VERSITA

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.2478/v10127-012-0042-8
Tatra Mt. Math. Publ. **53** (2012), 119–135

AN LWE-BASED KEY TRANSFER PROTOCOL WITH ANONYMITY

Adela Georgescu

ABSTRACT. We introduce a new cryptographic protocol based on the well-known Learning With Errors (LWE) problem: a group key transfer protocol which achieves *anonymity* of the members against each others. This issue is almost absent in the key transfer protocols from the literature but we argue it is a practical property. We motivate our construction by a practical need. We use two essential cryptographic primitives built from LWE: LWE Diffie-Hellman key exchange derived from Regev's work [Regev, O.: *On lattices, learning with errors, random linear codes, and cryptography*, in: Proc. of the 37th Annual ACM Symposium on Theory of Computing—STOC '05 (H. N. Gabow and R. Fagin, eds.), Baltimore, MD, USA, 2005, ACM, New York, 2005, pp. 84–93] and a public key cryptosystem secure under the LWE hardness. We provide a security definition for anonymous key transfer protocol and we achieve anonymity against IND-CPA adversaries.

1. Introduction

In this paper, we present a group key transfer protocol for the following problem: a manager wants to build a team of experts from a large database and establish a shared communication secret key with them. The current protocol has two important properties: it is built in the lattice-based cryptography and achieves *anonymity* of the team members against each others. We already introduced in previous papers [1], [2] two different solutions for the same problem in the classic cryptography. In the latter, we used almost the same cryptographic primitives: the traditional Diffie-Hellman key exchange for anonymity and public key encryption scheme for confidentiality. However, we did not manage to obtain ciphertext linear in the size on the target team, but rather linear in the size of the whole universe of users.

2010 Mathematics Subject Classification: 62K05.
Keywords: LWE, lattice-based cryptography, anonymity.
This research was supported by the European Social Fund, under doctoral and postdoctoral grant POSDRU/88/1.5/S/56668.

As far as we know, our construction is the first key transfer protocol built from LWE in the lattice-based cryptography. In recent years, lattices have served as a very attractive theoretic (not yet practical) alternative to the traditional number theory. They offer hard problems which lay at the basis of security for many cryptographic primitives, great simplicity and relatively efficient implementations; they are also believed to be secure against quantum attacks. With so many advantages, it is naturally to try to transfer as many cryptographic primitives in this field.

Anonymity is an important property that we achieve in our protocol, using a similar technique as [8] but different tools as we explain below. Maybe in key agreement protocols, anonymity is not needed or moreover, is undesirable, but in our protocol we find it very practical. Let us present a small real example where it is needed. Consider a scenario where there are several experts which evaluate papers submitted to conferences. The experts belong to a database but they do not need to know each other, they just need to have a secret shared key in order to communicate securely. They are conducted by a manager who can choose a team to participate in a certain project and who has to transfer securely a secret key to all the team members.

Our contribution. In this paper, we construct a group key transfer from lattices (LWE) achieving anonymity. Our approach is as follows. The first phase of the protocol is a lattice Diffie-Hellman key exchange that the manager carries with each expert in the database. The security of the key exchange relies on the difficulty of the Learning With Errors problem, a very famous problem proven to be as hard as certain lattice problems in the worst case. In the second phase, after selecting a team, the manager chooses a secret key and transfers it to the team. He uses the cryptosystem from [6], the "dual" of R e g e v cryptosystem [11] to hide the secret key. We achieve anonymity by providing a way for the manager to securely inform only the selected experts about the ciphertext component that is intended for them. More precisely, the manager computes a ciphertext whose size is linear in the size of the team and which contains pairs of elements for every selected member $\{(H_{\pi(1)}, C_{\pi(1)}), \ldots, (H_{\pi(k)}, C_{\pi(k)})\}$. The first element of every pair serves as an indicator for the right ciphertext addressed to a team member. So the expert is able to compute the first element in the pair based on the Diffie Hellman secret key he shares with the manager. Then he identifies the right pair for him in the ciphertext and is able to decrypt the second component and recover the shared secret key. This situation allows only the experts selected to decrypt the shared key.

We formalize the notion of anonymity similar to [8] and prove our protocol *anonymous IND-CPA* secure in a security model we present in Subsection 2.6. We achieve constant decryption time of the secret key and ciphertext size linear in the size of the selected team. Unfortunately, the number of encryptions is

a little bit oversized since we perform a Diffie-Hellman key exchange with every user in the universe.

Related work. Our cryptographic construction is based on the hardness of the learning with errors problem (LWE). We can not compare it to other lattice--based key transfer protocols, since ours is the first key transfer protocol of this type. Anyway, a drawback of our protocol is the number of Diffie-Hellman key exchanges linear in the size of the universe of experts. If we take a look to a broadcast encryption scheme [12] based on lattices, we can say our scheme is less efficient in terms of number of equations, but note that we drop efficiency for the sake of anonymity. When compared to anonymous broadcast encryption from [8], we obtain the same constant decryption time and also ciphertext linear in $|T|$, the size of the team; for encryption, we also require a number of operations linear in $|T|$.

2. Preliminaries

2.1. Lattices

A lattice is the set $\Lambda = L(\mathbf{B}) = \left\{\sum_{i=1}^{n} a_i \mathbf{b_i} | a_i \in \mathbb{Z}\right\}$ of all linear integer combination of the linearly independent vectors $\mathbf{B} = \{\mathbf{b_1}, \ldots, \mathbf{b_n}\}$ which constitute a *basis* of the lattice.

As most of the cryptographic applications do, we use in our work the following two types of lattices called *modular* lattices or $q - ary$ lattices. Given some positive numbers m, n and q and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the first type of lattice contains all the vectors that are orthogonal to the rows of $\mathbf{A}$:

$$\Lambda^{\perp}(\mathbf{A}) = \left\{\mathbf{z} \in \mathbb{Z}^m \mid \mathbf{Az} = \mathbf{0} \bmod \mathbf{q}\right\}.$$

The second type of lattice is generated by the rows of matrix $\mathbf{A}$

$$\Lambda(\mathbf{A}) = \left\{\mathbf{z} \in \mathbb{Z}^m \mid \exists\, \mathbf{s} \in \mathbb{Z}_q^n \text{ so that } \mathbf{z} = \mathbf{A}^{\top}\mathbf{s} \bmod \mathbf{q}\right\}.$$

For a set of linearly independent vectors $S = \{s_1, \ldots, s_n\} \subset \mathbb{R}^n$, denote by $\tilde{S} = \{\tilde{s_1}, \ldots, \tilde{s_n}\}$ its Gram-Schmidt orthogonalization, defined as follows: $\tilde{s_1} = s_1$, and for $i = 2, \ldots, n$, $\tilde{s_i}$ is the component of s_i orthogonal to $\mathsf{span}(s_1, \ldots, s_{i-1})$. Note that $||\tilde{s_i}|| \leq ||s_i||$.

