

ALGEBRAIC CRYPTOGRAPHY: NEW CONSTRUCTIONS AND THEIR SECURITY AGAINST PROVABLE BREAK

D. GRIGORIEV, A. KOJEVNIKOV, AND S. J. NIKOLENKO

ABSTRACT. Very few known cryptographic primitives are based on noncommutative algebra. Each new scheme is of substantial interest, because noncommutative constructions are secure against many standard cryptographic attacks. On the other hand, cryptography does not provide security proofs that might allow the security of a cryptographic primitive to rely upon structural complexity assumptions. Thus, it is important to investigate weaker notions of security.

In this paper, new constructions of cryptographic primitives based on group invariants are proposed, together with new ways to strengthen them for practical use. Also, the notion of a *provable break* is introduced, which is a weaker version of the regular cryptographic break. In this new version, an adversary should have a proof that he has correctly decyphered the message. It is proved that the cryptosystems based on matrix group invariants and a version of the Anshel–Anshel–Goldfeld key agreement protocol for modular groups are secure against provable break unless $NP = RP$.

§1. ALGEBRAIC CRYPTOGRAPHY

Public-key cryptography, since its very beginning [16, 53], has been actively employing algebraic constructions. For example, the RSA protocol is based on number theory; the very construction of the protocol requires computing the Euler totient $\varphi(n)$. Its security is based on factoring a number into prime divisors, or, more precisely, on the hardness of the so-called “RSA problem”: find roots of a given degree modulo a number $n = pq$, where p and q are prime (this task may not be equivalent to factoring; see [14, 54, 15] for more information).

However, the term *algebraic cryptography* is usually employed in a narrower meaning. Algebraic cryptography deals with constructions where encoding and decoding are both group homomorphisms. In [29], Grigoriev and Ponomarenko gave the following definition of a *homomorphic cryptosystem* (compare with Definition 2, where we introduce the general notion of a cryptosystem).

Definition 1. Let H be a finite nonidentity group, G a finitely generated group, and $f : G \rightarrow H$ an epimorphism. Assume that R is a set of distinct representatives of the right cosets of $\ker(f)$ in G , A is a set of words in some alphabet, and a mapping $P : A \rightarrow G$ satisfies $\text{Im}(P) = \ker(f)$. A triple $S = (R, A, P)$ is called a *homomorphic cryptosystem* over H with respect to f if the following conditions are satisfied:

- random elements (of the sets A , G , H) can be generated, and the inverse of an element and the product of two elements (in the group G or H) can be computed

2000 *Mathematics Subject Classification.* Primary 94A60, 68P25, 11T71.

Key words and phrases. Algebraic cryptography, cryptographic primitives, provable break.

The research was done during the stay at the Max-Planck-Institut für Mathematik, Bonn, Germany.

The second and third authors were supported in part by INTAS (YSF fellowship no. 05-109-5565) and by RFBR (grant nos. 05-01-00932 and 06-01-00502).

in probabilistic polynomial in N time, where N is the size of the presentations of G , H , and A ;

- $|R| = |H|$, and the image $f(g)$ of every element $g \in R$ as well as a unique preimage $g \in R$ such that $f(g) = h$ of every element $h \in H$ can be computed in probabilistic polynomial (in N) time;
- the mapping P is a trapdoor function.

Formally speaking, even in this stricter sense, algebraic cryptography was introduced almost at the same time as public-key cryptography: the quadratic residue cryptosystem was the first homomorphic cryptosystem and one of the first known cryptosystems in general [23, 22].

However, algebraic cryptography in its modern sense stems from the works on the elliptic curve cryptography [45, 37]. The basic constructions of elliptic curve primitives differ very little from the constructions based on the discrete logarithm, such as the Diffie–Hellman key agreement protocol. The main difference is that the elliptic curve cryptography computes in an Abelian group of points of an elliptic curve $y^2 = x^3 + ax + b$ or $y^2 + xy = x^3 + ax^2 + b$; this helps to reduce the key size substantially and make the cryptosystems more efficient [10, 58, 32]. Lately, the U.S. government introduced a number of cryptographic standards based on elliptic curve cryptography.

Note that both classical constructions and elliptic curve cryptography deal with *Abelian* groups. This additional structure allows encoding and decoding algorithms to get even more efficient, while the analysis of arising computational tasks simplifies. However, over the last ten years, Abelian constructions were found to be susceptible to a new kind of cryptographic attack, *quantum computing*. Starting with the seminal work of Peter Shor [55, 9], factoring and the discrete logarithm were solved in a simple and efficient manner on a quantum computer for Abelian groups [24] (see also [42, 48] and Chapter 20 of [7]). In fact, a quantum computer is able to efficiently solve the problem of calculating the order of an element in an Abelian group (via quantum Fourier transforms), and this problem generalizes the discrete logarithm.

The discrete logarithm and factoring were effectively the basis of all classical cryptography, including elliptic curve cryptography. Thus, in a situation when quantum computing is already gradually becoming feasible (although the prospects are not yet clear), cryptography can no longer limit itself to commutative constructions. Cryptographers aim to create *non-Abelian* constructions, which are expected to be more secure [5].

That is why elliptic constructions and the classical Diffie–Hellman and RSA constructions were generalized, and the task of constructing homomorphic cryptosystems was introduced in [19, 60]. The first steps in using algebraic constructions, in particular group theory, for building cryptographic primitives were made in [8, 47, 49, 50]. An important step on the road of non-Abelian cryptographic constructions was the papers of Anshel, Anshel, and Goldfeld [3, 4, 2]. We shall focus on one of their constructions in §9.

These ideas were further developed in the work of Grigoriev and Ponomarenko. They developed the basic definitions, such as Definition 1, and built constructions of non-Abelian homomorphic public-key cryptosystems [29], homomorphic public-key cryptosystems over rings [28], and a general scheme of building complex homomorphic cryptosystems from smaller “blocks” [31] (this scheme will play an important part in the present paper). They also investigated the relationship between homomorphic cryptosystems and encoding Boolean circuits [30].

In [25], Grigoriev suggested a method for using group invariants for public-key cryptography. In §5, we shall discuss this construction in detail, but now we proceed to

provable break — the second basic idea of the present work. This paper is a revised and extended version of [27].

§2. WEAK RESULTS IN MODERN CRYPTOGRAPHY

Virtually, modern cryptography does not allow one to prove the security of public key primitives. Starting with the very first Diffie–Hellman key agreement protocol [16] and the RSA public key cryptosystem [53], numerous cryptographic constructions have been devised, but *not a single* proof of their security has appeared so far. Indeed, it would be very hard to find an unconditional proof, because it would necessarily include the proof of $P \neq NP$. However, conditional proofs of the kind “if $P \neq NP$, then a protocol is secure” are also hard to get. The matter here is that the classical notion of cryptographic security is related naturally to average-case complexity rather than classical worst-case complexity.

We begin with a classical notion of *semantic security* of a cryptosystem; to define it, we need to begin with the definition of a cryptosystem [22].

Definition 2. A *public-key encryption scheme* S consists of three probabilistic worst-case polynomial-time algorithms (G, E, D) for key generation, encryption, and decryption, respectively.

The key generation algorithm G on input 1^n (n is the security parameter) produces a pair $G(1^n) = (e, d)$ of public and private keys. The encryption algorithm E takes as input a public key e and a plaintext message m and produces a ciphertext

$$E(e, m) = c.$$

Finally, the decryption algorithm D takes as input a private key d and a ciphertext c . The output of D is a message

$$D(d, c) = m',$$

which may fail to equal the original message m when $E(e, m) = E(e, m')$. These situations are called *collisions*; we assume that collisions happen with negligible probability.

