Homomorphic public-key cryptosystems and encrypting boolean circuits

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

Ilia Ponomarenko *
Steklov Institute of Mathematics,
Fontanka 27, St. Petersburg 191011, Russia
inp@pdmi.ras.ru
http://www.pdmi.ras.ru/~inp

13.09.2004

Abstract

Given an arbitrary finite nontrivial group we describe a probabilistic public-key cryptosystem in which the decryption function is chosen to be a suitable epimorphism from the free product of finite abelian groups onto this finite group. It extends the quadratic residue cryptosystem (based on a homomorphism onto the group of two elements) due to Rabin-Goldwasser-Micali. The security of the cryptosystem relies on the intractability of factoring integers. As an immediate corollary of the main construction we obtain a more direct proof (based on the Barrington technique) of Sander-Young-Yung result on an encrypted simulation of a boolean circuit of the logarithmic depth.

1 Homomorphic cryptography over groups

The main purpose of the paper is to find probabilistic public-key schemes in which the encryption function has a homomorphic property. More precisely, we are interested in

*Partially supported by RFFI, grants, 03-01-00349, NSH-2251.2003.1, 02-01-00093.

a scheme in which the spaces of messages and of ciphertexts are groups H_k and G_k respectively, depending on a security parameter k, and the decryption functions $f_k : G_k \to H_k$ are epimorphisms. In such a system the public key includes a set of generators of the group $\ker(f_k)$ and a system R_k of distinct representatives of the group G_k by $\ker(f_k)$ (transversal for a short). The probabilistic encryption of a message $h \in H_k$ is performed by computing an element $gr_h \in G_k$ where $r_h \in R_k$ is such that $f_k(r_h) = h$, and g is a random element of $\ker(f_k)$. We call this probabilistic public-key scheme a *homomorphic cryptosystem* with respect to the epimorphisms f_k. The security of such a system is based on the intractability of deciding whether or not the element of G_k belongs to the normal subgroup $\ker(f_k)$ of G_k. The case of special interest is when the group H_k does not depend on the security parameter k; in this case we speak on the homomorphic cryptosystem over the group H. The general problem of constructing homomorphic cryptosystems goes back to [22] (see also [6]). Concerning public-key cryptosystems using groups (not necessary homomorphic ones) we refer to [2, 9, 10, 11, 13, 14, 16, 21, 22].

Let H be a finite nontrivial group. A general approach to construct a homomorphic cryptosystem over H can be explained as follows. Given a natural number k we find groups A_k and G_k and an *exact* sequence of group homomorphisms

$$A_k \xrightarrow{P_k} G_k \xrightarrow{f_k} H \to \{1\} \tag{1}$$

(recall that the exact sequence means that the image of each homomorphism in it coincides with the kernel of the next one) such that under Assumption 1.1 below the homomorphism P_k and the inverse to f_k are *trapdoor functions*. The latter means that one can efficiently compute $P_k(a)$, $a \in A_k$, and generate random elements of the set $f_k^{-1}(h)$, $h \in H$, while generating random elements of the set $P_k^{-1}(g)$, $g \in G_k$, as well as computing elements $f_k(g)$, $g \in G_k$, can be performed efficiently only by means of secret keys.

Assumption 1.1 *The problem* $\mathrm{TEST}(P_k)$ *of testing whether a given* $g \in G_k$ *belongs to* $\mathrm{im}(P_k) = \ker(f_k)$ *is intractable.*

In fact, this assumption implies that the homomorphic cryptosystem over the group H with respect to the homomorphisms f_k is semantically secure against a passive adversary (see [7] and the proof of Theorem 2.1 below) whereas the intractability of the following problem means that P_k is a trapdoor function.

Problem $\mathrm{INVERSE}(P_k)$**.** *Given* $g \in \mathrm{im}(P_k)$ *find a random element* $a \in A_k$ *such that* $P_k(a) = g$.

To our best knowledge all the considered so far homomorphic cryptosystems are more or less extensions of the following one. Let n be the product of two distinct large primes of (bit-)size $k = O(\log n)$. Set

$$A_k = \mathbb{Z}_n^*, \quad G_k = \{g \in \mathbb{Z}_n^* : \ \mathbf{J}_n(g) = 1\}, \quad H = \mathbb{Z}_2^+ \tag{2}$$

where J_n denotes the Jacobi symbol. Then together with the natural homomorphisms $P_k : A_k \to G_k$ and $f_k : G_k \to H$ induced by the squaring function, these data define a homomorphic cryptosystem over H (see [8, 7, 15]). (In this case computing f_k^{-1} is provided by a fixed non-square of G.) We call it the *quadratic residue cryptosystem*. The security of this scheme is based on the quadratic residue assumption for the group G_k (see [8, 7, 15]). A generalization of the quadratic residue cryptosystem using m-residues for $m > 2$ was proposed in [2] (see also Section 2 below). For the Paillier cryptosystem from [19] we have

$$A_k = G_k = \mathbb{Z}_{n^2}^*, \quad H_k = \mathbb{Z}_n^+$$

with the same assumptions on n and k as in the quadratic residue cryptosystem and the corresponding homomorphisms P_k and f_k being induced by raising to the nth power. For the Okamoto-Uchiyama cryptosystem from [17] we have

$$A_k = G_k = \mathbb{Z}_{p^2q}^*, \quad H_k = \mathbb{Z}_p^+$$

where p, q are distinct large primes of the same size k, and again the corresponding homomorphisms P_k and f_k being induced by raising to the nth power where $n = pq$. Finally, we mention that homomorphic cryptosystems over certain dihedral groups were studied in [21].

The main result of the present paper consists in the construction of a homomorphic cryptosystem over an arbitrary finite nontrivial group H; the security of it is based on the assumption on the intractability of the following slight generalization of the factoring problem:

Problem FACTOR(n, m)**.** *Let $n = pq$ where p and q are primes of the same size. Suppose that $m > 1$ is a constant size divisor of $p - 1$ such that* $\mathrm{GCD}(m, q-1) = \mathrm{GCD}(m, 2)$*. Given a transversal of $(\mathbb{Z}_n^*)^m$ in the group $G_{n,m} = \{g \in \mathbb{Z}_n^* : \mathbf{J}_n(g) \in \{1, (-1)^m\}\}$, find the numbers p, q.*

First the main result is proved for a cyclic group H (see Section 2), in this case the groups G_k are finite and Abelian. Then in Section 3 a homomorphic cryptosystem is yielded for an arbitrary H, in this case the group G_k becomes a free product of certain Abelian groups produced in Section 2. In Section 4 we recall the result from [1] on a polynomial size simulation of any boolean circuit B of the logarithmic depth over an arbitrary unsolvable group H (in particular, one can take H to be the symmetric group Sym(5)). Combining this result with the homomorphic cryptosystem from Section 3 provides an *encrypted simulation* of B over the group G_k: the output of this simulation at a particular input is a certain element $g \in G_k$, and thereby to know the output of B one has to be able to calculate $f(g) \in H$, which is supposedly to be difficult due to Theorem 3.2. In contrast to a different approach to encrypt boolean circuits proposed in [24], our construction is more direct and allows one to accomplish the protocol called evaluating an encrypted circuit (see Section 4). Also the problem of encrypting boolean circuits is discussed in [21].