Discrete Gaussian Distribution. In lattice-based cryptography, "error" (perturbation) vectors are typically chosen according to the Gaussian distribution D_α which chooses each $\mathbf{x}$ with the following probability given by the Gaussian function centered at $\mathbf{c}$ with parameter r:

$$\forall \mathbf{x} \in \mathbb{R}^n, \rho_{r,c}(x) = exp\big(-\pi||\mathbf{x} - \mathbf{c}||^2/r^2\big).$$

For n-dimensional lattice Λ, the discrete Gaussian distribution over Λ is defined as

$$\forall \mathbf{x} \in \Lambda, D_{\Lambda,r,c}(x) = \frac{\rho_{r,\mathbf{c}}(\mathbf{x})}{\rho_{r,\mathbf{c}}(\Lambda)}.$$

Assume that the columns of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ generate $\mathbb{Z}_q^n$, fix $\mathbf{u} \in \mathbb{Z}_q^n$ and let $\mathbf{t} \in \mathbb{Z}^m$ be an arbitrary solution from the coset of $\Lambda^{\perp}(\mathbf{A})$ defined as $\Lambda^{\perp}(\mathbf{A})_y = \{\mathbf{z} \in \mathbb{Z}^m | \mathbf{Az} = \mathbf{y} \bmod \mathbf{q}\} = \mathbf{t} + \mathbf{\Lambda}^{\perp}(\mathbf{A})$. The discrete Gaussian distribution over $\Lambda_y^{\perp}(\mathbf{A})$ which is the conditional distribution of $D_{\mathbb{Z}^m,r}$ given $\mathbf{Az} = \mathbf{y} \bmod q$ is given by

$$\forall \mathbf{x} \in \Lambda, D_{\Lambda_y^{\perp}(\mathbf{A}),r}(x) = \frac{\rho_r(\mathbf{x})}{\rho_r\big(\mathbf{t} + \Lambda^{\perp}(\mathbf{A})\big)}.$$

2.2. Preimage sampleble functions

In [6] a collection of one-way preimage sampleble functions is described with the following two properties:

- the input $\mathbf{x}$ belongs to a Gaussian distribution while the output $\mathbf{y}$ is statistically close to uniform;
- given a trapdoor, the inversion algorithm not just samples an arbitrary preimage of $\mathbf{y}$ but samples an input from the Gaussian distribution under the condition that $f(\mathbf{x}) = \mathbf{y}$.

We are interested in the function introduced by A j t a i in [3] which simply chooses a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and evaluates the linear function $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{Ax} \bmod q$. The Small Integer Solution problem claims that inverting this function is hard when the vector $\mathbf{x}$ is small.

Short Integer Solution problem $SIS_{q,m,b}$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ from the uniform distribution, find $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{Ax} = \mathbf{0} \bmod q$ and $0 < ||\mathbf{x}|| < b$.

A variant of this problem, namely to find a short solution to a random *inhomogeneous* system $\mathbf{Ae} = \mathbf{u} \bmod q$ can be formalized as below.

Inhomogeneous Short Integer Solution problem $ISIS_{q,m,b}$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ from the uniform distribution and a syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ find $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{Ax} = \mathbf{u} \bmod q$ and $0 < ||\mathbf{x}|| < b$.

Both SIS and $ISIS$ problems are proven to be as hard on average as certain worst-case problems on lattices. We state the following result from [6] on this issue.

Proposition 1 ([6]). *For the following setting of parameters: $m, b = poly(n)$ and for any prime $q \geq b \cdot \omega\big(\sqrt{(n \log n)}\big)$, the average-cse problems $SIS_{q,m,b}$ and $ISIS_{q,m,b}$ are as hard as approximating the SIVP problem in the worst-case to within $\gamma = b \cdot O(\sqrt{n})$ factors.*

In order to present some concrete constructions [6] of the preimage sampleble functions that we will employ in our protocol, we first show a result of A j t a i [3] which can be transformed into an algorithm we need.

Proposition 2 ([3]). *For any prime $q = poly(n)$ and $m \geq 5n\ log\ q$, there is a probabilistic polynomial time algorithm that, on input 1^n, generates matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ statistically close to uniform and set $S \subset \Lambda^{\perp}(\mathbf{A})$ with length $||S|| \leq L = m^{2.5}$ (length of S denoted by $||S||$ refers to the length of its longest column).*

In a result from [10] it is shown how the set S can be converted efficiently to a "good" basis $\mathbf{T}$ of $\Lambda^{\perp}(\mathbf{A})$ such that $||\tilde{\mathbf{T}}|| \leq ||\tilde{\mathbf{S}}|| \leq L$.

Now, let us present the two algorithms we will use in our protocol.

- $\mathsf{TrapGen}(1^n)$ [6] — generating a function with trapdoor. Let n, q, m be integers with $q \geq 2$, $m \geq 2n\ lgq$, algorithm $\mathsf{TrapGen}(1^n)$ outputs a pair $(\mathbf{A}, \mathbf{T})$ such that $\mathbf{A} \in \mathbb{Z}^{n \times m}$ is statistically close to uniform and $\mathbf{T}$ is a good basis of $\Lambda^{\perp}(\mathbf{A})$ such that $||\tilde{\mathbf{T}}|| \leq m \cdot \omega\big(\sqrt{log\ m}\big)$.
- $\mathsf{SamplePre}(\mathbf{A}, \mathbf{B}, \mathbf{y}, r)$ [6] — allows preimage sampling of the function $f_{\mathbf{A}}$ given a short basis for $\Lambda_q^{\perp}(\mathbf{A})$: on input of $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, a good basis $\mathbf{B}$ for $\Lambda^{\perp}(\mathbf{A})$ as the trapdoor, a vector $\mathbf{y} \in \mathbb{Z}_q^n$ and r; the conditional distribution of the output $\mathbf{e}$ is within negligible statistical distance of $D_{\Lambda_y^{\perp}(\mathbf{A}), r}$.