Definition 3. An encryption scheme (G, E, D) is *semantically secure* if, for all probabilistic polynomial-time algorithms M and A , functions h , and polynomials Q , there exists a probabilistic polynomial-time algorithm B such that, for sufficiently large k ,

$$\begin{aligned} \Pr_r [A(1^k, c, e) = h(m) \mid (e, d) \leftarrow^r G(1^k), m \leftarrow^r M(1^k), c \leftarrow^r E(e, m)] \\ \leq \Pr_r [B(1^k) = h(m) \mid m \rightarrow^r M(1^k)] + \frac{1}{Q(k)}. \end{aligned}$$

Informally speaking, every adversary who knows the distribution of the messages M and receives as input the encoded message and the public key will not be able to decode it substantially more often than an algorithm that does not know anything except M (M is necessary because should, for example, only one message be transmitted all the time, then the adversary would indeed be able to decode it, even without the code and the public key). Here the main problem is that the probabilities in this definition are taken over the distribution on the inputs of the cryptosystem, over the messages; of course, this is natural because in practice it usually suffices to break a cryptosystem on a substantial fraction of the inputs, not necessarily on all of them.

Recent results relate cryptographic security (in the sense of Definition 3) to worst-case complexity assumptions for certain problems [1, 17, 51, 52]. However, so far these assumptions appear artificial and, as before, do not translate into assumptions about (in)equality of some complexity classes. In general, it seems that modern cryptography still has a very long way to go before any *provably* secure constructions. There even

exists evidence supporting the view that such constructions may not exist or may lead to improbable consequences for complexity classes [13].

Thus, it is natural that researchers, having encountered too hard a problem, began working on alternative definitions, criteria, and contexts, trying to prove security of some constructions in some definitions.

The first natural approach would be to consider not the “true” cryptographic primitives but their weaker counterparts, such as *one-way functions* (note that a one-way function does not imply a public-key cryptosystem; one would need a trapdoor function for it). Indeed, results on one-way functions are easier to prove. For example, complete one-way functions (by a *complete* one-way function we mean a function f such that if one-way functions exist, then f is a one-way function) have already been known for quite a long time [40, 21], while complete public-key cryptosystems appeared only recently in the papers [26, 33] by Harnik et al. and by Grigoriev et al. Moreover, recently Levin developed more natural combinatorial constructions of complete one-way functions [41]; this theme was continued by recent results by Kojevnikov and Nikolenko [38].

Another approach that also leads to weaker results is to change the notion of security itself. If we consider weaker notions of security, we are able to construct provably secure primitives. For example, consider the theory of *feebly one-way functions*, developed by A. Hiltgen [34, 35]. Hiltgen considered the circuit complexity of functions in the complete binary basis, which is the hardest case for proving lower bounds [12, 59]. He managed to devise a linear invertible function such that computing this function is almost twice as easy as inverting it. These bounds, however weak, were the first unconditional security proofs for one-way functions. Lately, Hirsch and Nikolenko created a feebly secure trapdoor function [36]; it appears possible to devise a provably secure cryptosystem along the same lines (security is understood in an extremely restricted sense, of course).

On the other hand, partial results were obtained under the assumption of a very strong adversary, a *worst-case adversary* who breaks the code *in all cases*. No wonder that in this setting one could base security on worst-case complexity assumptions [18, 39]. For a detailed survey on the subject, we refer the reader to the book [46] and to earlier papers of the first author [30, 31].

In the next section, we introduce and investigate another approach to weakening the notion of security. It turns out that our definition of *provable break*, while being somewhat artificial, enables us to associate the provable break of invariant-based cryptosystems and Anshel–Anshel–Goldfeld key agreement protocols with worst-case structural complexity assumptions.

§3. PROVABLE BREAK

Consider a system with three participants: Alice, Bob, and Charlie. Suppose that, as usual, Alice (A) and Bob (B) are engaged in a cryptographic protocol (in a key agreement protocol Alice and Bob are equal peers, and in a public key cryptosystem, Alice generates a pair of public and private keys and emits the public key, while Bob encodes his message and sends it to Alice over an open channel), and Charlie tries to eavesdrop, decoding the messages that Bob sends to Alice. But now Charlie’s task is different: not only does he need to decipher Bob’s message, he also wants to be able to *prove* that his decoded message is actually what Bob had in mind. Perhaps, Charlie does not really trust the results he receives; perhaps, he has a boss who does not trust Charlie’s algorithm of breaking the protocol. This is (informally) what we call a *provable break*.

In this setting, it is not sufficient for Charlie merely to recover the encrypted message m from a ciphertext c , he should also justify that it is possible to encode m into c . What

could serve as such a justification? We take the following natural idea as the definition: in the provable break security model, an adversary, given a codeword $E(m)$, should not only produce the message m , but also present suitable random bits of E that might lead to such a cipher.

Remark 1. In what follows, we (equivalently) redefine the encoding algorithm E as a deterministic worst-case polynomial-time algorithm with access to a random string r . It receives as input the public key e , the message m , and a random bit string r and outputs the encoded message $E(r, e, m) = c$.

There may be several sets of random bits $\{r_1, \dots, r_k\}$ that produce the same cipher: $E(m, pk, r_1) = \dots = E(m, pk, r_k)$. In this case, of course, an adversary only needs to present *some* random string that results in the cipher, not necessarily the one Bob actually used (when $k > 1$, Charlie has absolutely no chance to find it anyway).

Informal discussions of the provable break began in connection with the Rabin–Goldwasser–Micali cryptosystem based on quadratic residues [22]. It was shown that a provable break of this cryptosystem implies that factoring is contained in RP. However, we know of no reference where a formal definition was presented and studied.

As we have already mentioned in §2, one of the most fundamental unsolved questions in theoretical cryptography is to construct a secure encryption scheme based on some natural complexity assumptions such as $P \neq NP$. In this paper, we present two slightly different definitions of a provable break (one weaker than the other) and prove that two different cryptographic protocols, namely, the Anshel–Anshel–Goldfeld key agreement protocol and cryptosystems based on group invariants, are all secure against a provable worst-case break provided $NP \not\subseteq RP$. For the latter cryptosystem, we develop new ways to provide for their security in the usual cryptographic sense.

§4. DEFINITIONS

First, we define the provable break of public-key cryptosystems and then extend it to key agreement protocols. We present two separate definitions, one of them for the average case, and the other for the worst case. For this, we recollect Definition 2 and define the corresponding provable break.

Definition 4. An adversary C performs a *provable break* of a cryptosystem (G, E, D) if for a uniform distribution over messages m and random bits of all participating algorithms (the public key pk is taken from the pair (pk, sk) generated by the key generation algorithm $G(1^n)$) we have

$$\Pr [C(E(m, pk, r), pk) = (m, r')] \geq \frac{1}{\text{poly}(n)},$$

where $E(m, pk, r') = E(m, pk, r)$, and n is the security parameter.

The security parameter in Definition 4 is effectively the key length; the security parameter should be chosen so that the known adversary algorithms would run long enough to make a break hopeless.

If E is deterministic, then a provable break is equivalent to a usual break (the set of random bits is empty). An adversary can prove that he has correctly decoded a message by encoding it again. This is precisely the idea of a provable break: an adversary should not only decipher the message, but also check that the cipher is actually a valid one; while the former task may be trivial (as it will be in some of our examples), the latter may be very hard.

We also introduce the notion of a very strong adversary who breaks the cryptosystem in the worst case. The difference from the usual break is that the adversary should be successful on *all* inputs.

