A LWE-based Secret Sharing Scheme

Adela Georgescu
Faculty of Mathematics and
Computer Science, University
of Bucharest
Academiei Street 14,
Bucharest 010014

ABSTRACT

We present a (n,n) secret sharing scheme whose security can be reduced to the hardness of the Learning With Errors (LWE) problem. This is a strong property since the LWE problem is believed to be very hard, as hard as worst-case lattice problems hence offering security in the quantum world. The scheme has certain technical advantages: it requires only basic operations and it allows sharing several secrets at the same time.

Keywords:

Secret sharing, learning with errors, lattices

1. INTRODUCTION

A secret sharing scheme is a cryptographic primitive that allows a secret to be shared among a set of participants such that only a qualified subset (or even the whole set) can recover the secret. Secret sharing schemes were independently introduced both by Shamir and Blakley, in 1979, with the scope of safeguarding encryption keys. Today, their use is extended to access control systems, e-voting, authentication protocols, etc.

The original secret sharing scheme of Shamir [9] was based on polynomial interpolation while Blakley [2] based his scheme on intersection of affine hyperplanes. Among other improvements that those schemes miss we mention the imposibility to verify whether the shares of the participants are valid. Later secret sharing schemes are based on Chinese Remainder Theorem ([1], [6]), on Information Dispersal [5] etc.

A (t,n) threshold secret sharing scheme $(t \le n)$ requires the presence of at least t parties (up to n) with their shares to recover the secret, while for any subset of $t-1$ or less parties it is impossible to recover the secret.

In this paper, we propose a threshold secret sharing scheme with $t = n$ based on the LWE problem. The (n,n) threshold schemes are called unanimous consent schemes in the literature. Our scheme has a very simple construction, uses basic operations like addition modulo q and multiplication, it is verifiable - so that malicious parties trying to insert fake shares fail - and ideal - the size of one share is the size of the secret.

The rest of the paper is organized as follows: in Section 2 we define the LWE problem, Section 3 presents our proposed secret sharing scheme based on LWE and some conclusions in Section 4 .

2. THE LEARNING WITH ERROR PROBLEM

The Learning With Error problem was introduced in 2005 by Regev [8] and it is very famous since it is as hard as worst-case lattice problems. In fact, it is a generalization of the well-known *learning parity with noise - LPN* problem from learning theory.

The LWE is becoming wide used because of being very versatile and thus offering a good basis for cryptographic primitives. Even if it has a simple description, there is no known efficient solution for this problem. The best known algorithm for finding the secret takes exponential time 2^n. So the problem is believed to be hard. There is also another important reason for the claimed hardness of the problem: for certain choices of the parameters, a solution to LWE implies a quantum solution to worst-case lattice problems. This is a strong result since there are no known quantum algorithms for lattice problems that perform significantly better than known classical algorithms.

Another reason for LWE having many applications in cryptography is that the problem has two variants [9]: one of them is the search to decision variant which requires to distinguish LWE samples from samples coming from the uniform distribution; the second variant is the one we already presented above, which requires calculating the secret s and which can be solved if the decision variant can be solved over a uniform choice of the secret.

We enumerate some application of the LWE problem in cryptography: public-key encryption [8], CCA-Secure PKE, identity-based encryption [4], oblivious transfer [7], hierarchical identity-based encryption [3].

LWE problem [9]. The problem asks to recover a secret $s \in Z_q^n$ given an arbitrary number of random linear equations on s, each correct up to an additive error. In the absence of the error, the problem would be very easy: after n equations, we could recover s using Gaussian elimination, in polynomial time. But introducing the error seems to make the problem significantly more difficult.

Let us describe the LWE problem more precisely: fix the size $n \ge 1$, a modulus $q \ge 2$ and an error probability distribution $\chi \in Z_q$. Let $A_{s,\chi}$ defined over $Z_q^n \times Z_q$ be the probability distribution obtained by choosing a vector a uniformly at random, choosing the error $e \in Z_q$ according to χ and outputing these kind of pairs $(a, a*s+e)$ where additions are performed in Z_q. An algorithm solves the LWE problem with modulus q and probability distribution χ if for every $s \in Z_q^n$, given an arbitrary number of samples $A_{s,\chi}$ it outputs s with high probability. The special case $q = 2$ represents a well-known problem from learning theory: LPN - learning parity

with noise, an intensively studied problem in learning theory also believed to be hard.

3. OUR SECRET SHARING SCHEME

3.1 Initialization phase

Denote the participants of the scheme by P_i, $1 \le i \le n$ and the dealer of the scheme by D. His role is to compute the shares of the secret s and distribute them to all the participants in the scheme. Our scheme is a (n,n) threshold scheme, so in order to recover the secret, all the parties have to participate with their shares.

The dealer fixes a modulus q, a public prime number p such that the discrete logarithm problem is intractable in $GF(p)$, a public generator g of the cyclic group $GF(p)$, a size m (note that m, the size of the LWE problem is different from the size n of the secret sharing scheme) and an error probability distribution $\chi \in Z_q$; then he chooses a secret $s \in Z_q^m$ for which he will the compute the n shares. We stress that addition is performed modulo q, exactly like in the LWE problem, but in the following we won't use any special notation for this.