Theorem 1 ([6]). *The two algorithms described above give a collection of one-way preimage sampleble functions if $ISIS_{q,m,s\sqrt{m}}$ is hard.*

2.3. The Learning With Errors problem

The learning with errors problem (LWE) is a very well known problem in the field of lattice-based cryptography. Even if it is not related directly to lattices, the security of many cryptographic primitives in this field rely on its hardness believed to be the same as worst-case lattice problems.

Let us formally describe the LWE problem [11]. Fix the following parameters of the problem: $n \geq 1$, modulus $q \geq 2$ and Gaussian error probability distribution χ on $\mathbb{Z}_q$ (more precisely it is chosen to be the normal distribution rounded to the nearest integer, modulo q with standard deviation αq where $\alpha > 0$ is taken to be $1/(poly(n))$). The distribution of choosing the secret $\mathbf{s} \in \mathbb{Z}_q^n$, vector $\mathbf{a}$ uniformly at random from $\mathbb{Z}_q^n$, choosing e according to χ and outputting $(\mathbf{a}, \mathbf{a^T s} + e)$, over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ is denoted as $A_{s,\chi}$.

The goal of the LWE problem with modulus q and error distribution χ ($LWE_{q,\chi}$) is, given an arbitrary number of samples from $A_{s,\chi}$, to output $\mathbf{s}$ with high probability.

The decisional version of LWE requires to distinguish between LWE samples and uniformly chosen samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

Proposition 3 ([11]). *Let $\alpha = \alpha(n) \in (0,1)$ and let $q = q(n)$ be a prime such that $\alpha q > 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm that solves $LWE_{q,\chi}$, then there exists an efficient quantum algorithm for approximating SIVP in the worst-case to within $O(n/\alpha)$ factors.*

2.4. LWE Diffie-Hellman Key Exchange

This is one of the cryptographic primitives we make use in our construction from the next section. Since our desire is to build a LWE-based protocol, we need a LWE-based version of the famous Diffie-Hellman (DH) key exchange. We recall below the basic Diffie-Hellman protocol.

Diffie-Hellman Key Exchange

Setup: Choose and publish a prime p and α generator of $\mathbb{Z}_p^*$.

Execution:

- A chooses a random integer x, $1 \leq x \leq p-2$ and sends B message
$$A \longrightarrow B : \alpha^x \bmod p,$$
- B chooses a random integer y, $1 \leq y \leq p-2$ and sends A message
$$B \longrightarrow A : \alpha^y \bmod p.$$

Both A and B compute the shared key $K = \alpha^{yx} = \alpha^{xy}$. The security of the protocol relies on the difficulty of the Diffie-Hellman problem and the hardness of computing discrete logarithms.

We present now a LWE version of the basic Diffie-Hellman protocol [4] derived from the LWE problem [11].

LWE Diffie-Hellman Key Exchange

Setup: Choose m, n and q some positive integer numbers, choose a random matrix $A \in \mathbb{Z}_q^{n \times m}$ and make it public.

Execution:

- A chooses a random vector $\mathbf{x} \in \mathbb{Z}_q^n$ and short "error" vector $\mathbf{e} \in \mathbb{Z}_q^m$ and sends B the message
$$A \longrightarrow B : \mathbf{x'} = \mathbf{A^T x} + \mathbf{e} \in \mathbb{Z}_\mathbf{q}^\mathbf{m},$$
- B chooses a random short $\mathbf{y} \in \mathbb{Z}_q^m$ and sends A the message
$$B \longrightarrow A : \mathbf{y'} = \mathbf{A y} \in \mathbb{Z}_\mathbf{q}^\mathbf{n}.$$

At the end of the protocol, both A and B are able to compute a common key based on the values they generated and on the messages received one from another. A calculates $\mathbf{x^T} \cdot \mathbf{y'} = \mathbf{x^T A y}$ while B calculates $\mathbf{x'} \cdot \mathbf{y} = \mathbf{x^T A y} + \mathbf{e y}$ where $\mathbf{e y}$ is "small". By applying the $round(\cdot)$ function, both A and B compute the shared key
$$K = round(\mathbf{x} \cdot \mathbf{y'}) = round(\mathbf{x'} \cdot \mathbf{y}).$$

The $round(\cdot)$ function has a basic variant which is used in [11] for decryption:

$$round(x) = \begin{cases} 1, & x \in [0, \lfloor q/2 \rfloor], \\ 0, & \text{otherwise.} \end{cases}$$

Nevertheless, we prefer to use in our construction the extended variant of the function which rounds to smaller intervals, namely $round(x) = a$ if $x \in \left[a \cdot q/A, (a+1) \cdot q/A\right]$ where A is the total number of intervals.

The security of the LWE Diffie-Hellman key exchange relies on the difficulty of the LWE problem. An eavesdropper of the protocol may see matrix $\mathbf{A}$ and vectors $\mathbf{x}'$, $\mathbf{y}'$ but he cannot recover the secret key derived in the protocol. From $\mathbf{A}$ and $\mathbf{x}'$ he can not recover secret $\mathbf{x}$ since this implies inverting the function $f_{\mathbf{A}}(x) = \mathbf{A}\mathbf{x} \bmod q$ which is hard to invert when $\mathbf{x}$ is short (is equivalent to solving the SIS problem). On the other hand, the LWE function $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = (\mathbf{s}\ \mathbf{A} + \mathbf{e}) \bmod q$ with very short noise $\mathbf{e}$ is hard to invert; therefore, an eavesdropper is not able to recover none of the secret values from this key exchange and thus, neither the computed shared secret key.

Decisional LWE Diffie-Hellman. In the decision version of the LWE Diffie Hellman the goal is to distinguish between keys generated in the Diffie-Hellman key exchange as above and uniformly random values from $\mathbb{Z}_q$. The hardness of this problem follows directly from the hardness of the decisional version of LWE.