Definition 5. An adversary C performs a *provable worst-case break* of a cryptosystem (G, E, D) if for all messages m , all pairs of keys (pk, sk) generated by $G(1^n)$, and all random bits of the encoding algorithm E we have

$$\Pr [C(E(m, pk, r), pk) = (m, r')] \geq \frac{1}{\text{poly}(n)},$$

where $E(m, pk, r') = E(m, pk, r)$, n is the security parameter, and the distribution is taken over the random bits of the adversary C .

We say that a cryptosystem (G, E, D) is *secure against a provable (worst-case) break* if there is no polynomial probabilistic Turing machine C performing a provable (worst-case) break of (G, E, D) .

Remark 2. It is easy to think up a trivial cryptosystem that is secure against the provable break (in the sense of Definition 4, which automatically makes it secure against a worst-case provable break in the sense of Definition 5). Let Bob transfer the message openly (decryption is thus trivial), but add a value of some one-way function at the end of the message. Alice may disregard this one-way function, but Charlie would have to invert this one-way function in order to get a valid set of Bob's random bits. Therefore, our task is not to simply devise cryptosystems that are secure against a provable break, but to devise them in such a way that they are or at least may be made secure in the usual cryptographic sense. Of course, we cannot *prove* their security, but we provide constructions that we believe to produce reasonably secure cryptosystems.

§5. INVARIANT-BASED CRYPTOSYSTEMS AND THEIR PROVABLE BREAK

5.1. Cryptosystems based on group invariants. In [25], D. Grigoriev suggested a new class of public-key cryptosystems based on group invariants. In an invariant-based cryptosystem, Alice chooses a group $G \leq GL(n, F)$ acting on some vector space F^n . As a private key, Alice chooses a set X and an invariant $f : F^n \rightarrow X$ such that $f(gx) = f(x)$ for all $g \in G$. She also selects a set (or a space given by generators) of messages $M \subseteq F^n$ such that $f(m_1) \neq f(m_2)$ for all $m_1 \neq m_2 \in M$. Thus, an invariant-based cryptosystem is determined by a triple (G, f, M) . As a public key, Alice transmits generators of G and M .

Bob selects a vector $m \in M$ (m is Bob's message) and a random element $g \in G$. Then, Bob communicates gm to Alice. Alice can decipher the message by taking the invariant $f(gm) = f(m)$. Since she had chosen the set of messages M as a transversal set for the orbits, the value of $f(m)$ allows her to uniquely determine the original message m . We say that a triple (G, f, M) is *admissible* if it correctly determines an invariant-based cryptosystem.

It is now clear that the primary concern of the security of invariant-based cryptosystems is to find a well-concealed invariant. In what follows, we give several ways to do so. These ways are similar to those employed in [31] and may be summarized with the following construction. Consider a tree such that each node of it contains a triple (G, f, M) . Alice builds this tree from the leaves to the root, keeping track of G , f , and M at each step. After the tree is created, Alice takes the cryptosystem from the root and uses it.

An adversary will thus be able to break the cryptosystem if he knows the structure of the tree (we suppose that the cryptosystems in the leaves are easy to break, otherwise there is no point in growing a tree at all). From the security point of view this structure is equivalent to the description of the invariant and may also be regarded as Alice's private key. The security of this cryptosystem will rely on the difficulty of the conjugacy and membership problems, as in [31] (see §6 for the details).

5.2. An invariant-based cryptosystem secure against a provable break unless $\text{NP} \subseteq \text{RP}$. The construction is based on the modular group. The *modular group* is the multiplicative group $SL_2(\mathbb{Z})$ of (2×2) -matrices of determinant 1 (*unimodular matrices*). Algebraic properties of this group are described in detail, e.g., in [6].

In [11, Corollary 11.5], Blass and Gurevich proved that the following *bounded membership problem* (BM) for the modular group is NP-complete.

Problem 1. Let X be a unimodular matrix, S a finite set of unimodular matrices, and N a positive integer. Can X be represented as $\prod_{i=1}^m Y_i$, where $m \leq N$ and for each i either Y_i or Y_i^{-1} is in S ?

Remark 3. Do not confuse this problem with other problems for which in [11] it was proved that they are DistNP-complete (complete with respect to average case reductions). The primary difference is that in this case we are dealing with *group* membership, while RNP-complete problems arise from checking membership in *semigroups*.

Let us take G to be the unimodular group

$$G = \left\{ \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix}, x \in \mathbb{Z} \right\}.$$

As the invariant, we take a rather trivial map

$$f \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_2,$$

and as the message space, the following space of vectors:

$$M = \left\{ \begin{pmatrix} 1 \\ x \end{pmatrix}, x \in \mathbb{Z} \right\}.$$

Bob selects a random element g in the given group (obtained by multiplying not more than, say, N generators), transports the message vector m into gm and transmits gm and N . Alice computes $f(gm)$ and decides which m it was.

Note that this “cryptosystem” is trivial to break: encryption does not change the part of the vector that actually carries the message. However, we shall presently see that its provable break is NP-hard.

Theorem 1. *If there is a polynomial adversary C performing a provable worst-case break of the invariant-based cryptosystem described above, then $\text{NP} \subseteq \text{RP}$.*

Proof. In short, the provable break is NP-hard because the Integer Sum problem is easily reduced to deciding bounded membership in a subgroup of the modular group, as was shown in [11].

First, note that

$$\begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & \mu \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & \lambda + \mu \\ 0 & 1 \end{pmatrix}.$$

Thus, the problem of deciding bounded membership in a subgroup of the modular group is equivalent to the problem of deciding whether a given number is expressible as a bounded sum of other given numbers. This is the Integer Sum problem, shown to be NP-complete in [11].

If a polynomial-time algorithm solves a search problem with success probability $\frac{1}{n^{\text{Const}}}$, this probability can easily be amplified to $3/4$ by repeating the algorithm a polynomial number of times and taking the majority vote as an answer. Therefore, if a polynomially bounded adversary provably worst-case breaks the cryptosystem presented, then $\text{NP} \subseteq \text{RP}$. \square

In §§6 and 7 we present constructions aimed at making invariant-based cryptosystems more reasonable from the security viewpoint.

§6. THE TREE OF GROUPS

6.1. General remarks. The invariant-based protocol described in the preceding section shares a discouraging property with the cryptosystem presented in a remark in §4. It is easy to break in the common cryptographic sense. In this section, we provide a construction that allows us to “hide” these primitives inside a large tree of groups. We can also use it to improve the security of the Anshel–Anshel–Goldfeld key agreement protocol.

We follow the lines of [31] to produce a tree of triples “group–invariant–messages” such that knowing the structure of the tree, one can efficiently calculate the invariant in its root, while without knowing the structure, the invariant is “concealed” behind computationally hard problems in the tree.

In what follows, we concentrate on the invariant-based cryptosystems (introduced in §5), because [31] can be applied directly to the key agreement protocols described in §9. However, we need to develop several new techniques to handle invariant-based cryptosystems. We shall consider the same operations as in [31] and look at what happens with the invariants. But first we introduce some basic notions.

To each vertex v of the tree, we attach a triple (G_v, f_v, M_v) . We produce triples by recursion on the vertices proceeding from the leaves towards the root. At each step, we apply one of the operations described below. For every vertex v , the group G_v is a matrix group $G_v \leq \text{GL}(n, R)$ for some n and some base ring R .

Thus, a tree corresponds to the resulting triple (G, f, M) , where $G \leq \text{GL}(n, R)$ is a group, f is an invariant, that is, a function $f : R^n \rightarrow R$ such that $\forall g \in G \forall x \in R^n f(gx) = f(x)$, and $M \subset R^n$ is a canonical set of *messages* with the property that $\forall m \neq m' \in M f(m) \neq f(m')$.