We complete the introduction by making some remarks concerning our construction and cryptosystems based on groups. First, we notice that in the present paper the group H is always rather small, while the groups G_k could be infinite but being always finitely generated. However, the infiniteness of G_k is not an obstacle for performing algorithms of encrypting and decrypting (for the latter using the trapdoor information) since G_k is a free product of groups of a number-theoretic nature like Z_n^*; therefore one can perform group operations in G_k efficiently and on the other hand this allows one to provide evidence for the difficulty of a decryption. In this connection we mention a public-key cryptosystem from [5] in which f_k was the natural epimorphism from a free group G_k onto the group H (infinite, non-abelian in general) given by generators and relations. In this case for any element of H one can produce its preimages (encryptions) by inserting in a word (being already a produced preimage of f_k) from G_k any relation defining H. In other terms, decrypting of f_k reduces to the word problem in H. In our approach the word problem is solvable easily due to a special presentation of the group G_k (rather than given by generators and relations). The same is true for the homomorphic cryptosystem of [10] where free groups were given as subgroups of modular groups.

Another idea of a homomorphic (in fact, isomorphic) encryption E (and a decryption $D = E^{-1}$) was proposed in [13]. Unlike our construction the encryption $E : G \to G$ is executed in the same set G (being an elliptic curve over the ring $\mathbb{Z}_n$) treated as the set of plaintext messages. If n is composite, then G is not a group while being endowed with a partially defined binary operation which converts G in a group when n is prime. The problem of decrypting this cryptosystem is close to the factoring of n. In this aspect [13] is similar to the well-known RSA scheme (see e.g. [7]) if to interpret RSA as a homomorphism (in fact, isomorphism) $E : Z_n^* \to Z_n^*$, for which the security relies on the difficulty of finding the order of the group Z_n^*.

Finally, we mention some other cryptosystems using groups. The well-known example is a cryptosystem which relies on the Diffie-Hellman key agreement protocol (see e.g. [7]). It involves cyclic groups and relates to the discrete logarithm problem [14]; the complexity of this system was studied in [3]. Some generalizations of this system to non-abelian groups (in particular, the matrix groups over some rings) were suggested in [18] where security was based on an analog of the discrete logarithm problems in groups of inner automorphisms. One more example is a cryptosystem from [16] based on a monomorphism $\mathbb{Z}_m^+ \to \mathbb{Z}_n^*$ by means of which x is encrypted by $g^x \pmod{n}$ where n, g constitute a public key; its decrypting relates to the discrete logarithm problem and is feasible in this situation due to a special choice of n and m (cf. also [2]). Certain variations of the Diffie-Hellman systems over the braid groups were described in [11]; there several trapdoor one-way functions connected with the conjugacy and taking root problems in the braid groups were proposed.

2 Homomorphic cryptosystems over cyclic groups

To make the paper selfcontained we describe below an explicit homomorphic cryptosystem over a cyclic group of an order $m > 1$ proposed in [2]. The decryption of it is based on taking m-roots in the group $\mathbb{Z}_n^*$ for a suitable $n \in \mathbb{N}$. It can be considered in a sense as a generalization of the quadratic residue cryptosystem over $\mathbb{Z}_2^+$ (see (2)). Throughout this section we denote by $|n|$ the bit size of a number $n \in \mathbb{N}$.

Given a positive integer $m > 1$ denote by D_m the set of all pairs (p, q) where p and q are distinct odd primes such that

$$p - 1 = 0 \pmod{m} \quad \text{and} \quad \mathrm{GCD}(m, q-1) = \mathrm{GCD}(m, 2). \tag{3}$$

Let $(p, q) \in D_m$, $n = pq$ and $G_{n,m}$ be a group defined by

$$G_{n,m} = \{g \in \mathbb{Z}_n^* : \ \mathbf{J}_n(g) \in \{1, (-1)^m\}\}. \tag{4}$$

Thus $G_{n,m} = \mathbb{Z}_n^*$ for an odd m and $[\mathbb{Z}_n^* : G_{n,m}] = 2$ for an even m. In any case this group contains each element $h = h_p \times h_q$ such that $\langle h_p \rangle = \mathbb{Z}_p^*$ and $\langle h_q \rangle = \mathbb{Z}_q^*$ where h_p and h_q are the p-component and the q-component of h with respect to the canonical decomposition $\mathbb{Z}_n^* = \mathbb{Z}_p^* \times \mathbb{Z}_q^*$. From (3) it follows that m divides the order of any such element h and $\{1, h, \cdots, h^{m-1}\}$ is a transversal of the group $G_{n,m}^m = \{g^m : \ g \in G_{n,m}\}$ in $G_{n,m}$. This implies that $G_{n,m}/G_{n,m}^m \cong \mathbb{Z}_m^+$ where the corresponding epimorphism is given by the mapping

$$f_{n,m} : G_{n,m} \to \mathbb{Z}_m^+, \quad g \mapsto i_g$$

with i_g being the element of $\mathbb{Z}_m^+$ such that $g \in G_{n,m}^m h^{i_g}$. From (3) it follows that $\ker(f_{n,m}) = G_{n,m}^m = \mathrm{im}(P_{n,m})$ where

$$P_{n,m} : A_{n,m} \to G_{n,m}, \quad g \mapsto g^m$$

is a homomorphism from the group $A_{n,m} = \mathbb{Z}_n^*$ to the group $G_{n,m}$. In particular, we have the exact sequence (1) with $A_k = A_{n,m}$, $P_k = P_{n,m}$, $f_k = f_{n,m}$, $G_k = G_{n,m}$ where $k = |p| = |q|$, and $H = \mathbb{Z}_m^+$. Next, it is easily seen that any element of the set

$$\mathcal{R}_{n,m} = \{R \subset G_{n,m} : \ |f_{n,m}(R)| = |R| = m\}$$

is a right transversal of $G_{n,m}^m$ in $G_{n,m}$. We notice that by the Dirichlet theorem on primes in arithmetic progressions (see e.g. [4]) the set D_m is not empty. Moreover, by the same reason the set

$$D_{k,m} = \{n \in \mathbb{N} : \ n = pq, \ (p, q) \in D_m, \ |p| = |q| = k\} \tag{5}$$

is also nonempty for sufficiently large $k \in \mathbb{N}$.

Let H be a cyclic group of order $m > 1$ (below without loss of generality we assume that $H = \mathbb{Z}_m^+$). We describe a probabilistic polynomial time algorithm which yields a certain $n \in D_{k,m}$. The algorithm picks randomly integers $p = 1 \pmod m$ and $q = -1 \pmod m$ from the interval $[2^k, 2^{k+1}]$ and tests primality of the picked numbers by means of e.g. [23]. According to [4] there is a constant $c > 0$ such that for any b relatively prime with m there are at least $c2^k/(\varphi(m)k)$ primes of the form $mx + b$ in the interval $[2^k, 2^{k+1}]$. Therefore, after $O(k)$ attempts the algorithm would yield a pair $(p, q) \in D_{k,m}$ with a probability greater than ϵ for a certain constant $0 < \epsilon < 1$. Thus given $k \in \mathbb{N}$ one can design in probabilistic time $k^{O(1)}$ a number $n \in D_{k,m}$ and a random element $R \in \mathcal{R}_{n,m}$ (see e.g. [16]). This produces a homomorphic public-key cryptosystem $\mathcal{S}(H)$ over H with respect to the homomorphisms $f_k : G_k \to H$ where $f_k = f_{n,m}$ and $G_k = G_{n,m}$. We also set $A_k = A_{n,m}$ and $P_k = P_{n,m}$.