2.5. Public key cryptosystem based on LWE

Our key transfer protocol is based on a public key cryptosystem [6] which is proved to be secure based on the hardness of LWE. The cryptosystem is a dual of R e g e v' s cryptosystem [11], the first cryptosystem derived from the hardness of LWE. The original cryptosystem chooses the private key as a uniformly random binary vector $\mathbf{s} \in \{0,1\}^m$ and computes the public key as a set of LWE-samples $\mathbf{A}\mathbf{s} + \mathbf{e}$. In the cryptosystem below, the private key is a vector $\mathbf{e} \in \mathbb{Z}_q^m$ from the Gaussian distribution while the corresponding public key is the syndrome $\mathbf{u} = f_{\mathbf{A}(\mathbf{e})} = \mathbf{A}\mathbf{e} \in \mathbb{Z}_q^n$.

The dual cryptosystem is parameterized by $r > \omega\left(\sqrt{log\ m}\right)$ for some positive integer m which describes the Gaussian distribution $D_{\mathbb{Z}^m,r}$ from which the secret keys are drawn. A matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is public and shared by all users. This matrix describes the function $f_{\mathbf{A}(\mathbf{e})} = \mathbf{A}\mathbf{e} \bmod q$. Note that all operations in the cryptosystem below are performed in $\mathbb{Z}_q$. We present the extended version of the cryptosystem which allows encrypting messages of length k=poly(n) bits.

Public key cryptosystem ([6])

PKE.KeyGen**:** Choose k vectors $\mathbf{e_i} \leftarrow D_{\mathbb{Z}^m,r}$, $1 \le i \le k$ and consider matrix $\mathbf{E} \in \mathbb{Z}_q^{k \times m}$ defined by $\mathbf{E} = \{\mathbf{e_1}, \ldots, \mathbf{e_k}\}$ as the secret key. The public key $\mathbf{U} \in \mathbb{Z}_q^{n \times k}$ consists of k syndromes $\{\mathbf{u_1}, \ldots, \mathbf{u_k}\}$ where $\mathbf{u_i} = f_{\mathbf{A}(\mathbf{e_i})} = \mathbf{A}\mathbf{e_i}$.

PKE.Enc($pk = \mathbf{U}$, **M**): To encrypt message M, choose uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$, compute $\mathbf{p} = \mathbf{A^T s} + \mathbf{x_1} \in \mathbb{Z}_q^m$ where $\mathbf{x_1} \leftarrow \chi^m$. Compute also $\mathbf{c} = \mathbf{U^T s} + \mathbf{x_2} + \mathbf{M} \cdot \lfloor \mathbf{q/2} \rfloor \in \mathbb{Z}_\mathbf{q}^\mathbf{k}$ where $\mathbf{x_2} \leftarrow \chi^k$. Output the ciphertext (**p**,**c**).

PKE.Dec(**E**, (**p**,**c**)): Parse **c** as $[c_1, \ldots, c_k] \in \mathbb{Z}_q^k$. For all $1 \leq j \leq k$ compute $b_j^{'} = c_j - \mathbf{e_j^T p} \in \mathbb{Z}_q$. Compute the bits of the message M as follows: let $b_j = 0$ if $b_j^{'}$ is closer to 0 than to $\lfloor q/2 \rfloor$; otherwise set $b_j = 1$. Output message $M = [b_1, \ldots, b_k]$.

Theorem 2 ([6]). *For the following choosing of parameters $q \geq 5r(m+1)$, $\alpha \leq \left(1/r\sqrt{m+1} \cdot \omega(\sqrt{log\ n})\right)$ and χ a Gaussian distribution on $\mathbb{Z}_q$, the above cryptosystem is CPA-secure and anonymous, assuming the $LWE_{q,\chi}$ problem is hard. Anonymous cryptosystem means here that a ciphertext hides the identity to which it was encrypted.*

2.6. Anonymous key transfer protocol

In the next section we provide a key transfer protocol with anonymity of the participants against each other, so except the manager who knows the composition of the team, no other expert in the database (even if he is a team member) knows anything about all the other experts.

We present an appropriate security model for our anonymous key transfer protocol. We define security in our group key transfer using the *adaptive* security notion [7] used in broadcast encryption systems. In an adaptively secure system, the adversary is allowed to see the public keys and then require the corresponding secret keys of the set of identities he wishes to attack.

Definition 1. We define the ANO-IND-CPA security game (against adaptive adversaries) for our protocol as follows.

- **Setup.** The challenger runs the Setup to generate the public key of the manager MPK and the corresponding private key MSK and gives MPK to the adversary.
- **Phase 1.** $\mathcal{A}$ can issues two types of queries:
  - private key extraction queries to an oracle for any index $i \in U$; the oracle will respond by returning the private key sk_i corresponding to i;
  - Diffie Hellman key extraction queries to an oracle for any index $i \in U$; the oracle will respond by returning the sk_i^{DH} Diffie Hellman secret key shared by the manager and expert i;
- **Challenge.** The adversary selects two equal length messages m_0 and m_1 and two distinct sets S_0 and $S_1 \subseteq U$ of users. We impose the same requirements as in [8]: sets S_0 and S_1 should be of equal size and A has not issued any query to any $i \in (S_0 \setminus S_1) \cup (S_1 \setminus S_0)$.

Further, if there exists an $i \in S_0 \cap S_1$ for which A has issued a query, then we require that $m_0 = m_1$. The adversary gives m_0, m_1 and S_0, S_1 to the challenger. The latter picks a random bit $b \in \{0,1\}$, computes $c^* = Enc(m_b, S_b)$ and returns it to A.

- **Phase 2.** $\mathcal{A}$ continues to issue private key extraction queries and Diffie Hellman key extraction queries with the restriction that $i \notin (S_0 \setminus S_1) \cup (S_1 \setminus S_0)$; otherwise it is necessary that $m_0 = m_1$.
- **Guess.** The adversary A outputs a guess $b' \in \{0,1\}$ and wins the game if $b = b'$.

We denote $\mathcal{A}'s$ advantage by $Adv_{A,KT}^{ANO-IND-CPA}(\lambda) = |Pr[b' = b] - \frac{1}{2}|$, where λ is the security parameter of the protocol.

Definition. We say that a key transfer protocol is anonymous and semantically secure against chosen plaintext attacks (ANO-IND-CPA) if all polynomial-time adaptive adversaries A have at most negligible advantage in the above game.

3. Anonymous LWE-based key transfer protocol

3.1. Our construction

Denote by $\mathcal{B} = \{P_1, \ldots, P_n\}$ the database of experts from which the manager M selects his team. Every P_i is identified by a public id_i.