The public key consists of R, n, G (given by matrix generators), and M . The point of building such a tree is to conceal the secret invariant.

Remark 4. Note that in situations where we change the invariant we can either change the invariant from f to $f \circ h$ or change the message space from M to $h(M)$. Since we care about concealing the invariant, and the message space is to be given publicly, we shall always choose the first alternative.

We want to combine this regular security with the provable worst-case security of the modular group that we have proved in Theorem 1. For this, we place a provably secure construction based on the modular group in one of the leaves of the tree. Then, to solve the membership problem in the root of the tree, Charlie would have to solve the membership problem for all leaves of the tree (our construction has this property).

6.2. Base of recursion. To treat the construction formally, consider a class of groups \mathcal{G} closed under a certain set of group-theoretical operations \mathcal{O} (we list the relevant operations below) defined on triples (G, f, M) and preserving admissibility. For a set $\mathcal{G}_0 \subset \mathcal{G}$ (which is the base of the construction), we define recursively a class $\mathcal{P}(\mathcal{G}_0, \mathcal{O})$ of quadruples (G, f, M, T) in the following way.

- *Base of recursion:* any quadruple (G, f, M, T) , where $G \in \mathcal{G}_0$, (G, f, M) is an admissible triple, and T is a single node labeled by G .
- *Recursive step:* given quadruples $\{(G_i, f_i, M_i, T_i)\}_{i=1}^s$ and an operation $o \in \mathcal{O}$ of arity s , the class $\mathcal{P}(\mathcal{G}_0, \mathcal{O})$ contains the quadruple (G, f, M, T) , where $G = o(G_1, \dots, G_s)$, $f = o(f_1, \dots, f_s)$, $M = o(M_1, \dots, M_s)$, and T is the tree obtained

from T_1, \dots, T_s by adding a new root labeled by o , its sons being the roots of T_1, \dots, T_s .

6.3. Recursive step. Now we list the “building blocks” of the tree, the operations acting on admissible triples. A number of such operations were introduced in [30]; we need to check what happens with the invariants in these cases.

- (1) *Changing the base ring* $\phi : R \rightarrow R'$. If the ring becomes smaller (R' embeds in R with $\varphi : R' \rightarrow R$, and $\phi\varphi = \text{id}$), an invariant f transforms into an invariant $\phi(f)$ that acts like $\phi(f)(x') = f(\varphi(x'))$. If $\forall x \in R^n, g \in G: f(x) = f(gx)$, then

$$\forall x' \in R'^n, g \in G : \phi(f)(gx') = f(g\varphi(x')) = f(\varphi(x')) = \phi(f)f(x').$$

If the ring becomes larger, invariance may break (since there are new elements in the ring, old equalities may fail now). This property allows us to reason that any invariant known from invariant theory over fields will carry on to the rings that are subsets of these fields; e.g., any invariant over \mathbb{C} will be an invariant over \mathbb{Z} .

However, this action requires taking the message space into account. Should different representatives $m, m' \in M$ be such that $\phi(m) = \phi(m')$, the corresponding messages would be identical in the resulting message space $\phi(M)$. Therefore, it is sensible to reduce the underlying ring only if $\phi(M)$ is nontrivial.

- (2) *Conjugation* $g \mapsto h^{-1}gh$. The invariant $f(x)$ becomes the invariant $f'(x) = f(hx)$. If $\forall g \in G, \forall x \in R^n: f(gx) = f(x)$, then

$$\forall g \in G, \forall x \in R^n : f'(h^{-1}ghx) = f(hh^{-1}ghx) = f(g(hx)) = f(hx) = f'(x).$$

The message space M does not change.

- (3) *Direct product* $G_1, G_2 \mapsto G_1 \times G_2$. Here we consider the natural representation of the direct product: if $G_1 \leq GL(n_1, F)$ and $G_2 \leq GL(n_2, F)$, then $G_1 \times G_2 \leq GL(n_1 + n_2, F)$, acting componentwise. In this situation, if $f_1(x), f_2(x)$ are invariants of G_1, G_2 , then any element $f \in \langle f_1(x), f_2(y) \rangle \leq R[x, y]$ is an invariant of $G_1 \times G_2$. We can choose a random element of this set, and the message space will in any case become $M_1 \times M_2$ (if we do not need that many different messages, we can choose several at random and discard the others).
- (4) *Wreath product* $G \wr H$, where $G \leq GL(n, R), H \leq S_m$. In this case, we take the natural representation of $G \wr H$ on R^{mn} acting as

$$(g_1, \dots, g_m, \pi) \begin{pmatrix} x_1 \\ \dots \\ x_m \end{pmatrix} = \begin{pmatrix} g_1 x_{\pi(1)} \\ \dots \\ g_m x_{\pi(m)} \end{pmatrix}.$$

Then, for any invariant f , if $\forall g \in G, x \in R^n: f(gx) = f(x)$, the same will be true for $G \wr H$ if we take f^m to act componentwise. The permutation disturbs nothing in the invariant identity. Accordingly, the message space will grow to M^m (again, we may choose several messages at random or choose the diagonal $\Delta = \{(x, \dots, x) \mid x \in M\}$ if we do not need that many messages).

Apart from the previously considered ways to extend the tree, invariant theory suggests new ways. We can consider several transformations $o \in \mathcal{O}$ that leave the group intact ($o(G) = G$) and only change the invariant f and the message space M . The following will only work if f is a polynomial.

- (1) *Hessian* $H(f)$. If f is a polynomial invariant of G , and $\forall g \in G \leq GL(n, F): \det g = \pm 1$ (note that F is a field), then

$$H(f) = \det \left(\frac{\partial^2 f}{\partial z_i \partial z_j} \right)$$

is also an invariant. The group G and the message space M remain unchanged.

- (2) *Jacobian J* . If f_1, \dots, f_n are polynomial invariants of $G \leq SL(n, F)$ (note that F is a field), then

$$J(f_1, \dots, f_n) = \det \left(\frac{\partial f_i}{\partial z_j} \right)$$

is also an invariant. In this way, we can unite n identical groups with different invariants into one; this will probably be useful only on the first level of the tree, where we can choose arbitrarily many identical groups.

§7. THE LEAVES OF THE TREE

In the previous section, we explained how to build a new invariant out of existing ones (thus, the recursive step). The question that remains is to find the base of this recursion. What should we put in the leaves of this tree?

7.1. General remarks. The first remark to be made is that in computer science, we cannot truly work over \mathbb{C} or \mathbb{R} . Anything we do is actually over \mathbb{Q} . Invariant theory over \mathbb{Q} is a little different from the classical well-known invariant theory over \mathbb{C} . Fortunately, we must not discard the theory: if f is an invariant of a group $G \leq GL(n, \mathbb{C})$ represented by matrices with rational coefficients, then it is still an invariant of the group $G \leq GL(n, \mathbb{Q})$ because the elements of G have rational coefficients. Therefore, in what follows we shall refer to invariants over \mathbb{C} , but they will always be the same for \mathbb{Q} .

We may also look at invariants over finite fields, usually called *modular invariants*, but they provide a completely different story with a completely different theory (see Example 4 in Subsection 7.3).

7.2. Orbit Chern classes. As an example of a standard well-known construction in invariant theory (see, e.g., [56]), we recall the so-called *orbit Chern classes*. They provide most known invariants of finite groups. The idea is simple: take an orbit a^G of an element $a \in F^n$ (suppose for the moment that G acts over a field) and note that $\prod_{b \in a^G} (x + b)$, where x is a formal variable, is invariant under G (the elements of G only permute the factors in this expression). The coefficients of this polynomial are called *orbit Chern classes*. For example, $\sum_{b \in a^G} b$ is an invariant of G , namely, the first orbit Chern class.