Theorem 2.1 *Let H be a cyclic group of order $m > 1$. Then under Assumption 1.1 the homomorphic cryptosystem $\mathcal{S}(H)$ is semantically secure against a passive adversary. In addition, the problems* INVERSE$(P_{n,m})$ *and* FACTOR(n, m) *are probabilistic polynomial time equivalent.*

Proof. We recall that the cryptosystem $\mathcal{S}(H)$ is semantically secure iff it is impossible in polynomial in k time to find $h_1, h_2 \in H$ such that a probabilistic polynomial time algorithm can't distinguish for $g \in G_k$ between $f_k(g) = h_1$ and $f_k(g) = h_2$ (see [7]). Thus the first part of the theorem immediately follows from the definition of the problem TEST(P_k) (cf. [8, 7]).

To prove the second part suppose that we are given an algorithm solving the problem FACTOR(n, m). Then one can find the decomposition $n = pq$. Now using Rabin's probabilistic polynomial-time algorithm for finding roots of polynomials over finite prime fields (see [20]), one can solve the problem INVERSE$(P_{n,m})$ for an element $g \in G_{n,m}$ as follows:

> **Step 1.** Find the numbers $g_p \in \mathbb{Z}_p^*$ and $g_q \in \mathbb{Z}_q^*$ such that $g = g_p \times g_q$, i.e. $g_p = g \pmod p$, $g_q = g \pmod q$.
>
> **Step 2.** Apply Rabin's algorithm for the field of order p to the polynomial $x^m - g_p$ and for the field of order q to the polynomial $x^m - g_q$. If at least one of this polynomials has no roots, then output "$P^{-1}(g) = \emptyset$"; otherwise let h_p and h_q be corresponding roots.
>
> **Step 3.** Output "$P_{n,m}^{-1}(g) \neq \emptyset$" and $h = h_p \times h_q$.

We observe that the set $P_{n,m}^{-1}(g)$ is empty, i.e. the g is not an m-power in $G_{n,m}$, iff at least one of the elements g_p and g_q found at Step 1 is not an m-power in $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$ respectively.

This implies the correctness of the output at Step 2. On the other hand, if the procedure terminates at Step 3, then $h^m = h_p^m \times h_q^m = g_p \times g_q = g$, i.e. $h \in P_{n,m}^{-1}(g)$. Thus the problem INVERSE($P_{n,m}$) is reduced to the problem FACTOR(n,m) in probabilistic time $k^{O(1)}$.

Conversely, suppose that we are given an algorithm solving the problem INVERSE($P_{n,m}$). Then the following procedure using well-known observations [7] enables us to find the decomposition $n = pq$.

Step 1. Randomly choose $g \in \mathbb{Z}_n^*$. Set $T = \{g\}$.

Step 2. While $|T| < 3 - (m \pmod 2)$, add to T a random m-root of the element g^m yielded by the algorithm for the problem INVERSE($P_{n,m}$).

Step 3. Choose $h_1, h_2 \in T$ such that $q = \mathrm{GCD}(h_1 - h_2, n) \neq 1$. Output q and $p = n/q$.

To prove the correctness of the procedure we observe that there exists at least 2 (resp. 4) different m-roots of the element g^m for odd m (resp. for even m) where g is the element chosen at Step 1. So the loop at Step 2 and hence the entire procedure terminates with a high probability after a polynomial number of iterations. Moreover, let $T_q = \{h_q : \ h \in T\}$ where h_q is the q-component of h. Then from (3) it follows that $|T_q| = 1$ for odd m, and $|T_q| \leq 2$ for even m. Due to the construction of T at Step 2 this implies that there exist different elements $h_1, h_2 \in T$ such that $(h_1)_q = (h_2)_q$, and consequently

$$h_1 = (h_1)_q = (h_2)_q = h_2 \pmod q.$$

Since $h_1 \neq h_2 \pmod n$, we conclude that $h_1 - h_2$ is a multiple of q and the output at Step 3 is correct.■

We complete the section by mentioning that the decryption algorithm of the homomorphic cryptosystem $\mathcal{S}(H)$ can be slightly modified to avoid applying Rabin's algorithm for finding roots of polynomials over finite fields. Indeed, it is easy to see that an element $g = g_p \times g_q$ of the group $G_{n,m}$ belongs to the subgroup of m-powers iff $g_p^{(p-1)/m} = 1 \pmod p$ and $g_q^{(q-1)/m'} = 1 \pmod q$ where $m' = \mathrm{GCD}(m, q-1)$.

3 Homomorphic cryptosystems using free products

Throughout the section we denote by W_X the set of all the words w in the alphabet X; the length of w is denoted by $|w|$. We use the notation $G = \langle X; \mathcal{R} \rangle$ for a presentation of a group G by the set X of generators and the set $\mathcal{R}$ of relations. Sometimes we omit $\mathcal{R}$

to stress that the group G is generated by the set X. The unity of G is denoted by 1_G and we set $G^{\#} = G \setminus \{1_G\}$.

3.1. Calculations in free products of groups. Let us remind the basic facts on free products of groups (see e.g. [12, Ch. 4]). Let $G_1, \ldots, G_n$ be finite groups, $n \geq 1$. Given a presentation $G_i = \langle X_i; \mathcal{R}_i \rangle$, $1 \leq i \leq n$, one can form a group $G = \langle X;\ \mathcal{R} \rangle$ where $X = \cup_{i=1}^n X_i$ (the disjoint union) and $\mathcal{R} = \cup_{i=1}^n \mathcal{R}_i$. It can be proved that this group does not depend on the choice of presentations $\langle X_i; \mathcal{R}_i \rangle$, $1 \leq i \leq n$. It is called the *free product* of the groups G_i and is denoted by $G = G_1 * \cdots * G_n$; one can see that it does not depend on the order of factors. Without loss of generality we assume below that G_i is a subgroup of G and $X_i = G_i^{\#}$ for all i. In this case $G \subset W_X$ and 1_G equals the empty word of W_X. Moreover, it can be proved that

$$G = \{x_1 \cdots x_l \in W_X :\ x_j \in G_{i_j} \text{ for } 1 \leq j \leq l, \text{ and } i_j \neq i_{j+1} \text{ for } 1 \leq j \leq l-1\}. \qquad (6)$$

Thus each element of G is a word of W_X in which no two adjacent letters belong to the same set among the sets X_i, and any two such different words are different elements of G. To describe the multiplication in G let us first define recursively the mapping $W_X \to G$, $w \mapsto \overline{w}$ as follows

$$\overline{w} = \begin{cases} w, & \text{if } w \in G, \\ \overline{\ldots (x \cdot y) \ldots}, & \text{if } w = \ldots xy \ldots \text{ with } x, y \in X_i \text{ for some } 1 \leq i \leq n, \end{cases} \qquad (7)$$

where $x \cdot y$ is the product of x by y in the group G_i. One can prove that the word $\overline{w}$ is uniquely determined by w and so the mapping is correctly defined. In particular, this implies that given $i \in \overline{n}$ we have

$$\overline{x_1 \cdots x_l} \in G_i \ \Leftrightarrow\ \overline{x_1 \cdots x_l} = \overline{x_{j_1} \cdots x_{j_{l'}}} \qquad (8)$$

where $\{j_1, \ldots, j_{l'}\} = \{1 \leq j \leq l :\ x_j \in G_i\}$ and $j_1 < \cdots < j_{l'}$. Now given $g, h \in G$ the product of g by h in G equals $\overline{gh}$.