Our protocol is parameterized by three positive integers m, n and q. A key generation center generates a public matrix $M \in \mathbb{Z}_q^{n \times m}$ which can be used by anyone in the protocol. Let t be the maximum size of the team chosen by M. Let $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ a lattice-based signature scheme. We won't present a particular scheme here, since it is not important for our protocol, but we point, for example, to the scheme from [9]. $\mathcal{H}$ is a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^m$.

Before presenting our construction, let us first review the main steps of the protocol. The first step is just a LWE Diffie-Hellman key exchange (as presented in 2.4) that the manager makes with each participant, being at the same time a request for participation. In the next steps, after building a team, the manager sends to every team member the secret key encrypted using the cryptosystem from Subsection 2.5.

The secret keys exchanged in the previous step are used to achieve anonymity of the team members against each other. In the last step, the team members are able to recover the secret key they share with the manager without any of them knowing the structure of the team.

Setup the team

(1) M exchanges with every expert $P_i \in \mathcal{B}$ a LWE Diffie-Hellman key:
   (a) M generates random $\mathbf{a_i} \in \mathbb{Z}_q^n$, random short $\mathbf{e_i} \in \mathbb{Z}_q^n$ and signature key pair $(SK_M, VK_M) \leftarrow \mathcal{G}(\lambda)$ and sends the following message to P_i

$$M \longrightarrow P_i : \left\{\mathbf{a'} = \mathbf{M^T} \cdot \mathbf{a_i} + \mathbf{e_i}, \sigma, VK_M\right\} \qquad \text{for every} \quad 1 \leq i \leq n,$$

   where $\sigma = \mathcal{S}(SK_M, \mathbf{a'})$. This step represents the request for participation addressed to all the experts in the database.
   (b) P_i first verifies the validity of the signature and if decides to accept the request, then he generates a random short $\mathbf{b_i} \in \mathbb{Z}_q^m$, public $\mathbf{v_i} \in \mathbb{Z}_q^{n \times t}$, signature key pair $(SK_i, VK_i) \leftarrow \mathcal{G}(\lambda)$ and runs the algorithm $\mathsf{TrapGen}(n)$ to generate a matrix $\mathbf{A_i} \in \mathbb{Z}_q^{n \times m}$ (public key) together with a short basis T_i for $\Lambda^{\perp}(\mathbf{A_i})$ (which he will use in a further step) and sends back to M the message

$$P_i \longrightarrow M : \left\{\mathbf{b'} = \mathbf{M} \cdot \mathbf{b_i}, \sigma, VK_i\right\},$$

   where $\sigma = \mathcal{S}(SK_i, \mathbf{b'})$. At the end of this step, M shares with every P_i a secret key $k_i = round(\mathbf{a_i} \cdot \mathbf{b'}) = round(\mathbf{a'} \cdot \mathbf{b_i})$.

(2) M selects a team of experts $T = \{P_{i_1}, \ldots, P_{i_k}\}$, $M \in T$ from the experts who accepted the request. Let $I = \{i_1, \ldots, i_k\}$.

Note that from now on, we will refer to any letter which has index i_j by index j. So, for the sake of simplicity of notation, we will use P_j instead of P_{ij} and so on.

Transfer session key

M computes the team session key

$$K = \sum_{j \in I} k_j \pmod q \tag{1}$$

and publishes it as follows:

- M computes for every selected member P_j the pair $C_j = (l_j, c_j) = \mathsf{PKE.Enc}(\mathbf{v_j}, K)$ and the value $H_j = \mathcal{H}(k_j)$,
- M chooses a random permutation $\pi : \{1, \ldots, k\} \to \{1, \ldots, k\}$ and sets the public ciphertext as

$$C = \left\{VK_M, (H_{\pi(1)}, C_{\pi(1)}), \ldots, (H_{\pi(k)}, C_{\pi(k)}), \sigma\right\},$$

  where $\sigma = \mathcal{S}(SK_M, C)$.

(3) Every P_i: checks that the signature on the ciphertext is valid; if the verification fails, he aborts; otherwise, he computes $H = \mathcal{H}(k_i)$ and if $H \neq H_j$ for every $j \in \{1, \ldots, k\}$ then he was not selected.

Otherwise, for every $1 \leq d \leq t$ he generates

$$\mathbf{e_{jd}} = \mathsf{SamplePre}\big(\mathbf{A_j}, T_j, \mathbf{v_d}, r(k+1)\big),$$

denotes by $\mathbf{e} = [\mathbf{e_{j_1}}, \ldots, \mathbf{e_{j_d}}]$ and recovers message M = PKE.Dec($\mathbf{e}$,C_j) representing the shared secret key.

3.2. Correctness

The correctness of our scheme is ensured by the LWE—cryptosystem [6] and the properties of the trapdoor function from the same paper. In the original cryptosystem, the public key is the one-way function $f_{\mathbf{A}} : D_n \leftarrow \mathbb{Z}_q^n$ with $D_n = \{\mathbf{e} \in \mathbb{Z}^m : ||\mathbf{e}|| \leq s\sqrt{m}\}$, $f_{\mathbf{A}}(\mathbf{e}) = \mathbf{Ae} \bmod q$ applied to an error vector $\mathbf{e}$ chosen as the secret key.

In our construction, in the encryption process the syndrome $\mathbf{v_j}$ is randomly chosen and then $\mathbf{e_j} \in D_T = \{\mathbf{e} \in \mathbb{Z}^m : ||\mathbf{e}|| \leq r(k+1)\}$ is sampled using the algorithm $\mathsf{SamplePre}\big((\mathbf{A_j}, T_j, \mathbf{v_d}, r(k+1)\big)$. The distribution under which the latter algorithm samples is within negligible statistical distance of $D_{\Lambda_{v_j}^{\perp}(A),r(k+1)}$, as stated in [5], which is the conditional distribution $D_{Z_m,r}$ conditioned on $\mathbf{Ae} = y \bmod q$. Lemma 5.2 from [6] states that the distribution of the syndrome $v_j = \mathbf{A_j e_j} \bmod q$ is within negligible distance of uniform over $\mathbb{Z}_q^n$, so we make the right choice of $\mathbf{v_j}$ in our construction.