All orbit Chern classes are simply symmetric functions in the elements of the orbit; if we take a to be an unknown, we obtain the invariants we are looking for. Similar statements hold for compact groups.

7.3. Examples of finite groups' invariants. In this subsection, we give several examples of invariants of different finite groups. The examples may easily be multiplied.

Example 1. The symmetric group S_n has a monomial representation on $F^n : S_n \rightarrow GL(n, F)$ that permutes the variables. The ring of invariants in this case is generated by all symmetric polynomials, from $x_1 + \dots + x_n$ to $x_1 \cdots x_n$. This is a simple example of orbit Chern classes.

Example 2. A cyclic group \mathbb{Z}_n may be represented by any matrix $g \in GL(m, F)$ such that $g^n = e$ (a unipotent matrix of a matching order). For a function f to be an invariant of a cyclic group's representation, it suffices to ensure that it remains unchanged under the action of the only generator: $f(x) = f(gx)$.

For example, the cyclic group \mathbb{Z}_n is naturally represented by the subgroup generated by $\xi_n e$, where ξ_n is a primitive n th root of unity and e is the identity matrix. Obviously, any homogeneous polynomial of degree n is an invariant of this group. We can go one

step further and consider the representation of the cyclic group \mathbb{Z}_n generated by a matrix of the form

$$\begin{pmatrix} \xi_1 & \dots & 0 \\ \vdots & & \vdots \\ 0 & \dots & \xi_m \end{pmatrix},$$

where the ξ_i are (possibly different) primitive roots of unity, $\xi_i^n = 1$. The invariant ring of this group will be $\mathbb{C}[x_1^n, \dots, x_m^n]$.

Note that invariants depend not only on the groups themselves, but also on their representations; the same group with different representations can have different invariants.

Example 3. A dihedral group D_{2k} has a representation $D_{2k} \rightarrow GL(2, \mathbb{R})$ as the symmetry group of a regular polygon. In this representation, D_{2k} is generated by two matrices:

$$D_{2k} = \left\langle \begin{pmatrix} \cos \frac{2\pi}{k} & -\sin \frac{2\pi}{k} \\ \sin \frac{2\pi}{k} & \cos \frac{2\pi}{k} \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \right\rangle.$$

Then the invariant ring of the dihedral group in this representation is generated by the polynomials

$$q = x^2 + y^2, \quad h = \prod_{i=0}^{k-1} \left(\left(\cos \frac{2\pi i}{k} \right) x + \left(\sin \frac{2\pi i}{k} \right) y \right).$$

Example 4. For an odd prime p , the dihedral group D_{2p} has a representation $D_{2p} \rightarrow GL(2, \mathbb{F}_p)$ over the finite field \mathbb{F}_p . This representation is given by the matrices

$$D_{2k} = \left\langle \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \right\rangle.$$

In this case the invariant ring is isomorphic to $\mathbb{F}_p[y, (xy^{p-1} - x^p)^2]$. However, if we switch to the dual representation (by simply transposing the matrices), the invariants will change substantially; now the ring will be isomorphic to $\mathbb{F}_p[x^2, y(y^{p-1} - x^{p-1})]$. In this example, it was important that the group was represented over a finite field of degree not coprime with the group's degree.

These two examples show how much invariants depend on the actual representation. Some other examples of invariants of finite and classical groups may be found in [25].

7.4. Invariants of classical groups. In this subsection, we give two examples of well-known invariants of classical groups. They may also lie in the leaves of the tree of groups.

Example 5. The orthogonal group in an even dimension $SO(2l, F)$ has the well-known *Dickson invariant*: if $\text{char} F \neq 2$, which we assume to be the case, it is $(-1)^{\det g}$ for $g \in SO(2l, F)$. This invariant works for any field with characteristic not equal to two. Note that this invariant only has two values, so it is good for encrypting only one bit.

Example 6. By definition, the symplectic group $Sp(2n, F)$ preserves a nondegenerate skew-symmetric bilinear form. The value of this form is an invariant (and, unlike the previous example, a polynomial invariant).

§8. ATTACKS ON INVARIANT-BASED CRYPTOSYSTEMS

When a new cryptosystem (or a family of cryptosystems) is presented, it is common to analyze the attacks on such cryptosystems. In this section, we analyze several attacks on invariant-based cryptosystems and give practical advice on how to avoid their success.

8.1. Linear algebra attacks. The most dreaded attacks on algebraic cryptosystems usually go by linear algebra: an adversary constructs a system of linear equations and finds the private key (the most notable example of this approach breaks the Polly Cracker scheme [20], which was only recently augmented with special techniques to make linear algebra attacks less efficient [44]).

Suppose that the invariant f is a polynomial of degree d . In this case, an adversary can view it as a polynomial with $\binom{n+d+1}{d}$ indefinite coefficients. To find the coefficients, he considers the equations $f(g_i m_j) = f(m_j)$ for all elements of the message space $m_j \in M$ and all generators $g_i \in G$. The space of solutions will yield an invariant separating the orbits of M (along with trivial invariants such as $f = \text{const}$, of course). If d is a constant, this attack will actually succeed, so Alice should choose invariants in such a way that $\binom{n+d+1}{d}$ be superpolynomial.

Example 7. Suppose that we are trying to build an invariant-based cryptosystem based on the monomial representation of the symmetric group S_n generated by transpositions τ_{ij} and its first degree invariant

$$f(x_1, \dots, x_n) = x_1 + \dots + x_n.$$

For the message space we should choose a collection of vectors such that the sums of their coordinates are different; we denote them by $m_i = (m_{i1}, \dots, m_{in})$. An adversary performing this kind of attack will simply consider a polynomial

$$h = \lambda_1 x_1 + \dots + \lambda_n x_n$$

and solve a system of equations to ensure that transpositions do not change h . The equation corresponding to τ_{ij} is $h(\tau_{ij}x) = h(x)$, which is equivalent to $\lambda_i = \lambda_j$. So, the adversary will arrive at the correct invariant (or a constant factor of it) after performing a polynomial algorithm. Note that in order to overcome this algorithm, one should choose the message space in such a way that it contains messages with identical sums of elements. The adversary does not need to find *the same* invariant; he only needs to find *an* invariant that separates the vectors of M .

8.2. Monte-Carlo attack and orbit sizes. Another concern comes from the sizes of the orbits of elements of M . Indeed, suppose that an element $m \in M$ has an orbit m^G of polynomial size. In this case, an adversary has a polynomial chance of hitting the correct cipher $E(m)$ by simply picking an element $g \in G$ at random and comparing $E(m)$ and gm . Thus, the elements of the message space should be chosen with care to ensure that their orbits are large.

Example 8. For a trivial yet representative example, consider a message space consisting of a zero vector and some other vector (the following analysis will do for any subgroup of $GL(n, F)$ and any invariant). The size of the zero vector orbit is 1, so that an adversary has no need to do anything: if he sees a zero vector, the message was zero; if he sees a nonzero vector, it was the other vector that got “encrypted”.

8.3. Tree reconstruction attack. Finally, an adversary may attempt to reconstruct the tree with which the invariant was built. Along this way he will encounter, for example, the problem of finding a matrix a such that $a^{-1}Ga = H$ for given G and H . This is a well-known hard problem; for example, in [43], it was shown that the graph isomorphism problem reduces to the problem of group conjugation. This kind of attacks was considered in detail in [30]; the same argument applies in this case, since the task of reconstructing the tree has not become any easier. In fact, it has become harder, as the tree nodes are now augmented with invariants that may change nontrivially when going up the tree; consequently, to reconstruct a tree an adversary needs not only to reconstruct the groups, but also to reconstruct invariants.