Lemma 3.1 *Let $G = G_1 * \cdots * G_n$, $K = K_1 * \cdots * K_n$ be free products of groups and f_i be an epimorphism from G_i onto K_i, $1 \leq i \leq n$. Then the mapping*

$$\varphi : G \to K, \quad x_1 \cdots x_l \mapsto \overline{f_{i_1}(x_1) \cdots f_{i_l}(x_l)} \qquad (9)$$

where $x_j \in G_{i_j}$, $1 \leq j \leq l$, is an epimorphism. Moreover, $\varphi|_{G_i} = f_i$ for all $1 \leq i \leq n$.

Proof. Since $K = \langle Y \rangle$ where $Y = \cup_{i=1}^n K_i^{\#}$, the surjectivity of the mapping φ follows from the surjectivity of the mappings f_i, $1 \leq i \leq n$. Next, let $\varphi_0 : W_X \to W_Y$ be the mapping taking $x_1 \cdots x_l$ to $f_{i_1}(x_1) \cdots f_{i_l}(x_l)$. Then $\varphi(g) = \overline{\varphi_0(g)}$ for all $g \in G$ and

$\varphi_0(ww') = \varphi_0(w)\varphi_0(w')$ for all $w, w' \in W_X$. Since $\overline{\overline{w}\,\overline{w'}} = \overline{ww'}$ for all $w, w' \in W_X$, this implies that

$$\overline{\varphi(g)\varphi(h)} = \overline{\overline{\varphi_0(g)}\,\overline{\varphi_0(h)}} = \overline{\varphi_0(g)\varphi_0(h)} = \overline{\varphi_0(gh)} = \varphi(\overline{gh})$$

for all $g, h \in G$. Thus the mapping φ is a homomorphism. Since obviously $\varphi|_{G_i} = f_i$ for all i, we are done.■

Let H be a finite nontrivial group and K be the free product of cyclic groups generated by all the elements of $H^\#$. Set

$$\mathcal{R}^{(0)} = \{h^{(m_h)} \in W_{H^\#} : h \in H^\#\},$$

$$\mathcal{R}^{(1)} = \{h^{(i)}h' \in W_{H^\#} :\ h, h' \in H^\#,\ 0 < i < m_h,\ h^i \cdot h' = 1_H\},$$

$$\mathcal{R}^{(2)} = \{hh'h'' \in W_{H^\#} :\ h, h', h'' \in H^\#,\ h' \notin \langle h\rangle,\ h \cdot h' \cdot h'' = 1_H\}$$

where $h^{(i)}$ is the word of length $i \geq 1$ with all letters being equal h, m_h is the order of $h \in H$ and $\cdot$ denotes the multiplication in H. Then one can see that

$$K = \langle H^\#; \mathcal{R}^{(0)}\rangle \tag{10}$$

and there is the natural epimorphism $\psi' : K \to H'$ where $H' = \langle H^\#; \mathcal{R}^{(0)} \cup \mathcal{R}^{(1)} \cup \mathcal{R}^{(2)}\rangle$. Since relations belonging to $\mathcal{R}^{(i)}$, $i = 0, 1, 2$, are satisfied in H, we conclude that $\ker(\psi')h_1 \neq \ker(\psi')h_2$ whenever h_1 and h_2 are different elements of H (we identify 1_K and 1_H). On the other hand, it is easy to see that any right coset of K by $\ker(\psi')$ contains a word of length at most 1, i.e. an element of H. Thus $K = \cup_{h\in H} \ker(\psi')h$ and $H \cong H'$, whence the mapping

$$\psi : K \to H, \quad l \mapsto h_l \tag{11}$$

where h_l is the uniquely determined element of H for which $l \in \ker(\psi')h_l$, is an epimorphism with $\ker(\psi) = \ker(\psi')$.

3.2. Main construction of a homomorphic cryptosystem. Let us describe a homomorphic cryptosystem $\mathcal{S}(H)$ over a finite nontrivial group H. If it is a cyclic group of an order $m > 1$, then we define $\mathcal{S}(H)$ to be the homomorphic cryptosystem from Section 2 (see Theorem 2.1). Otherwise we proceed as follows.

Let us fix a natural k (being a security parameter). Let $H^\# = \{h_1, \ldots, h_n\}$ where n is a positive integer (clearly, $n \geq 3$). Set $D_{k,H} = \cup_{i=1}^n D_{k,m_i}$ where m_i is the order of the group $K_i = \langle h_i\rangle$ (see (5)). Given $1 \leq i \leq n$ choose $n_i \in D_{k,m_i}$ and set $\mathcal{S}_i = \mathcal{S}(K_i)$ to be the homomorphic cryptosystem over the cyclic group K_i with respect to the epimorphism $f_i : G_i \to K_i$ (see Theorem 2.1). Without loss of generality we assume that G_i is a subgroup of the group $\mathbb{Z}_{n_i}^*$. Then $f_i = f_{n_i,m_i}$, and we set $A_i = A_{n_i,m_i}$, $P_i = P_{n_i,m_i}$, $R_i = R_{n_i,m_i}$ and

$$G = G_1 * \cdots * G_n, \quad f = \psi \circ \varphi, \tag{12}$$

where the mappings φ and ψ are defined by (9) and (11) respectively, with $K = K_1 * \cdots * K_n$. From Lemma 3.1 and the definition of ψ it follows that the mapping $f : G \to H$ is an epimorphism from G onto H.

To complete the construction we need to define a group $A = A_k$, a homomorphism $P = P_k$ from A to G and randomly choose a transversal of $\ker(f)$ in G. To do this we set

$$X_\varphi = X \cup A_0 \quad X = \cup_{i=1}^n G_i \setminus \ker(f_i), \quad A_0 = \cup_{i=1}^n A_i, \tag{13}$$

all the unions are assumed to be the disjoint ones. Denote by $\to$ the transitive closure of the binary relation $\Rightarrow$ on the set W_{X_φ} defined by

$$v \Rightarrow w \quad \text{iff} \quad w = x^{-1}x_0vx, \qquad v, w \in W_{X_\varphi} \tag{14}$$

where $x \in X \cup \{1_A\}$ and $x_0 \in A_0 \cup \{1_A\}$ with 1_A being the empty word of W_{X_φ}. Thus $v \to w$ if there exist words $v = w_1, w_2, \ldots, w_l = w$ of W_{X_φ} such that $w_i \Rightarrow w_{i+1}$ for $1 \le i \le l-1$. We set

$$A_\varphi = \{a \in W_{X_\varphi} :\ 1_{A_\varphi} \to a\}, \quad P_\varphi : A_\varphi \to G,\ a_1 \cdots a_k \mapsto \overline{P_\varphi(a_1) \cdots P_\varphi(a_k)} \tag{15}$$

where $P_\varphi|_X = \mathrm{id}_X$ and $P_\varphi|_{A_i} = P_i$ for all i. We observe that if $\overline{v} \in \ker(\varphi)$ and $v \Rightarrow w$ for some $v, w \in W_{X_\varphi}$ then obviously $\overline{w} \in \ker(\varphi)$ (see (14)). By induction on the size of a word this implies that $P_\varphi(A_\varphi) \subset \ker(\varphi)$. A straightforward check shows that A_φ is a subgroup of the group $\langle X_\varphi \rangle$. (Indeed, let $v, w \in A_\varphi$. Obviously, $vw \in A_\varphi$ whenever $v \in A_0 \cap \{1_A\}$. Arguing by induction of $|v|$ it suffices to verify that $vw \in A_\varphi$ whenever $v = x^{-1}x_0x$ with $x \in X \cup \{1_A\}$ and $x_0 \in A_0 \cup \{1_A\}$. However, in this case we have $1_A \to w \Rightarrow xwx^{-1} \Rightarrow x^{-1}x_0(xwx^{-1})x = vw$.) In particular, the mapping P_φ is a homomorphism. Similarly, the group A_ψ and the mapping P_ψ defined by

