, April, 1988.

[6] B. Eick, D. Kahrobaei, *Polycyclic groups: a new platform for cryptology?*, `arXiv.math.GR/0411077`.

[7] J. Feigenbaum, M. Merritt, *Open questions, talk abstracts, and summary of discussions*, DIMACS series in discrete mathematics and theoretical computer science, **2** (1991), 1–45.

[8] S. Goldwasser, M. Bellare, *Lecture Notes on Cryptography*, http://www-cse.ucsd.-edu/users/mihir/papers/gb.html, 2001.

[9] S. Goldwasser. S. Micali, *Probabilistic encryption*, J.Comput.Syst.Sci., **28** (1984), 270–299.

[10] D. Grigoriev, *Public-key cryptography and invariant theory.*, J. Math. Sci., **126** (2005), 1152–1157.

[11] D. Grigoriev, I. Ponomarenko, *Homomorphic public-key cryptosystems and encrypting boolean circuits*, to appear in Appl. Alg. Eng. Communic. Comput., 2006.

[12] D. Grigoriev, I. Ponomarenko, *Homomorphic public-key cryptosystems over groups and rings*, Quaderni di Matematica, **13** (2004), 305–325.

[13] W. M. Kantor, A. Seress, *Computing with matrix groups*, Groups, combinatorics & geometry (Durham, 2001), 123–137, World Sci. Publishing, River Edge, NJ, 2003.

[14] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang, C. Park, *New public-key cryptosystem using braid groups*, Lecture Notes in Computer Science, **1880** (2000), 166–183.

[15] C. R. Leedham-Green, *The computational matrix group project*, pp. 229–247 in: Groups and Computation III (eds. W. M. Kantor and A. Seress), The Ohio State Univ. Math. Res. Inst. Publ. 8, Walter deGruyter, BerlinNew York 2001.

[16] E. M. Luks, *Permutation groups and polynomial-time computation*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, **11** (1993), 139–175.

[17] W. Magnus, A. Karrass, D. Solitar, *Combinatorial group theory: Presentations of groups in terms of generators and relations*, Interscience Publishers, New York-London-Sydney, 1966.

[18] B. R. MacDonald, *Finite Rings with Identity*, New York, Marcel Dekker, 1974.

[19] K. A. Mihailova, *The occurrence problem for a direct product of groups*, Dokl. Akad. Nauk, **119** (1958), 1103–1105. (in Russian)

[20] T. Miyazaki, *Polynomial-time computation in matrix groups*, Ph.D. dissertation, Tech. Rep. CISTR9911, Department of Computer and Information Science, University of Oregon, Eugene, 1999.

[21] D. Naccache, J. Stern, A new public-key cryptosystem based on higher residues, Proc. 5th ACM Conference on Computer and Communication Security, 1998, 59–66.

[22] S.-H. Paeng, D. Kwon, K.-C. Ha, J. H. Kim, *Improved public-key cryptosystem using finite non-abelian groups*, IACR ePrint 2001/066.

[23] R. Rivest, L. Adleman, M. Dertouzos, *On data banks and privacy homomorphisms*, Found. of Secure Computations, Academic Press, 1978, 169–179.

[24] V. Shpilrain, A. Ushakov, *The conjugacy search problem in public-key cryptography: unnecessary and insufficient*, to appear in Appl. Alg. Eng. Communic. Comput., 2006.

[25] D. A. Suprunenko, *Matrix groups*, AMS, Providence, 1976.

[26] Z.-X. Wan, *Lectures on finite fields and Galois rings*, World Scientific Publishing Co., Inc., River Edge, NJ, 2003.