§9. ANSHEL–ANSHEL–GOLDFELD KEY AGREEMENT PROTOCOL
SECURE AGAINST PROVABLE BREAK

First, we recall the construction of the Anshel–Anshel–Goldfeld key agreement protocol [3]. Let G be a group, and let two players A and B choose two subgroups of G :

$$G_A = \langle a_1, \dots, a_m \rangle, \quad G_B = \langle b_1, \dots, b_n \rangle.$$

Remark 5. Note that all arguments below go without change if instead of G_A and G_B we consider subsemigroups of G , \tilde{G}_A and \tilde{G}_B , generated by the same elements $\langle a_1, \dots, a_m \rangle$ and $\langle b_1, \dots, b_n \rangle$, respectively, but generated as semigroups rather than as groups. All commutators are taken in the larger group G .

The group G and elements a_i , $1 \leq i \leq m$, and b_j , $1 \leq j \leq n$, are made public. Both players A and B randomly choose secret elements $a \in G_A$ and $b \in G_B$ as products of no more than N generators and transmit the following sequences to each other:

$$X_A = \{a^{-1}b_ja\}_{j=1}^n, \quad X_B = \{b^{-1}a_ib\}_{i=1}^m.$$

After this transmission, player A (respectively, B) has a representation of the element a (respectively, b) in the subgroup G_A (respectively, G_B). Therefore, he can compute a representation of the element $b^{-1}ab$ (respectively, $a^{-1}ba$) by using elements of the sequence X_A (respectively, X_B). Thus, both players have shared a common key, namely, the commutator

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

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

To provably break the Anshel–Anshel–Goldfeld key agreement protocol, one needs to find representations of certain elements a' in G_A and b' in G_B , where

$$X_A = \{a'^{-1}b_ja'\}_{j=1}^n, \quad X_B = \{b'^{-1}a_ib'\}_{i=1}^m.$$

Theorem 2. *The Anshel–Anshel–Goldfeld key agreement protocol for a modular group G and its subgroups G_A and G_B is secure against a provable worst-case break unless $NP \subseteq RP$. The same statement is true if, instead of G_A and G_B , we consider subsemigroups of G , \tilde{G}_A and \tilde{G}_B , generated by the same elements $\langle a_1, \dots, a_m \rangle$ and $\langle b_1, \dots, b_n \rangle$, respectively, but generated as semigroups rather than as groups.*

Proof. Assume that there is a probabilistic polynomial-time Turing machine M such that for infinitely many security parameters N and an input $I = \{a_1, \dots, a_m, b_1, \dots, b_n, a^{-1}b_1a, \dots, a^{-1}b_ma, b^{-1}a_1b, \dots, b^{-1}a_nb\}$ we have

$$\Pr[M(I) = a'_1, s_1, \dots, a'_f, s_f, b'_1, t_1, \dots, b'_g, t_g] \geq 1/p(N),$$

where $G_A = \langle a_1, \dots, a_m \rangle$ and $G_B = \langle b_1, \dots, b_n \rangle$ are subgroups of the modular group, $a \in G_A$, $b \in G_B$, $a' = \prod_{i=1}^f a_i^{s_i}$, $b' = \prod_{j=1}^g b_j^{t_j}$, $a'_i \in \{a_i\}_{i=1}^m$, $b'_j \in \{b_j\}_{j=1}^n$, $a'^{-1}b_ja' = a^{-1}b_ja$ for all $1 \leq j \leq n$, $b'^{-1}a_ib' = b^{-1}a_ib$ for all $1 \leq i \leq m$, s_i and t_j are in $\{-1, 1\}$ for all $1 \leq i \leq f$ and $1 \leq j \leq g$, $f, g \leq N$, and p is some polynomial. Note that we can check the correctness of the answer of M , so we also assume that M produces only correct answers.

Using M , we can construct a probabilistic polynomial time Turing machine M' that contains $p(N)/2$ copies of M and is such that on the input $(X, \{Y_i\}_i, N)$ it does the following.

- (1) If $X = \prod_{i=1}^m Y_i^{s_i}$, where $Y' \in \{Y_i\}_i$, $m \leq N$, $s_i \in \{-1, 1\}$ (if we regard G_A and G_B as semigroups, here we take positive degrees only), then $\Pr[M' \text{ accepts}] \geq 1/2$.
- (2) Otherwise, $\Pr[M' \text{ accepts}] = 0$.

For inputs of all copies of M we take $a = b = X, a_i = b_i = Y_i$, and compute all $a^{-1}b_1a, \dots, a^{-1}b_m a, b^{-1}a_1b, \dots, b^{-1}a_nb$ in polynomial time. By [11, Corollary 11.5], the BM problem is NP-complete; hence, $\text{NP} \subseteq \text{RP}$. \square

Remark 6. If G_A and G_B are semigroups, the BM problem is hard on the average [57].

Note that the key agreement protocol described above can be insecure against a linear algebra attack (cf. Subsection 8.1): it gives an adversary the decision of the conjugacy problem, which could be unique, provided that the ring generated by G_A (or by G_B) coincides with the entire ring of matrices (that is the case if we build our protocol on the Blass-Gurevich groups). To make a cryptosystem more resistant against linear algebra attacks, one can replace G by a tree-like construction of groups or semigroups as in §6.

Formally speaking, we produce the following recursive construction for a class of groups \mathcal{G} closed under a certain set of group-theoretical operations \mathcal{O} ; this time we need not worry about admissible triples, and the operations are defined simply on groups of \mathcal{G} . For a set $\mathcal{G}_0 \subset \mathcal{G}$ (which is the base of the construction) we define recursively a class $\mathcal{P}(\mathcal{G}_0, \mathcal{O})$ of pairs (G, T) .

The recursive definition is done precisely as in Subsection 6.2, omitting the constructions of invariants and message spaces. In our case, the set \mathcal{O} of admissible operations consists of changing the underlying ring, direct products, wreath products, and conjugations (similar to invariant-based cryptosystems, but without invariant-specific operations).

The security of the Anshel–Anshel–Goldfeld key agreement protocol for matrix groups is based on the following problem.

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

If an adversary can efficiently solve LTP, he can obviously break the Anshel–Anshel–Goldfeld protocol. In [31], the following proposition was proved (Lemma 3.4).

Proposition 1. *Let $G \in \mathcal{G}$. Then, given a derivation tree of G , LTP for G can be solved in time polynomial in the size of the tree and the times of solving LTP for leaves of the tree.*

Of course, this does not prove that the security of the Anshel–Anshel–Goldfeld key agreement protocol in the root of the tree depends on the security of this protocol in the leaves of the tree. We have a much weaker statement that goes in the undesirable direction *twice*: if we can solve LTP, we can break the Anshel–Anshel–Goldfeld protocol, and if we can solve LTP for leaves of the tree, we can solve LTP for its root. To prove security, we would need to reverse both statements. However, this is the best we can do, and we know of no similar constructions with stronger dependencies.

§10. CONCLUSIONS AND FURTHER WORK

In the paper, we have introduced a new notion of a provable break and of provable security in general. While this notion is undoubtedly much weaker than regular cryptographic security, it appears natural, well-defined, and sensible. Moreover, this notion of security is one of the few known notions for which provable positive statements are possible. We have provided two examples of cryptographic protocols: an invariant-based cryptosystem secure against provable break (we have also substantially advanced the theory of invariant-based cryptosystems since [25]) and a key agreement protocol secure against provable break, a special case of the Anshel–Anshel–Goldfeld key agreement protocol. We are sure that more examples can be produced along the same lines.