$$A_\psi = \{r \in W_{R_\psi} :\ f(\overline{r}) = 1_H\}, \quad P_\psi : A_\psi \to G,\ a \mapsto \overline{a} \tag{16}$$

where $R_\psi = \cup_{i=1}^n R_i$, are the subgroup of the group $\langle R_\psi \rangle$ and the homomorphism of it to G respectively. Besides, it is easily seen that the restriction of φ to the set $R_\varphi = G \cap W_R$ induces a bijection from this set to the group K. This shows that R_φ is a right transversal of $\ker(\varphi)$ in G. Finally we define the group A and the homomorphism P by

$$A = A_\varphi \times A_\psi, \quad P : A \to G,\ (a, b) \mapsto \overline{P_\varphi(a)P_\psi(b)}. \tag{17}$$

Let R be a right transversal of $\ker(f)$ in G, for instance one can take $R = \{1_G\} \cup \{r'_i\}_{i \in \overline{n}}$ where r'_i is the element of R_i such that $\psi(r'_i) = h_i$, $1 \le i \le n$.

We claim that the homomorphism $P : A \to \ker(f)$ is in fact an epimorphism. Indeed, the set R_φ defined after (16) is a right transversal of $\ker(\varphi)$ in G. So given $g \in \ker(f)$

there exist uniquely determined elements $g_\varphi \in \ker(\varphi)$ and $r_\varphi \in R_\varphi$ such that $g = \overline{g_\varphi r_\varphi}$. Since

$$1_H = f(g) = \psi(\varphi(\overline{g_\varphi r_\varphi})) = \psi(\varphi(r_\varphi)) = f(r_\varphi),$$

we see that $r_\varphi \in A_\psi$ (see (16)). Besides, from statement (i2) of Lemma 3.3 below it follows that there exists $a \in A_\varphi$ for which $P_\varphi(a) = g_\varphi$. Therefore, due to (17) we have

$$P(a, r_\varphi) = \overline{P_\varphi(a)P_\psi(r_\varphi)} = \overline{g_\varphi r_\varphi} = g$$

which proves the claim.

Let us describe the presentations of the groups A, G, K and H. Given $1 \le i \le n$ the elements $a \in A_i$ and $g \in G_i$ being the elements of $\mathbb{Z}^*_{n_i}$ will be represented by the "letters" $]a, i[$ and $[g, i]$ respectively. To multiply two elements $g, h \in G$ one has to find the word $\overline{gh}$ of W_X. It is easy to see that this can be done by means of the recursive procedure (7) in time $((|g| + |h|)k)^{O(1)}$ (here $[x, i] \cdot [y, i] = [xy, i]$ for all $x, y \in \mathbb{Z}^*_{n_i}$ where xy is the product modulo n_i of the numbers x and y, and $n_i \le \exp^{O(k)}$ because $n_i \in D_{k,m_i}$). Since taking the inverse of $g \in G$ can be easily implemented in time $(|g|k)^{O(1)}$, we will estimate further the running time of the algorithms via the number of performed group operations in G and via the sizes of the involved operands. The similar arguments work for the group A. Moreover, relying on (14), (15) and (16) one can randomly generate elements of A. Finally, the group H as well as the groups K_i, $1 \le i \le n$, are given by their multiplication tables, and the group K is given by the presentation (10). So the group operations in K can be performed in time polynomial in the lengths of the input words belonging to $W_{H\#}$. Thus for the data we described the following statements hold:

(H1) *the elements of the group A are represented by words in the alphabet $X_\varphi \cup R_\varphi$; one can get randomly an element of A of size k within probabilistic time $k^{O(1)}$,*

(H2) *the elements of the group G are represented by words in the alphabet X; one can test the equality of elements in G and perform group operations in G (taking the inverse and computing the product) in time $k^{O(1)}$, provided that the sizes of corresponding words are at most k,*

(H3) *the set R, the group H and the bijection $R \to H$ induced by f, are given by the list of elements, the multiplication table and the list of pairs $(r, f(r))$, respectively; $|R| = |H| = O(1)$,*

(H4) *given a word $a \in A$ of the length $|a|$ an element $P(a)$ can be computed within probabilistic time $|a|^{O(1)}$, whereas the problem* INVERSE(P) *can be solved by means of the collection of the secret keys of cryptosystems $\mathcal{S}_i$, $1 \le i \le n$.*

Statement (H4) needs to be explained more precisely. First, the epimorphism P is polynomial time computable because of statement (i1) of Lemma 3.3 and by Lemma 3.5 below the mappings P_φ and P_ψ are polynomial time computable. Second, the problem INVERSE(P) can be efficiently solved by means of using the trapdoor information for the homomorphic cryptosystems $\mathcal{S}_i$, i.e. the factoring of integers $n_i \in D_{k,m_i}$. Indeed, suppose that for each $1 \leq i \leq n$ there is an oracle for the problem INVERSE(P_i). Then given $g_i \in G_i$ one can find the element $f_i(g_i)$ in time $k^{O(1)}$. So given $g \in G$ the element $l = \varphi(g)$ can be found in time $(|g|k)^{O(1)}$ (see (9)). Since $f(g) = \psi(\varphi(g)) = \psi(l)$ and $|l| \leq |g|$, one can find $\psi(l)$ by Lemma 3.5 and then to test whether $g \in \ker(f)$ within the same time. Moreover, due to condition (H3) for cryptosystems $\mathcal{S}_i$ one can efficiently find an element r belonging to the right transversal R_φ of $\ker(\varphi)$ in G such that $\varphi(r) = l$ and $|r| \leq |l|$. Now if $g \in \ker(f)$ then $\psi(l) = 1_H$ and so $r \in A_\psi$. Furthermore,

$$\varphi(gr^{-1}) = \varphi(g)\varphi(r^{-1}) = ll^{-1} = 1_K.$$

Finally, from statement (i3) of Lemma 3.3 it follows that one can find in time $(|g|k)^{O(1)}$ an element $a \in A_\varphi$ such that $P_\varphi(a) = gr^{-1}$. Thus we obtain

$$P(a,r) = \overline{P_\varphi(a)P_\psi(r)} = \overline{gr^{-1}r} = \overline{g} = g,$$

which proves our claim.

We observe that given an element $g \in G$ there exists the uniquely determined element $r \in R$ such that $f(g) = f(r)$ or, equivalently, $f(gr^{-1}) = 1_H$. Since $|R| = O(1)$, this implies that the problem of the computation of the epimorphism f is polynomial time equivalent to the problem of recognizing elements of $\ker(f)$ in G, i. e. in our setting equivalent to the problem TEST(P). The latter together with conditions (H1)-(H4) enable us to define a homomorphic cryptosystem $\mathcal{S}(H)$ over the group H in which the elements of G playing the role of the alphabet of ciphertext messages, all the computations are performed in G and the result is decrypted to H. More precisely:

Encryption: given a plaintext $h \in H$ encrypt as follows: take $r \in R$ such that $f(r) = h$ (invoking (H3)) and a random element $a \in A$ (using (H1)); the ciphertext of h is the element $P(a)r$ of G (computed by means of (H2) and (H4)).