3.3. Security

Theorem 3. *The group key transfer protocol we built is adaptively ANO-IND--CPA secure assuming that the underlying public key cryptosystem (Section 2.5) is CPA-secure and anonymous, the $\mathcal{H}$ hash function is collision resistant and Σ is a strongly unforgeable signature.*

P r o o f. According to the ANO-IND-CPA security game, the two challenged sets S_0 and S_1 must have equal size, that is $|S_0| = |S_1| = l$. In this proof we consider a sequence of games, where, in the first game, the adversary is given an encryption of M_0 for S_0 and he ends by obtaining, in the last game, an encryption of M_1 for S_1. Note that we adapt the security proof of [8, Theorem 4] to our case.

Game 0_{real}**:** corresponds to the real experiment when the challenger chooses bit $b = 0$. In the first phase, the adversary adaptively chooses indices of the users from set $\{1, \ldots, n\}$ for whom he wants to obtain the corresponding secret keys sk_i, knowing the public keys pk_i. He also makes Diffie-Hellman queries to the challenger in order to obtain private keys sk_i^{DH} corresponding to user indexed by i with $i \in \{1, \ldots, n\}$. In the challenge phase, adversary $\mathcal{A}$ chooses messages M_0 and M_1 and two sets $S_0, S_1 \subset \{1, \ldots, n\}$ of size $|S_0| = |S_1| = l$ with $S_0 \neq S_1$. The challenger $\mathcal{C}$ generates a signature key pair (SK^*, VK^*), parses $S_0 = \{\alpha_1, \ldots, \alpha_l\}$, computes $H_i = \mathcal{H}(k_{\alpha_i})$ for all $1 \leq i \leq l$ where k_{α_i} is the DH key that M shares with user P_i. Finally, the challenger returns the challenge ciphertext

$C^* = \big(VK^*, (H_{\pi(1)}, C_{\pi(1)}), \ldots, (H_{\pi(l)}, C_{\pi(l)}), \sigma\big)$ where $\pi : \{1, \ldots, l\} \to \{1, \ldots, l\}$ is a random permutation.

At the end, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$ and we denote by E_0^{real} to be the event that $b' = 0$.

Game 0**:** is identical to Game 0_{real} except that the challenger now rejects all post challenge Diffie Hellman key extraction queries for user indexed by i such that $\mathcal{H}(sk_i^{DH}) = \mathcal{H}(sk_j^{DH})$ with $j \neq i$ where $H_j = \mathcal{H}(sk_j^{DH})$ appeared before in the challenge phase. We denote by E_0 the event that $\mathcal{A}$ outputs $b' = 0$ in Game 0.

Game 0′**:** is just identical to Game 0.

Game $k (1 \leq k \leq l)$**:** As in the first game, $\mathcal{A}$ selects two equal-length messages M_0, M_1 and two distinct sets S_0, $S_1 \subset \{1, \ldots, n\}$ and passes them to $\mathcal{C}$. $\mathcal{C}$ defines the value $i = |S_0 \cap S_1|$ and re-orders the two received sets in the following way

$$S_0^{'} = \{\alpha_1, \ldots \alpha_i, \alpha_{i+1}, \ldots, \alpha_l\}$$

and

$$S_1^{'} = \{\beta_1, \ldots \beta_i, \beta_{i+1}, \ldots, \beta_l\}$$

such that $\alpha_j = \beta_j$ for each $j \in \{1, \ldots, i\}$ and $\alpha_j \neq \beta_j$ if $j \in \{i + 1, \ldots, l\}$. If one of the α_j or β_j with $j \in \{1, \ldots, l\}$ is contained in the list of $\mathcal{H}$ (assume that all the relevant queries to $\mathcal{H}$ have been asked before), $\mathcal{C}$ aborts and returns a random bit. Then $\mathcal{C}$ generates the challenge ciphertext as follows:

- For $j = 1$ to i
  (1) Compute $H_j = \mathcal{H}(sk_{\alpha_j}^{DH})$;
  (2) set $C_j = \mathsf{PKE.Encrypt}(pk_{\alpha_j}, M_1 || VK^*)$ if $j \leq k$ and $C_j = \mathsf{PKE.Encrypt}(pk_{\alpha_j}, M_0 || VK^*)$ if $j > k$.
- For $j = i + 1$ to l
  - if $j < k$, compute $H_j = \mathcal{H}(sk_{\beta_j}^{DH})$ and $C_j = \mathsf{PKE.Encrypt}(pk_{\beta_j}, M_1 || VK^*)$;
  - if $j > k$, compute $H_j = \mathcal{H}(sk_{\alpha_j}^{DH})$ and $C_j = \mathsf{PKE.Encrypt}(pk_{\alpha_j}, M_0 || VK^*)$;
  - if $j = k$, compute $H_k = \mathcal{H}(sk_{\beta_k}^{DH})$ and $C_k = \mathsf{PKE.Encrypt}(pk_{\alpha_k}, M_0 || VK^*)$.

The adversary is then returned

$$C^* = \big(VK^*, (H_{\pi(1)}, C_{\pi(1)}), \ldots, (H_{\pi(l)}, C_{\pi(l)})\big),$$

where $\pi : \{1, \ldots, l\} \to \{1, \ldots, l\}$ is a random permutation. We denote by E_k the event of $\mathcal{A}$ outputting $b' = 0$ at the end of Game k.

Game $k'(1 \leq k \leq l)$**:** is almost identical to Game k except the following modification:

For $j = i+1$ to l, if $j = k$ compute $H_k = \mathcal{H}\big(sk_{\beta_k}^{DH}\big)$ and $C_k = \mathsf{PKE.Encrypt}(pk_{\beta_k}, M_1||VK^*)$.

We denote by E'_k the event that $\mathcal{A}$ outputs bit $b' = 0$ at the end of this game.

Game l_{real}**:** corresponds to the real game when the challenger's bit is $b' = 1$. We denote by E_l^{real} the event that $\mathcal{A}$ outputs bit $b' = 0$ at the end of this game.

We remark that Game 0_{real} and Game 0 are indistinguishable if the hash function $\mathcal{H}$ is collision resistant. Note that we can reason the same way concerning Game l and Game l_{real}.