The parameters of our scheme are the same parameters from the LWE problem, so we adhere to suggestion in [9]: the error probability distribution χ is the normal distribution rounded to the nearest integer with standard deviation αq where $\alpha > 0$ is usually taken to be $1/poly(m)$ and the modulus q is chosen to be polynomial in m.

3.2 Construction phase

In order to compute the shares for the n participants, the dealer proceeds in the following manner: for every $1 \le i \le n-1$ he

1. chooses a vector a_i uniform at random from Z_q^m and $e_i \in Z_q$ according to the error probability distribution χ.
2. computes the pair $S_i = (a_i, b_i) = (a_i, a_i s + e_i)$ (addition is performed in Z_q) representing the i-th share corresponding to P_i.
3. computes and publishes $V_i = g^{(a_i, b_i)} = (g^{a_i}, g^{b_i})$.

For the n-th share (the last one), he chooses a_n uniform at random from Z_q^m but this time the additive error is computed from the previous ones: $e_n = (-e_1) + (-e_2) + \cdots + (-e_{n-1})$. So, the last share is the pair $S_n = (a_n, b_n) = (a_n, a_n s + (-e_1 - e_2 - \ldots - e_{n-1}))$.

Then, he computes $V_n = g^{S_n} = (g^{a_n}, g^{b_n})$ Eventually, the dealer publishes the n shares consisting of the following pairs:

$$S_1 = (a_1, a_1 s + e_1)$$
$$S_2 = (a_2, a_2 s + e_2)$$
$$\vdots$$
$$S_{n-1} = (a_{n-1}, a_{n-1} s + e_{n-1})$$
$$S_n = (a_n, a_n s + (-e_1 - e_2 - \ldots - e_{n-1}))$$

3.3 Verification and recovery phase

To recover the secret, all the shares are needed. The recovery with verification is made as follows:

1. Each P_i, $1 \le i \le n$ can verify whether the *j*th share S_j is valid by computing g^{S_j} and comparing it against V_j.
2. If all the shares are valid, they can be used to recover the secret by summing them all

$$S_1 + \cdots + S_n = \left(\sum_{i=1}^{n} a_i, \sum_{i=1}^{n} a_i s \right)$$

Since every participant P_i knows a_i from the his share, it is trivial to compute the secret s (using Gaussian elimination, for example).

Notice that if any team of $n-1$ or less participants try to recover the secret, they will have to solve the LWE problem which is believed to be difficult. In the following, we detail two separate cases concerning the $n-1$ participants trying to recover the secret:

1. Suppose that the $n-1$ participants are exactly those who hold the first $n-1$ shares $S_1, ..., S_{n-1}$. In order to recover the secret s, they have to solve an instance of the LWE problem which is believed to be difficult no matter how many pairs of the type $(a_i, a_i s + e_i)$ are given.
2. Suppose now, in constrast to the first situation, that P_n, holding the n-th share, belongs to group of $n-1$ participants. Without loosing generality, suppose that the following $n-1$ shares are used: $S_1, S_2, ..., S_{n-2}, S_n$. The secret still can not be recovered. Summing the shares $S_1 + ... + S_{n-2} + S_n$ doesn't cancel the error, but just leads to calculating $(a_1 + ... + a_{n-2} + a_n)s - e_{n-1}$. Anyway, this is not helpful, since the error can not be canceled by any means. Summing all the n shares is the only case when the error is canceled. So, even in this situation, the $n-1$ participants have to solve the LWE problem in order to recover the secret s.

4. CONCLUSIONS AND FUTURE WORK

We constructed a simple, efficient unanimous consent secret sharing scheme based on the famous Learning With Errors Problem. We showed that the scheme doesn't allow recovering the secret if at least one participant is missing and this situation reduces to the LWE problem which is believed to be hard. The scheme offers the possibility for the participants to check if all the shares distributed by the dealer are valid.

There is still a lot of work to be done in order to improve the capabilities of the scheme: it would be good to find a (t,n)

variant of the scheme with $t \neq n$ and a way to make it multisecret (to allow sharing severeal secrets instead of one secret shared on each round).

5. REFERENCES

[1] C. A. Asmuth and J. Bloom. A modular approach to key safeguarding. IEEE Transactions on Information Theory, IT-29(2):208–210, 1983.

[2] G. Blakley. Safeguarding cryptographic keys. In Proc. AFIPS 1979 Natl. Computer Conf., N.Y., vol. 48, pp. 313-317, 1979.

[3] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In EUROCRYPT 2010.

[4] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Proc. 40th ACM Symp. on Theory of Computing (STOC), pages 197–206. 2008.

[5] H. Krawczyk. Secret sharing made short. In D. R. Stinson, editor, Advances in Cryptology - CRYPTO '93, volume 773 of Lecture Notes in Computer Science, pages 136–146. Springer-Verlag, 1994.

[6] M. Mignotte. How to share a secret. In T. Beth, editor, Cryptography- Proceedings of the Workshop on Cryptography, Burg Feuerstein, 1982, volume 149 of Lecture Notes in Computer Science, pages 371–375. Springer-Verlag, 1983.

[7] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In CRYPTO, pages 554–571. 2008.

[8] O. Regev. On lattices, learning with errors, randomlinear codes, and cryptography. Journal of the ACM, 56(6):34, 2009. Preliminary version in STOC'05.

[9] O. Regev. The Learning With Errors Problem, Invited survey in CCC 2010.

[10] A. Shamir. How to share a secret. Communications of the ACM 22 (1979) 612-