Decryption: given a ciphertext $g \in G$ decrypt as follows: find the elements $r \in R$ and $a \in A$ such that $gr^{-1} = P(a)$ (using (H4)); the plaintext of g is the element $f(r)$ of H (computed by means of (H3)).

Now, the main result of the paper can be formulated as follows.

Theorem 3.2 *Let H be a finite nontrivial group. Then under Assumption 1.1 the homomorphic cryptosystem $\mathcal{S}(H)$ is semantically secure against a passive adversary. In* *addition, given a number k the problem* INVERSE(P_k) *is probabilistic polynomial time equivalent to the family of problems* FACTOR(n, m) *for appropriate $n = \exp(O(k))$ and m ranging over the set of the orders of all the elements of H.*

We complete the subsection by making a remark concerning the construction of the cryptosystem $\mathcal{S}(H)$. In fact, the group K and the epimorphism ψ defined by (10) and (11) can be constructed without using all elements of the group H. To do this it suffices to define K to be the free product of cyclic groups generated by the elements of a set of generators of H. In this case all we need is that any element of H has a short representation in terms of this set of generators and that this representation can be found efficiently.

3.3. Security of $\mathcal{S}(H)$. Proof of Theorem 3.2.

First we observe that if H is a cyclic group, then the required statement follows from Theorem 2.1. Suppose from now on that the group H is not cyclic. Again the first part of the theorem is straightforward (cf. [8, 7]). To prove the second part we consider the following sequence of the homomorphisms:

$$A_\varphi \times A_\psi \xrightarrow{P} G_1 * \cdots * G_n \xrightarrow{\varphi} K_1 * \cdots * K_n \xrightarrow{\psi} H.$$

In the following two lemmas we study the homomorphisms φ and ψ from the algorithmic point of view.

Lemma 3.3 *For the homomorphism P_φ defined in (15) the following statements hold:*

(i1) *given $a \in A_\varphi$ the element $P_\varphi(a)$ can be found in time $|a|^{O(1)}$,*

(i2) $\mathrm{im}(P_\varphi) = \ker(\varphi)$,

(i3) *given an oracle Q_i for the problem* INVERSE(P_i) *for all $1 \le i \le n$, the problem* INVERSE(P_φ) *for $g \in G$ can be solved by means of at most $|g|^2$ calls of oracles Q_i, $1 \le i \le n$,*

(i4) *for each $1 \le i \le n$ the problem* INVERSE(P_i) *is polynomial time reducible to the problem* INVERSE(P_φ).

Proof. Let us prove statement (i1). Let $a = a_1 \cdots a_l$ be an element of A_φ. To find $P_\varphi(a)$ according to (15) we need to compute the words $P_\varphi(a_j)$, $1 \le j \le l$, and then to compute the word $\overline{w}$ where $w = P_\varphi(a_1) \cdots P_\varphi(a_l)$. The first stage can be done in time $|a|^{O(1)}$ because each mapping P_i, $1 \le i \le n$, is polynomial time computable due to Section 2. Since the size of w equals $|a|$, the element $P_\varphi(a)$ can be found within the similar time bound (one should take into account that in the recursive procedure (7) applied to computing $\overline{w}$ from w the length of a current word decreases at each step of the procedure).

To prove statements (i2) and (i3) we note first that the inclusion $\mathrm{im}(P_\varphi) \subset \ker(\varphi)$ was proved after the definition of A_φ and P_φ in (15). The converse inclusion as well as statement (i3) will be proved by means of the following recursive procedure which for a given element $g = x_1 \cdots x_l$ of G with $x_j \in G_{i_j}$ for $1 \le j \le l$, produces a certain pair $(a_g, t_g) \in A_\varphi \times G$. Below we show that this procedure actually solves the problem INVERSE(P_φ).

Step 1. If $g = 1_G$, then output $(1_{A_\varphi}, 1_G)$.

Step 2. If the set $J = \{1 \le j \le l : \ x_j \in \ker(f_{i_j})\}$ is empty, then output $(1_{A_\varphi}, g)$.

Step 3. Set $h = \overline{x_{j+1} \cdots x_l x_1 \cdots x_{j-1}}$ where j is the smallest element of the set J.

Step 4. Recursively find the pair (a_h, t_h). If $t_h \ne 1_G$, then output (a_h, t_h).

Step 5. If $t_h = 1_G$, then output $(a_g, 1_G)$ where $a_g = x_1 \cdots x_{j-1} a_j a_h x_{j-1}^{-1} \cdots x_1^{-1}$ with a_j being an arbitrary element of A_{i_j} such that $P_{i_j}(a_j) = x_j$.■

Since each recursive call at Step 4 is applied to the word $h \in G$ of size at most $|g| - 1$, the number of recursive calls is at most $|g|$. So the total number of oracle Q_i calls, $1 \le i \le n$, at Step 2 does not exceed $|g|^2$. Thus the running time of the algorithm is $(|g|)^{O(1)}$ and statements (i2), (i3) are consequences of the following lemma.

Lemma 3.4 *$g \in \ker(\varphi)$ iff $t_g = 1_G$. Moreover, if $t_g = 1_G$, then $a_g \in A_\varphi$ and $P_\varphi(a_g) = g$.*

Proof. We will prove the both statements by induction on $l = |g|$. If $l = 0$, then the procedure terminates at Step 1 and we are done. Suppose that $l > 0$. If the procedure terminates at Step 2, then $t_g \ne 1_G$. In this case we have $|\varphi(g)| = |g| = l > 0$, whence $g \notin \ker(\varphi)$. Let the procedure terminate at Step 4 or at Step 5. Then $|h| \le |g| - 1$ (see Step 3). So by the induction hypothesis we can assume that $h \in \ker(\varphi)$ iff $t_h = 1_G$. On the other hand, taking into account that $x_j \in \ker(f_{i_j})$ (see the definition of j at Step 3) we get that $h \in \ker(\varphi)$ iff $\overline{u x_j h u^{-1}} \in \ker(\varphi)$ where $u = x_1 \ldots, x_{j-1}$. Since

$$\overline{u x_j h u^{-1}} = \overline{x_1 \cdots x_{j-1} x_j h x_{j-1}^{-1} \cdots x_1^{-1}} = \overline{x_1 \cdots x_l} = \overline{g} = g, \tag{18}$$

this means that $g \in \ker(\varphi)$ iff $h \in \ker(\varphi)$ iff $t_h = 1_G$. This proves the first statement of the lemma because $t_h = t_g$ due to Steps 4 and 5.