Therefore, on the one hand, further work lies in the search for more cryptographic primitives secure against a provable break. On the other hand, it is desirable to look for relationships between the provable break and other notions of security. It is easy to think of a trivial cryptosystem for which provable security is equivalent to regular cryptographic security (for example, Bob may not use random bits at all); however, it may be useful to look for nontrivial examples of the same. These lines will probably be similar to the research carried out by Ajtai and Dwork [1], later augmented by Regev [51, 52]. They managed to reduce a worst-case problem to an average-case one and thus produced a cryptosystem that is secure under some worst-case assumptions.

ACKNOWLEDGMENTS

The authors are grateful to Edward A. Hirsch for valuable discussions and for Remark 2.

REFERENCES

- [1] M. Ajtai and C. Dwork, *A public-key cryptosystem with worst-case/average-case equivalence*, STOC'97: 29th Annual ACM Symposium on Theory of Computing (El Paso, TX, 1997), ACM, New York, 1999, pp. 284–293 (electronic). MR1715640
- [2] I. Anshel, M. Anshel, B. Fisher, and D. Goldfeld, *New key agreement protocols in braid group cryptography*, Topics in Cryptology — CT-RSA 2001 (San Francisco, CA), Lecture Notes in Comput. Sci., vol. 2020, Springer, Berlin, 2001, pp. 13–27. MR1907085 (2003c:94016)
- [3] I. Anshel, M. Anshel, and D. Goldfeld, *An algebraic method for public-key cryptography*, Math. Res. Lett. **6** (1999), 287–291. MR1713130 (2000e:94034)
- [4] ———, *Non-abelian key agreement protocols*, Discrete Appl. Math. **130** (2003), no. 1, 3–12. MR2008401 (2004h:94050)
- [5] M. Anshel, *Braid group cryptography and quantum cryptoanalysis*, 8th Internat. Wigner Symposium (New York, U.S.A., 2003), Baruch College of CUNY, pp. 13–27.
- [6] T. M. Apostol, *Modular functions and Dirichlet series in number theory*, Grad. Texts in Math., vol. 41, Springer, New York, 1990. MR1027834 (90j:11001)
- [7] S. Arora and B. Barak, *Complexity theory: a modern approach*, <http://www.cs.princeton.edu/theory/complexity/>, 2008.
- [8] J. Benaloh, *Dense probabilistic encryption*, 1st Annual Workshop on Selected Areas in Cryptology (Queen's Univ., Kingston, Canada, 1994), Springer-Verlag, London, 1994, pp. 120–128.
- [9] C. H. Bennett and P. W. Shor, *Quantum information theory*, Information Theory: 1948–1998, IEEE Trans. Inform. Theory **44** (1998), no. 6, 2724–2742. MR1658902 (99h:94025)
- [10] I. Blake, G. Seroussi, and N. Smart, *Elliptic curves in cryptography*, London Math. Soc. Lecture Note Ser., vol. 265, Cambridge Univ. Press, Cambridge, 2000. MR1771549 (2001i:94048)
- [11] A. Blass and Y. Gurevich, *Matrix transformation is complete for the average case*, SIAM J. Comput. **24** (1995), 3–29. MR1313476 (96f:68037)
- [12] N. Blum, *A Boolean function requiring $3n$ network size*, Theoret. Comput. Sci. **28** (1984), 337–345. MR0742295 (86g:68053)
- [13] A. Bogdanov and L. Trevisan, *On worst-case to average-case reductions for NP problems*, 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS'03) (Cambridge, MA, U.S.A., 2003), IEEE Computer Soc., Washington, DC, 2003, pp. 308–317.
- [14] D. Boneh and R. Venkatesan, *Breaking RSA may not be equivalent to factoring*, Advances in Cryptology — EUROCRYPT'98 (Espoo), Lecture Notes in Comput. Sci., vol. 1403, Springer, Berlin, 1998, pp. 59–71. MR1729052
- [15] D. R. L. Brown, *Breaking RSA may be as difficult as factoring*, Tech. Rep. 2005/380, Cryptology ePrint Archive, 2005.
- [16] W. Diffie and M. E. Hellman, *New directions in cryptography*, IEEE Trans. Inform. Theory **IT-22** (1976), 644–654. MR0437208 (55:10141)
- [17] C. Dwork, *Positive applications of lattices to cryptography*, Mathematical Foundations of Computer Science 1997 (Bratislava) (MFCS'97), Lecture Notes in Comput. Sci., vol. 1295, Springer, Berlin, 1997, pp. 44–51. MR1640207 (99k:94034)
- [18] S. Even and Y. Yacobi, *Cryptocomplexity and NP-completeness*, Automata, Languages and Programming (Proc. Seventh Internat. Colloq., Noordwijkerhout, 1980), Lecture Notes in Comput. Sci., vol. 85, Springer, Berlin–New York, 1980, pp. 195–207. MR0589004 (82a:94085)

