Development of a Lattice-based Cryptosystem

S. I. Anyanwu
Fed. Univ. of Tech., Akure,
Dept. of Computer Science,
Ondo State.

B. K. Alese
Fed. Univ. of Tech., Akure,
Dept. of Computer Science,
Ondo State.

O. O. Obe
Fed. Univ. of Tech., Akure,
Dept. of Computer Science,
Ondo State.

ABSTRACT

This work proposes a lattice-based cryptosystem using embedded technique of the closest vector problem (CVP). It adopts the key-gen algorithms of [1], and improves on vector reduction method for encryption/decryption. With this we achieved great implementation speed and time for an acceptable security parameter.

General Term

Cryptography, Lattice

Keywords

Lattice-based Cryptography, Quantum, Cryptography, Closest Vector Problem, Lattices, Quantum.

1. INTRODUCTION

*Lattices we*re first studied in the late 18th century by mathematicians Joseph Louis Lagrange, Carl Friedrich Gauss and later Minkowski. Lattice is applied in different fields but its application to cryptography was based on the work of [2]. The construction of cryptographic system constitutes mathematical problems which are hard to solve. Mathematical problems based on lattices include the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP) which are used as an algorithmic tool to solve a wide variety of problems.

Cryptography [3], it is the art and science of encrypting messages for secure communication. According to [4], its definition needs to be extended as designing of algorithms, protocols and systems which are used to protect information against threats. This is achieved via authenticity, confidentiality, data integrity and non-repudiation. The three division of cryptography are Symmetric-key cryptography, Public-key (asymmetric-key) cryptosystems and Hash function.

Symmetric-key cryptography refers to the use of a single key by both the sender and receiver for encryption and decryption. Thus use to achieve confidentiality and privacy. The modern study of symmetric key ciphers relates mainly to the study of block ciphers and stream ciphers, and to their applications. This type of cryptography exited for a long time and was the only kind of encryption publicly known until the work of [5] proposing the public key. *Public-key* cryptography is characterized by the use of different encryption and decryption keys that is, each user makes the encryption key publicly available but keeps the decryption key secret [6]. This scheme is suited for non-repudiation and user authentication since key exchange is a major application. Cryptographic *hash function* does not make use of key since the plaintext is not recoverable from the ciphertext unlike the other two types of cryptography. It takes a message of any length as input, and output a short, fixed length hash. Hence, hash functions are suited for ensuring data integrity since any change made to the contents of a message will produce a different hash value. This makes it difficult for two different messages to yield the same hash value.

Lattice-based cryptography is the construction of cryptographic functions which are at least hard to break via the use of lattices as a source of computational hardness [7]. The work of [2] sparked a great interest in understanding the complexity of lattice problems and their relation to cryptography. With further studies in the field of lattice-based cryptography, its cryptographic constructions are found to be typically quite efficient, simple to implement, compete with the best known alternatives, they are believed to be secure against quantum computers and of course, hold a great promise for post-quantum cryptography [8]. Unlike lattice-based cryptography according to [1], with the continuous advancements in the field of quantum computing, the security of many existing public key cryptosystems has been demonstrated to be broken in the theoretical sense.

Worst-case hardness of lattice problems indicates how hard it is to break the cryptographic construction even in its worst case. According to [8], even with some small non-negligible probability, breaking the cryptographic construction is provably at least as hard as solving several lattice problems (approximately, within polynomial factors) in the worst case. Hence, successfully attacking a random instance of a cryptosystem immediately implies being able to solve all instances of the underlying problem [9]. Virtually all other cryptographic constructions are based on average-case hardness. The importance of the worst-case security guarantee as stated by [8] is twofold. First, it assures us that attacks on the cryptographic construction are likely to be effective only for small choices of parameters and not asymptotically. In other words, it guides us in making design decisions. Second, in principle the worst-case security guarantee can help us in choosing concrete parameters for the cryptosystem, although in practice this leads to what seems like overly conservative estimates.