Therefore, we can state that $|Pr[E_0^{real}] - Pr[E_0]| = |Pr[E_l^{real}] - Pr[E_l]|$ is negligible. For the other transitions, we can also show in the below lemma that they can not be distinguished. First, note that for $k \in \{1, \ldots, i\}$, Game k and Game k' cannot be distinguished since they are identical. Regarding these games, we only have to show indistinguishability for $k \in \{i+1, \ldots, l\}$ which follows from the below lemma.

In Lemma 2, we show also that Game k and Game $k-1'$ are indistinguishable under the IND-CPA security of the LWE encryption scheme. This concludes our proof, since we showed that an IND-CPA adversary of the encryption scheme is not able to distinguish the above games. □

Lemma 1. *For each $k \in \{i+1, \ldots, l\}$, Game k' is indistinguishable from Game k if the underlying cryptosystem from LWE problem is IND-CPA secure.*

P r o o f. We prove that if the adversary $\mathcal{A}$ can distinguish Game k and Game k', there is a chosen plaintext adversary against the LWE encryption scheme. Note that for each $k \in \{i+1, \ldots, l\}$, Game k and Game k' are identical when $M_0 = M_1$ and thus, we assume $M_0 \neq M_1$.

The IND-CPA adversary $\mathcal{P}$ receives a public key pk^* from its challenger and prepares n pairs of public/private keys for the adversary $\mathcal{A}$ of the protocol in the following way: he picks i^* at random from $\{1, \ldots, n\}$ and defines $pk_{i^*} = pk^*$; then, he generates $n-1$ key encryption pairs $(pk_i, sk_i) \leftarrow \mathsf{PKE.Keygen}$ for each $i \in \{1, \ldots, n\} \setminus \{i^*\}$ and gives to $\mathcal{A}$ all the public keys $\{pk_i\}_{i=1}^n$. $\mathcal{A}$ is allowed to make Diffie-Hellman queries to the challenger in order to obtain private keys sk_i^{DH} corresponding to user indexed by i with $i \in \{1, \ldots, n\}$.

Adversary $\mathcal{P}$ is now able to answer all the corruption queries from $\mathcal{A}$ since he knows the secret keys sk_i. Anyway, $\mathcal{P}$ aborts and fails if $\mathcal{A}$ chooses to corrupt user i^*.

The challenge phase goes like this: $\mathcal{A}$ outputs messages M_0, M_1 and two subsets S_0, $S_1 \subset \{1,\ldots,n\}$ of equal size. $\mathcal{P}$ orders the sets as $S_0' = \{\alpha_1,\ldots,\alpha_i, \alpha_{i+1},\ldots,\alpha_l\}$, $S_1' = \{\beta_1,\ldots,\beta_i,\beta_{i+1},\ldots,\beta_l\}$ with $\alpha_j = \beta_j$ for each $j \in \{1,\ldots,i\}$. In the situation when $\alpha_k \neq i^*$, $\mathcal{P}$ aborts and fails. We denote by Good the event that $\alpha_k = i^*$.

When Good occurs, adversary $\mathcal{P}$ chooses a signature key pair $(SK^*, VK^*) \leftarrow \mathcal{G}(\lambda)$ and send to his IND-CPA challenger the two messages $(M_0||VK^*)$, $(M_1||VK^*)$. The latter chooses a random bit b and replies with the challenge ciphertext $C^* = \mathsf{PKE.Encrypt}(\mathsf{pk}^*, \mathsf{M_b}||\mathsf{VK}^*)$. The anonymous challenge ciphertext is defined as follows.

(1) For $j = 1$ to $k-1$, $\mathcal{P}$ computes $H_j = \mathcal{H}\big(sk_{\alpha_j}^{DH}\big)$ and sets $C_j = \mathsf{PKE.Encrypt}(pk_{\alpha_j}, M_1||VK^*)$.

(2) For $j = k+1$ to l, $\mathcal{P}$ computes $H_j = \mathcal{H}\big(sk_{\alpha_j}^{DH}\big)$ and sets $C_j = \mathsf{PKE.Encrypt}(pk_{\alpha_j}, M_0||VK^*)$.

(3) For $j = k$, $\mathcal{P}$ computes $H_k = \mathcal{H}(sk_{\alpha_k}^{DH})$ and sets $C_k = C^*$.

Adversary $\mathcal{A}$ is then returned the ciphertext $C = \big(VK^*, (H_{\pi(1)}, C_{\pi(1)}), \ldots \ldots, (H_{\pi(l)}, C_{\pi(l)})\big)$ where $\pi : \{1,\ldots,l\} \to \{1,\ldots,l\}$ is a random permutation.

At the end of the game, adversary $\mathcal{A}$ is able to output bit b' and $\mathcal{P}$ produces the same result. If $\mathcal{P}$ did not abort, its IND-CPA advantage is equal to the difference between $\mathcal{A}$'s probability of outputting 0 in Game k and Game k'. According to the games k and k' described above, if the challenger of $\mathcal{P}$ chooses $b = 0$, then $\mathcal{P}$ is in Game k, otherwise he is in Game k'.

It remains only to evaluate $\mathcal{P}$'s probability not to abort. Note that, from the ANO-IND-CPA game, we have that whenever $M_0 \neq M_1$, $\mathcal{A}$ is not allowed to corrupt (i.e., ask for the secret keys) any user in $S_0 \cap S_1 = \{\alpha_1,\ldots,\alpha_i\}$. Since $\alpha_k \notin S_0 \cap S_1$, we only need to be assured that $\alpha_k = i^*$, which means that Good happens. Note that $Pr[\mathsf{Good}] = 1/n$ since index i^* is chosen at random. □

Lemma 2. *For each $k \in \{1,...,l\}$, Game k is indistinguishable from Game $k-1'$ if the underlying cryptosystem from LWE problem is IND-CPA secure.*

Proof. We can show below, by simple checking, that Game k and Game $k-1'$ are indistinguishable.

(1) For $j = 1$ to i

In both games, Game $k-1'$ and Game k, H_j is computed in the same way $H_j = \mathcal{H}\big(sk_{\alpha_j}^{DH}\big)$;

As for C_j, in both games, $C_j = \mathsf{PKE.Encrypt}(pk_{\alpha_j}, M_1||VK^*)$, for $j \in \{1,\ldots,k-1\}$ and $C_j = \mathsf{PKE.Encrypt}(pk_{\alpha_j}, M_0||VK^*)$, for $j \in \{k+1,\ldots,i\}$. The only difference in these games is for $j = k$: in Game $k-1'$,

we set $C_k = \mathsf{PKE.Encrypt}(pk_{\alpha_k}, M_0||VK^*)$ while in Game k we set $C_k = \mathsf{PKE.Encrypt}(pk_{\alpha_k}, M_1||VK^*)$.