- [19] J. Feigenbaum and M. Merritt, *Open questions, talk abstracts, and summary of discussions*, Distributed Computing and Cryptography (Princeton, NJ, 1989), DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 2, Amer. Math. Soc., Providence, RI, 1991, pp. 1–45. MR1105537 (92b:68006)
- [20] M. Fellows and N. Kobitz, *Combinatorial cryptosystems galore!* Finite Fields: Theory, Applications, and Algorithms (Las Vegas, NV, 1993), Contemp. Math., vol. 168, Amer. Math. Soc., Providence, RI, 1994, pp. 51–61. MR1291417 (95e:94028)
- [21] O. Goldreich, *Introduction to complexity theory*, Lecture notes, Weizmann Inst. Sci., 1998–1999.
- [22] S. Goldwasser and M. Bellare, *Lecture notes on cryptography*, Summer course on cryptography at MIT, 2001.
- [23] S. Goldwasser and S. Micali, *Probabilistic encryption*, J. Comput. System Sci. **28** (1984), 270–299. MR0760548 (86j:94047)
- [24] D. Grigoriev, *Testing shift-equivalence of polynomials by deterministic, probabilistic and quantum machines*, Theoret. Comput. Sci. **180** (1997), 217–228. MR1453867 (98b:68090)
- [25] ———, *Public-key cryptography and invariant theory*, Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI) **293** (2002), 26–38; English transl., J. Math. Sci. (N. Y.) **126** (2005), no. 3, 1152–1157. MR1948823 (2003m:94065)
- [26] D. Grigoriev, E. A. Hirsch, and K. Pervyshev, *A complete public-key cryptosystem*, Groups, Complexity, Cryptology **1** (2009), 1–12. MR2502933
- [27] D. Grigoriev, A. Kojevnikov, and S. I. Nikolenko, *Invariant-based cryptosystems and their security against provable break*, Tech. Rep. 158, Max-Planck-Inst. Preprints, 2007.
- [28] D. Grigoriev and I. Ponomarenko, *Homomorphic public-key cryptosystems over groups and rings*, Complexity of Computations and Proofs, Quad. Mat., No. 13, Dept. Math. Seconda Univ. Napoli, Caserta, 2004, pp. 305–325. MR2131411 (2006b:94027)
- [29] ———, *On non-abelian homomorphic public-key cryptosystems*, Zap. Nauchn. Sem. S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI) **293** (2002), 39–58; English transl., J. Math. Sci. (N. Y.) **126** (2005), no. 3, 1158–1166. MR1948824 (2004a:94044)
- [30] ———, *Homomorphic public-key cryptosystems and encrypting Boolean circuits*, Appl. Algebra Engrg. Comm. Comput. **17** (2006), 239–255. MR2233784 (2008b:94067)
- [31] ———, *Constructions in public-key cryptography over matrix groups*, Algebraic Methods in Cryptography (L. Gerritzen, D. Goldfeld, M. Kreuzer, R. Gerhard, and V. Shpilrain, eds.), Contemp. Math., vol. 418, Amer. Math. Soc., Providence, RI, 2006, pp. 103–119. MR2389292
- [32] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to elliptic curve cryptography*, Springer-Verlag, New York, 2004. MR2054891 (2005c:94049)
- [33] D. Harnik, J. Kilian, M. Naor, O. Reingold, and A. Rosen, *On robust combiners for oblivious transfer and other primitives*, Advances in Cryptology — EUROCRYPT 2005, Lecture Notes in Comput. Sci., vol. 3494, Springer, Berlin, 2005, pp. 96–113. MR2352183 (2008i:94043)
- [34] A. P. Hiltgen, *Constructions of feebly-one-way families of permutations*, Advances in Cryptology — AUSCRYPT’92 (Gold Coast, 1992), Lecture Notes in Comput. Sci., vol. 718, Springer, Berlin, 1993, pp. 422–434. MR1292706 (96e:94014)
- [35] ———, *Towards a better understanding of one-wayness: Facing linear permutations*, Advances in Cryptology — EUROCRYPT’98 (Espoo), Lecture Notes in Comput. Sci., vol. 1403, Springer, Berlin, 1998, pp. 319–333. MR1729060 (2000i:94050)
- [36] E. A. Hirsch and S. I. Nikolenko, *A feebly trapdoor function*, PDMI Preprints no. 16/2008, S.-Peterburg. Otdel. Mat. Inst. Steklov. (POMI), St. Petersburg, 2008.
- [37] N. Kobitz, *Elliptic curve cryptosystems*, Math. Comp. **48** (1987), 203–209. MR0866109 (88b:94017)
- [38] A. Kojevnikov and S. I. Nikolenko, *New combinatorial complete one-way functions*, 25th Symposium on Theoretical Aspects of Computer Science (STACS’08) (Bordeaux, February 2008), Bordeaux, 2008.
- [39] A. Lempel, *Cryptography in transition*, Comput. Surveys **11** (1979), no. 4, 215–220.
- [40] L. A. Levin, *One-way functions and pseudorandom generators*, Combinatorica **7** (1987), 357–363. MR0931193 (89c:68048)
- [41] ———, *One-way functions*, Problemy Peredachi Inf. **39** (2003), no. 1, 103–117; English transl., Probl. Inf. Transm. **39** (2003), no. 1, 92–103. MR2101668 (2005g:94080)
- [42] H.-K. Lo, T. Spiller, and S. Popescu (eds.), *Introduction to quantum computation and information*, World Sci. Publ. Co., Inc., River Edge, NJ, 1998. MR1750536 (2000k:81058)
- [43] E. M. Luks, *Permutation groups and polynomial-time computation*, Groups and Computation (New Brunswick, NJ, 1991), DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 11, Amer. Math. Soc., Providence, RI, 1993, pp. 139–175. MR1235801 (94h:20005)
- [44] L. V. Ly, *Polly Two: A new algebraic polynomial-based public-key scheme*, Appl. Algebra Engrg. Comm. Comput. **17** (2006), 267–283. MR2233786 (2007c:94149)

- [45] V. S. Miller, *Use of elliptic curves in cryptography*, Advances in Cryptology — CRYPTO'85 (Santa Barbara, CA, 1985), Lecture Notes in Comput. Sci., vol. 218, Springer, Berlin, 1986, pp. 417–426. MR0851432 (88b:68040)
- [46] A. G. Myasnikov, V. Shpilrain, and A. Ushakov, *Group-based cryptography*, Birkhäuser, Basel, 2008. MR2437984 (2009d:94098)
- [47] D. Naccache and J. Stern, *A new public-key cryptosystem based on higher residues*, 5th ACM Conference on Computer and Communication Security (San Francisco, CA, U.S.A., 1998), ACM Press, San Francisco, CA, 1998, pp. 59–66. MR1603095
- [48] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*, Cambridge Univ. Press, Cambridge, 2000. MR1796805 (2003j:81038)
- [49] T. Okamoto and S. Uchiyama, *A new public-key cryptosystem as secure as factoring*, Advances in Cryptology — EUROCRYPT'98 (Espoo), Lecture Notes in Comput. Sci., vol. 1403, Springer, Berlin, 1998, pp. 308–318. MR1729059
- [50] D. K. Rappe, *Algebraisch homomorphe Kryptosysteme*, Diplomarbeit, Dem Fachbereich Mathematik der Universität Dortmund, 2000.
- [51] O. Regev, *On lattices, learning with errors, random linear codes, and cryptography*, STOC'05: 37th Annual ACM Symposium on Theory of Computing (Baltimore, MD, U.S.A., 2005), ACM, New York, 2005, pp. 84–93. MR2181605 (2006g:94031)
- [52] ———, *Lattice-based cryptography*, 26th Annual International Cryptology Conference (CRYPTO'06) (Santa Barbara, CA, U.S.A., 2006), Lecture Notes in Comput. Sci., vol. 4117, Springer, Berlin, 2006, pp. 131–141. MR2422159
- [53] R. Rivest, A. Shamir, and L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Comm. ACM **21** (1978), no. 2, 120–126. MR0700103 (83m:94003)
- [54] R. L. Rivest and B. Kaliski, *RSA problem*, Encyclopedia of Cryptography and Security, Kluwer Publ. House, 2005.
- [55] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput. **26** (1997), no. 5, 1484–1509. MR1471990 (98i:11108)
- [56] L. Smith, *Polynomial invariants of finite groups*, Res. Notes Math., vol. 6, A. K. Peters, Wellesley, MA, 1995. MR1328644 (96f:13008)
- [57] R. Venkatesan and S. Rajagopalan, *Average case intractability of matrix and diophantine problems*, STOC'92: 24th Annual ACM Symposium on Theory of Computing (Victoria, British Columbia, Canada, May 1992), ACM, New York, 1992, pp. 632–642.
- [58] L. Washington, *Elliptic curves. Number theory and cryptography*, Chapman & Hall / CRC, Boca Raton, FL, 2003. MR1989729 (2004e:11061)
- [59] I. Wegener, *The complexity of Boolean functions*, John Wiley and Sons, Ltd., Chichester; B. G. Teubner, Stuttgart, 1987. MR0905473 (89b:03066)
- [60] A. Yao, *How to generate and exchange secrets*, 27th IEEE Symposium on the Foundations of Computer Science (FOCS'86) (Toronto, Canada, October 1986), IEEE Computer Society, 1986, pp. 162–187.

CNRS, LABORATOIRE DES MATHÉMATIQUES, UNIVERSITÉ DE LILLE, 59655 VILLENEUVE D'ASCQ, FRANCE

E-mail address: `Dmitry.Grigoryev@math.univ-lille1.fr`

ST. PETERSBURG BRANCH, STEKLOV MATHEMATICAL INSTITUTE, RUSSIAN ACADEMY OF SCIENCES, FONTANKA 27, 191023 ST. PETERSBURG, RUSSIA

E-mail address: `arist@pdm1.ras.ru`

ST. PETERSBURG BRANCH, STEKLOV MATHEMATICAL INSTITUTE, RUSSIAN ACADEMY OF SCIENCES, FONTANKA 27, 191023 ST. PETERSBURG, RUSSIA

E-mail address: `sergey@logic.pdm1.ras.ru`

Received 9/JAN/2008

Translated by THE AUTHORS