2. LATTICE

2.1 Lattice Problems

Lattice problems have unique computational complexity properties and have many important applications. This includes [10]: the polynomial time algorithm to factor polynomials, breaking certain public-key cryptosystems, solving integer programming in fixed dimensions in polynomial time, and solving simultaneous Diophantine approximations. Lattice is used for cryptography for a number of reasons [11]: simple to implement; efficient (linear, parallelizable); resists sub-exponential & quantum attacks (so far); security from worst-case assumptions and lattice problems offer the possibility of faster encryption and decryption algorithms.

Definitions of some lattice problems.

Definition (SVP): Find the shortest non-zero vector in L, i.e. find $\mathbf{v} \in L \neq 0$ such that $||\mathbf{v}||$ is minimized.

Definition (CVP): Given a target vector t (not necessarily in the lattice) in L, find the vector $\mathbf{v} \in L$ closest to t, i.e. find $\mathbf{v} \in L$ such that $||\mathbf{v}- t||$ is minimized.

Definition (SIVP): Given a lattice basis $B \in Z^{n\times\ n}$, find n linearly independent lattice vectors $S = [s_1,...,s_n]$ (where $s_i \in \mathcal{L}(B)$ for all i) minimizing the quantity $\|S\| = \max_i \|s_i\|$.

Definition (uSVP): Find the shortest vector in a lattice whose shortest non-zero vector is shorter by some factor γ than all other non-parallel lattice vector.

2.2 Lattice Theory

A lattice is a set of points in n-dimensional space with a periodic structure, such as the one illustrated in Figure 1. [8] defined lattice as the set of all integer combinations of n linearly independent vectors $b_1, \dots, b_n$ in $\mathbb{R}^n$. The set of vectors $b_1, \dots, b_n$ is called a basis for the lattice. A basis can be represented by the matrix $B = [b_1, \dots, b_n] \in \mathbb{R}^{n\times n}$ having the basis vectors as columns.

$$L(b_1, \dots, b_n) = \{\sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z}\, for\ 1 \leq i \leq n\}$$

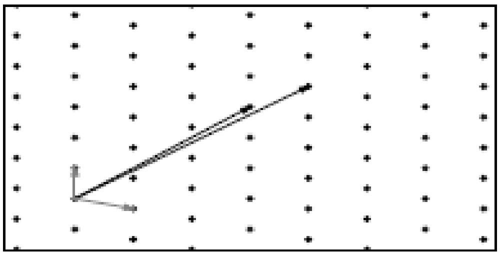

Figure 1: A two-dimensional lattice and two possible bases

Lattice notations: the set of real numbers is denoted by R and the set of integers by Z; real numbers by Greek letters (such as α, γ, δ) and integers by small letters (such as *i, j, l*); vectors by bold-face lower case letters (such as v, t, x); and capital letters denotes matrices or set of vectors (such as B, M).

Lattice dimension: Let $n \in \mathbb{N}$. A lattice of dimension n is a set of the form

$$\mathcal{L} = \mathcal{L}(B) := \{Bx|\ x\ \mathbb{Z}^n\} \subseteq \mathbb{R}^n$$

The dimension of a vector space is equal to the number of elements in the basis set e.g. R^2

$$Dim(\mathcal{L}) = n$$

Bases: A bases can be represented by the matrix $B = [b_1, \dots, b_n] \in R^{mxn}$ where $[b_1, \dots, b_n]$ is the column vectors.
A basis is any set of vectors that are spanning set and are linearly independent

Determinant: Given a basis B of the lattice $\mathcal{L}$, the lattice determinant is

$$\det(L) = \sqrt{det(BB^T)}$$

The value of the determinant is independent of the choice of the basis and geometrically corresponds to the inverse of the density of the lattice

All bases have the same determinant: For two bases B and C of the same lattice $\mathcal{L}$, there exist a unique Transformation matrix T, which is invertible over the integers such that BT = C and conversely, multiplying C with another transformation matrix T^{-1} will yield another bases of the same lattice i.e. $CT^{-1} = B$