But note that an IND-CPA adversary LWE encryption scheme is not able to distinguish the two encrypted messages C_k with non-negligible probability. This property comes from the definition of IND-CPA security when the two messages M_0 and M_1 have equal length (this is what we have in our hypothesis also).

(2) For $j = i + 1$ to l

It is easy to see that the pair (H_j, C_j) is computed exactly in the same way in both Game k and Game $k - 1'$.

In both games k and $k - 1'$, $H_j = \mathcal{H}\big(sk_{\alpha_j}^{DH}\big)$,

$$C_j = \mathsf{PKE.Encrypt}(pk_{\alpha_j}, M_1||VK^*),$$

for $j \in \{i + 1, \ldots, k - 1\}$ and $H_j = \mathcal{H}\big(sk_{\beta_j}^{DH}\big)$

$$C_j = \mathsf{PKE.Encrypt}(pk_{\beta_j}, M_0||VK^*),$$

for $j \in \{k + 1, \ldots, i\}$.

When $j = k$, in both games $H_j = \mathcal{H}\big(sk_{\beta_k}^{DH}\big)$ and

$$C_k = \mathsf{PKE.Encrypt}(pk_{\beta_k}, M_0||VK^*).$$

□

4. Efficiency

In the protocol from section 3, we manage to achieve the property of anonymity with two benefits: we obtain constant decryption time, size of the ciphertext being linear in the size of the selected team. Regarding the number of rounds, our protocol is not efficient. The Diffie-Hellman exchange increases the number of rounds, but also provides anonymity, so we accept this trade-off.

We remark that our construction is not very practical, but it can be seen as a starting point for an anonymous group key transfer protocol secure in the lattice-based environment.

5. Conclusions

We introduced in this paper the first lattice-based group key transfer protocol achieving anonymity via a lattice-based Diffie-Hellman key exchange. We managed to obtain reduced size of ciphertext (linear in $|T|$) and constant decryption time. Instead, we added some cost to our protocol since we increased the number of encryptions.

We believe that work is still to be done in this protocol, for improving its efficiency. We need to reduce somehow the number of encryptions without loosing anonymity. In the current protocol, we managed to achieve anonymity only at this cost, but we believe that it can be done at a lower cost, maybe without the Diffie-Hellman key exchange protocol as a first part of our protocol.

REFERENCES

[1] ATANASIU, A.—MIHAITA, A.: *A key agreement protocol based on Identity-Based Proxy Re-encryption*, in: Proc. of the Internat. Conf. on Security and Management—SAM '11, Las Vegas, USA, 2011.

[2] ATANASIU, A.—GEORGESCU, A.: *A secure authenticated group key transfer protocol*, in: Proc. of the 7th South East European Doctoral Student Conf.—DSC '12, Thessaloniki, Greece, 2012.

[3] AJTAI, M.: *Generating hard instances of the short basis problem*, in: 26th Internat. Colloq.—ICALP '99 (J. Wiedermann et al., eds.), Prague, Lecture Notes in Comput. Sci., Vol. 1644, Springer, Berlin, 1999, pp. 1–9.

[4] BONEH, D.: *Recent developments in cryptography: lattices and beyond*, http://forum.stanford.edu/events/2010slides/security/DanBonehSecurity2010.pdf

[5] CASH, D.—HOFHEINZ, D.—KLITZ, E.: *How to delegate a lattice basis*, IACR Cryptology ePrint Archive 2009/351, 2009.

[6] GENTRY, C.—PEIKERT, C.—VAIKUNTANATHAN, V.: *Trapdoors for hard lattices and new cryptographic constructions,* in: Proc. of the 40th Annual ACM Symposium on Theory of Computing—STOC '08, Victoria, Canada, 2008, ACM, New York, NY, pp. 197–206.

[7] GENTRY, C.—WATERS, B.: *Adaptive security in broadcast encryption systems*, in: Advances in Cryptology—EUROCRYPT '09, 28th Annual Internat. Conf. on the Theory and Appl. of Cryptographic Techniques (A. Joux, ed.), Cologne, Germany, 2009, Lecture Notes in Comput. Sci., Vol. 5479, Springer, Berlin, 2009, pp. 171–188.

[8] LIBERT, B.—PATERSON, K. G.—QUAGLIA, E. A.: *Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model*, in: Public Key Cryptography—PKC '12, 15th Internat. Conf. on Practice and Theory in Public Key Cryptography (M. Fischlin et al., eds.), Darmstadt, Germany, 2012, Lecture Notes in Comput. Sci., Vol. 7293, Springer, Berlin, 2012, pp. 206–224.

[9] LYUBASHESKY, V.: *Lattice signatures without trapdoors*, in: Advances in Cryptology––EUROCRYPT '12 31st Annual Internat. Conf. on the Theory and Applications of Cryptographic Techniques (D. Pointcheval et al., eds.), Cambridge, UK, 2012, Lecture Notes in Comput. Sci., Vol. 7237, Springer, Berlin, 2012, pp. 738–755.

[10] MICCIANCIO, D.—GOLDWASSER, S.: *Complexity of Lattice Problems: a cryptographic perspective*, in: The Kluwer Internat. Ser. in Engineering and Comput. Sci., Vol. 671, Kluwer Academic Publishers, Boston, MA, 2002.

[11] REGEV, O.: *On lattices, learning with errors, random linear codes, and cryptography*, in: Proc. of the 37th Annual ACM Symposium on Theory of Computing—STOC '05 (H. N. Gabow and R. Fagin, eds.), Baltimore, MD, USA, 2005, ACM, New York, 2005, pp. 84–93.

[12] WANG, J.—BI, J.: *Lattice-based Identity-Based Broadcast Encryption*, ePrint IACR, 2010, http://eprint.iacr.org/2010/288.pdf.

Received October 30, 2012

Department of Computer Science
Faculty of Mathematics and Computer Science
University of Bucharest
Academiei Street no. 14, sector 1
Bucharest
ROMANIA
E-mail: adela@fmi.unibuc.ro