To prove the second statement, suppose that $t_g = 1_G$. Then the above argument shows that $h \in \ker(\varphi)$ and so $a_h \in A_\varphi$ and $P_\varphi(a_h) = h$ by the induction hypothesis. This implies that $1_{A_\varphi} \to a_h$. On the other hand, from the definition of a_g at Step 5 it follows that

$a_h \to a_g$ (see (14)). Thus $1_{A_\varphi} \to a_g$, i.e. $a_g \in A_\varphi$ (see (15)). Besides, from the minimality of j it follows that $x_{j'} \in X$ (see (13)) and hence $P_\varphi(x_{j'}) = x_{j'}$ and $P_\varphi(x_{j'}^{-1}) = x_{j'}^{-1}$ for all $1 \le j' \le j-1$ (see (15)). Since $P_\varphi(a_j) = x_j$ and $\overline{h} = h = \overline{x_{j+1} \cdots x_l x_1 \cdots x_{j-1}}$ (see Step 3), we obtain by (18) that

$$P_\varphi(a_g) = \overline{u x_j P_\varphi(a_h) u^{-1}} = \overline{u x_j h u^{-1}} = g$$

which completes the proof of the Lemma 3.4.∎

To prove statement (i4) let $1 \le i \le n$ and $g \in G_i$. Then since obviously $g \in \ker(f_i)$ iff $g \in \ker(\varphi)$, one can test whether $g \in \ker(f_i)$ by means of an algorithm solving the problem INVERSE(P_φ). Moreover, if $g \in \ker(f_i)$, then this algorithm yields an element $a \in A_\varphi$ such that $P_\varphi(a) = g$. Then assuming $a = a_1 \cdots a_l$ with $a_j \in X_\varphi$, the set $J_a = \{1 \le j \le l : \ a_j =]a_j^*, i[\}$ can be found in time $O(|a|)$ (we recall that due to our presentation any element a_j is of the form either $]a_j^*, i_j[$ or $[a_j^*, i_j]$ where $1 \le i_j \le n$ and $a_j^* \in \mathbb{Z}_{n_{i_j}}^*$, and $P_{i_j}(a_j) \in \ker(f_{i_j})$ iff $a_j \in A_0$ iff $a_j =]a_j^*, i_j[$). Now the element

$$a^* =]\prod_{j \in J_a} a_j^*, i[$$

obviously belongs to the set $A_i \subset A_0$. On the other hand, since $g \in G_i$, we get by (8) that

$$g = \overline{P_\varphi(a_1) \cdots P_\varphi(a_l)} = \overline{\prod_{j \in J} P_\varphi(a_j)} \tag{19}$$

where $J = \{1 \le j \le k : \ P_\varphi(a_j) \in G_i\}$. Taking into account that G_i is an Abelian group and the mapping $P_i : A_i \to G_i$ is a homomorphism, we have

$$\overline{\prod_{j \in J} P_\varphi(a_j)} = \overline{\prod_{j \in J_a} P_i(a_j) \prod_{j \in J \setminus J_a} P_\varphi(a_j)} = \overline{P_i(a^*) \prod_{j \in J \setminus J_a} P_\varphi(a_j)}. \tag{20}$$

Moreover, since $1_{A_\varphi} \to a$, from (14) it follows that there exists involution $j \to j'$ on the set $J \setminus J_a$ such that $a_j = [a_j^*, i]$ iff $a_{j'} = [(a_j^*)^{-1}, i]$ (we recall that $a_j =]a_j^*, i[$ for $j \in J_a$ and $a_j = [a_j^*, i]$ for $j \in J \setminus J_a$). This implies that $\prod_{j \in J \setminus J_a} P_\varphi(a_j) = 1_G$. Thus from (19) and (20) we conclude that:

$$g = \overline{P_i(a^*)} = \overline{P_\varphi(a^*)} = P_\varphi(a^*).$$

This shows that the element $a^* \in A_i$ with $P_\varphi(a^*) = g$ can be constructed from a in time $O(|a|)$. Generating random elements of the groups A_i, one can efficiently transform the element a^* to a random element $\widetilde{a}$ so that $P_\varphi(\widetilde{a}) = P_\varphi(a^*) = g$. Thus the problem INVERSE(P_i) is polynomial time reducible to the problem INVERSE(P_φ). The Lemma 3.3 is proved.∎

Lemma 3.5 *Let K be the group given by presentation (10) and the epimorphism ψ is defined by (11). Then given $v \in K$ one can find the element $\psi(v)$ in time $(|v||H|)^{O(1)}$.*

Proof. It is easy to see that the group K can be identified with the subset of the set $W_{H^\#}$ so that $w \in K$ iff the length of any subword of w of the form $h \cdots h$ (i.e. the repetition of a letter h) is at most $m_h - 1$. Having this in mind we claim that the following recursive procedure computes $\psi(v)$ for all $v = x_1 \cdots x_t \in K$.

Step 1. If $t \leq 1$, then output $\psi(v) = v$.

Step 2. Choose $h \in H$ such that $x_1 x_2 h \in \mathcal{R}^{(1)} \cup \mathcal{R}^{(2)}$.

Step 3. Output $\psi(v) = \psi(h^{-1} x_3 \cdots x_t)$.

The correctness of the procedure follows from the definitions of sets $\mathcal{R}^{(1)}$, $\mathcal{R}^{(2)}$, and the fact that recursion at Step 3 is always applied to a word the length of which is smaller than the length of the current word. In fact, the above procedure produces the representation of v in the form $v = w_1 \cdots w_{t-1} \psi(v)$ where $w_j \in \mathcal{R}^{(1)} \cup \mathcal{R}^{(2)}$ for all $1 \leq j \leq t-1$ and $\psi(v) \in H$. Since obviously $w_1 \cdots w_{t-1} \in \ker(\psi)$, we conclude that $\psi(v) = h_v$ (see (11)). To complete the proof it suffices to note that the running time of the above procedure is $O(|v|(|\mathcal{R}^{(1)}| + |\mathcal{R}^{(2)}|))$.■

Finally, let us complete the proof of Theorem 3.2. We have to show only that for any $1 \leq i \leq n$ the problem INVERSE(P_i) (to which the factoring of integers n_i is reduced) is polynomial time reducible to the problem INVERSE(P). To do this let $g \in G$. If $g \notin \ker(f)$, then obviously $g \notin \ker(\varphi)$. Now let $g \in \ker(f)$ and $(a, b) \in A$ be such that $P_\varphi(a) P_\psi(b) = g$. Since $P_\psi(b)$ belongs to the right transversal R_φ of $\ker(\varphi)$ in G, it follows that $g \in \ker(\varphi)$ iff $P_\psi(b) = 1_G$. Moreover, if $P_\psi(b) = 1_G$, then obviously $P_\varphi(a) = g$. Taking into account that the element $P_\psi(b)$ can be found in time $|b|^{O(1)}$ (see (16)), we conclude that the problem INVERSE(P_φ) is polynomial time reducible to the problem INVERSE(P). Thus our claim follows from statement (i4) of Lemma 3.3. Theorem 3.2 is proved.■

4 Encrypted simulating of boolean circuits

Let $B = B(X_1, \ldots, X_n)$ be a boolean circuit and H be a group. Following [1] we say that a word

$$h_1^{X_{l_1}} \cdots h_m^{X_{l_m}}, \quad h_1, \ldots, h_m \in H, \quad l_1, \ldots, l_m \in \{1, \ldots, n\}, \tag{21}$$

is a *simulation* of size m of B in H if there exists a certain element $h \in H^\# = H \setminus \{1\}$ such that the equality

$$h_1^{x_{l_1}} \cdots h_m^{x_{l_m}} = h^{B(x_1, \ldots, x_n)}$$

holds for any boolean vector $(x_1, \dots, x_n) \in \{0,1\}^n$. It is proved in [1] that given an arbitrary *unsolvable* group H and a boolean circuit B there exists a simulation of B in H, the size of this simulation is exponential in the depth of B (in particular, when the depth of B is logarithmic $O(\log n)$, then the size of the simulation is $n^{O(1)}$).

We say that for the circuit B we have an *encrypted simulation* over a homomorphic cryptosystem with respect to epimorphisms $f_k : G_k \to H$ if for each k there exist $g_1, \dots, g_m \in G_k$, and a certain element $h \in H^{\#}$ (depending on k) such that

$$f_k({g_1}^{x_{l_1}} \cdots {g_m}^{x_{l_m}}) = h^{B(x_1,\dots,x_n)} \tag{22}$$

for any boolean vector $(x_1, \dots, x_n) \in \{0,1\}^n$. Thus having a simulation (21) of the circuit B in H one can produce an encrypted simulation of B by choosing randomly $g_i \in G_k$ such that $f_k(g_i) = h_i$, $1 \le i \le m$ (in this case, equality (22) is obvious). Now combining the homomorphic cryptosystem of Section 3 with the above mentioned result from [1] we get the following statement.

Corollary 4.1 *For an arbitrary finite unsolvable group H, a homomorphic cryptosystem $\mathcal{S}$ over H, the security parameter k and any boolean circuit of the logarithmic depth $O(\log k)$ one can design in time $k^{O(1)}$ an encrypted simulation of this circuit over $\mathcal{S}$.* ∎

The meaning of an encrypted simulation is that given (publically) the elements $g_1, \dots, g_m \in G_k$ and $h \in H^{\#}$ from (22) it should be supposedly difficult to evaluate $B(x_1, \dots, x_n)$ since for this purpose one has to verify whether an element $g_1^{x_{l_1}} \cdots g_m^{x_{l_m}}$ belongs to $\ker(f_k)$. On the other hand, the latter can be performed using the trapdoor information. In conclusion let us mention the following two known protocols of interaction (cf. e.g. [2, 24, 21, 22]) based on encrypted simulations.

The first protocol is called *evaluating an encrypted circuit.* Assume that Alice knows a trapdoor in a homomorphic cryptosystem over a group H with respect to epimorphisms $f_k : G_k \to H$ and possesses a boolean circuit B which she prefers to keep secret, and Bob wants to evaluate $B(x)$ at an input $x = (x_1, \dots, x_n)$ (without knowing B and without disclosing x). To accomplish this Alice transmits to Bob an encrypted simulation (22) of B, then Bob calculates the element $g = g_1^{x_{l_1}} \cdots g_m^{x_{l_m}}$ and sends it back to Alice, who computes and communicates the value $f_k(g)$ to Bob. If the depth of the boolean circuit B is $O(\log k)$ and the homomorphic cryptosystem is as in Subsection 3.2, then due to Corollary 4.1 the protocol can be realized in time $k^{O(1)}$ (here we make use of that the size of a product of two elements in G_k does not exceed the sum of their sizes).

In a different setting one could consider in a similar way evaluating an encrypted circuit $B_H(y_1, \dots, y_n)$ over a group H (rather than a boolean one), being a sequence of group operations in H with inputs $y_1, \dots, y_n \in H$. The second (dual) protocol is called *evaluating at an encrypted input.* Now Alice has an input $y = (y_1, \dots, y_n)$ (desiring to conceal it) which she encrypts randomly by the tuple $z = (z_1, \dots, z_n)$ belonging to G_k^n

such that $f_k(z_i) = y_i$, $1 \le i \le n$, and transmits z to Bob. In his turn, Bob who knows a circuit B_H (which he wants to keep secret) yields its "lifting" $f_k^{-1}(B_H)$ to G_k by means of replacing every constant $h \in H$ occurring in B_H by a random $g \in G_k$ such that $f_k(g) = h$ and replacing the group operations in H by the group operations in G_k, respectively. Then Bob evaluates the element $(f_k^{-1}(B_H))(z) \in G_k$ and sends it back to Alice, finally Alice applies f_k and obtains $f_k((f_k^{-1}(B_H))(z)) = B_H(y)$ (even without revealing it to Bob). Again if the depth of the circuit B_H is $O(\log k)$ and the homomorphic cryptosystem is as in Subsection 3.2, then the protocol can be realized in time $k^{O(1)}$. Note that the protocol of evaluating at an encrypted input for a boolean circuit was also accomplished in [24] in a way different from the above (in [24] Alice encrypts bits by means of pertinent boolean vectors). However, the approach of [24] unlike our construction is not applicable directly to the protocol of evaluating an encrypted circuit.

It would be interesting to design homomorphic cryptosystems over *rings* rather than groups (see [10]).

Acknowledgements. The authors would like to thank the Max-Planck Institut fuer Mathematik (Bonn) during the stay in which this paper was initiated; also Eberhard Becker and Igor Shparlinski for useful discussions. The research of the second author was supported by the grant of NATO.

References

[1] D. M. Barrington, H. Straubing, D. Therien, *Non-uniform automata over groups*, Information and Computation, **132** (1990), 89–109.

[2] J. Benaloh, *Dense probabilistic encryption*, First Ann. Workshop on Selected Areas in Cryptology, 1994, 120–128.

[3] D. Coppersmith, I. Shparlinski, *On polynomial approximation of the discrete logarithm and the Diffie-Hellman mapping*, J. Cryptology, **13** (2000), 339–360.

[4] H. Davenport, *Multiplicative number theory*, Springer, 1980.

[5] Do Long Van, A. Jeyanthi, R. Siromony, K. Subramanian, *Public key cryptosystems based on word problems*, in ICOMIDC Symp. Math. of Computations, Ho Chi Minh City, April, 1988.

[6] 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.

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

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

[9] D. Grigoriev, *Public-key cryptography and invariant theory*, `arXiv:math.cs.-CR/0207080`.

[10] D. Grigoriev, I. Ponomarenko, *Homomorphic public-key cryptosystems over groups and rings*, to appear in Quaderni di Matematica, 2004.

[11] 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.

[12] 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.

[13] K. Koyama, U. Maurer, T. Okamoto, S. Vanstone, *New public-key schemes based on elliptic curves over the ring* $\mathbb{Z}_n$, Lecture Notes in Computer Science, **576** (1991), 252–266.

[14] U. Maurer, S. Wolf, *Lower bounds on generic algorithms in groups*, Lecture Notes in Computer Science, **1403** (1998), 72–84.

[15] A. Menezes, P. van Oorshot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press (1997), p.307.

[16] 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.

[17] T. Okamoto, S. Uchiyama, *A New Public-Key Cryptosystem as Secure as Factoring*, Lecture Notes in Computer Science, **1403** (1998), 308–317.

[18] S.-H. Paeng, D. Kwon, K.-C. Ha, J. H. Kim, *Improved public key cryptosystem using finite non-abelian groups*, Preprint NSRI, Korea.

[19] P. Paillier, *Public-Key Cryptosystem Based on Composite Degree Residuosity Classes*, Lecture Notes in Computer Science, **1592** (1999), 223-238.

[20] M. O. Rabin, *Probabilistic algorithms in finite fields*, SIAM J. Comput., **9** (1980), 273–280.

[21] D. K. Rappe, *Algebraisch homomorphe kryptosysteme*, Diplomarbeit, Dem Fachbereich Mathematik der Universität Dortmund, Oktober 2000, http://www.matha.-mathematik.uni-dortmund.de/˜rappe/.

[22] R. L. Rivest, L. Adleman, M. Dertouzos, *On Data Banks and Privacy Homomorphisms*, Foundation of Secure Computation, Academic Press, 1978, 169–177.

[23] R. Solovay, V. Strassen, *A fast Monte-Carlo test for primality*, SIAM J. Comput., **6** (1977), 84–85.

[24] T. Sander, A. Young, M. Yung, *Non-interactive cryptocomputing for* NC^1, Proc. 40th IEEE Symp. Found. Comput. Sci, 1999, 554–566.

[25] A. Yao, *How to generate and exchange secrets*, Proc. 27th IEEE Symp. Found. Comput. Sci, 1986, 162–167.