Dual lattice: Let $B = [b_1, \dots, b_n]$ be a basis for some lattice in R^n. The *dual* of a lattice $\mathcal{L}$ in R^n, denoted $\mathcal{L}^*$, is spanned by the rows of the matrix

$$B^{-1} = [b_1^*, \dots, b_n^*]$$

$$\mathcal{L}^* = \mathcal{L}(B^{-T})$$

Orthogonality defect: Let B be a real non-singular n x n matrix. The orthogonality defect of B is defined as

$$orth-defect(B) \overset{\text{def}}{=} \frac{\Pi_i||b_i||}{\det(B)}$$

where $||bi||$ is the Euclidean norm of the i^{th} column in B, $\overset{\text{def}}{=}$ = equality by definition and ∏ = N-ary product of i

Orthogonality defect is the quantity used to measure how close a basis is to the orthogonal.

q-ary lattices: Let $q \in Z_+$. A lattice of $\mathcal{L}$ of dimension n is said to be q-ary if $qZ^n \subseteq \mathcal{L}$. Given positive integers n, m, q and a matrix $A \in \mathbb{Z}_q^{nxm}$, we can define two q-ary lattice:

The first is generated by the (transposed) rows of A
$\Lambda_q(A) = \{y \in Z^m : y = A^T s \bmod q \text{ for some } s \in Z^n\}$

The second lattice of those integer vectors that are orthogonal (modulo q) to the rows of A
$\Lambda\frac{1}{q}(A) = \{y \in Z^m : Ay = 0 \bmod q\}$

where q is a prime integer and y is a vector. Most *lattice-based cryptographic constructions* use q-ary lattices as their hard-on-average problem.

Lattice basis reduction: Given an integer lattice basis as input, find a basis with short, nearly orthogonal vectors.

This is realized using different algorithms, whose running time is usually at least exponential in the dimension of the lattice.

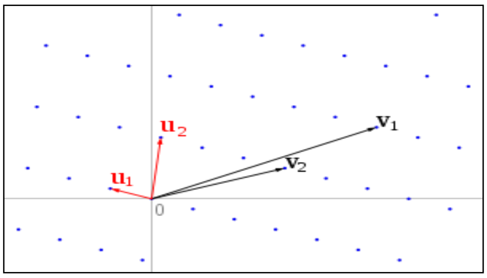

Figure 2: Lattice reduction in two dimensions

The black vectors are the given basis for the lattice (represented by blue dots), the red vectors are the reduced basis.

3. IMPLEMENTATION

To implement the algorithms, we used the libraries Number Theory Library (NTL) and GNU MultiPrecision (GMP) as supplemental long integer package. The programming language used was C on Linux (Ubuntu), Core 2 Duo 1.66GHz of 1GB RAM.

3.1 Closest Vector Problem (CVP)

There are several algorithms for solving CVP such as Babai's nearest plane method (although not guaranteed to solve CVP), Babai's rounded technique, exponential time algorithm but this thesis considers embedded technique.

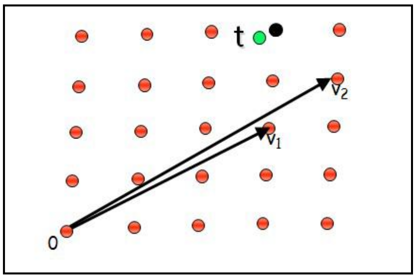

Figure 3: Closest vector problem

Embedded technique: Let B be a basis matrix for a lattice $\mathcal{L}$ and suppose w ∈ R^n, a solution to the CVP corresponds to integers $l_1,\ldots l_n$ such that

$$w \approx \sum_{i=1}^{n} l_i b_i$$

The crucial observation is that

$$e = w - \sum_{i=1}^{n} l_i b_i$$

is such that ||e|| is small

The idea of embedding technique is to define a lattice $\mathcal{L}'$ that contains the short vector *e* [12]

Lemma: Consider the basis matrix of a lattice Z^3 to solve the CVP instance with w = (100, 100, 100).

To solve this, we apply the LLL algorithm to the basis matrix (taking M = 1) for the lattice L'. The LLL lattice reduction is the process of calculating a nearly orthogonal lattice basis from an arbitrary one. Then we find e by solving the SVP problem in the lattice L′. One can then solve for the CVP by subtracting e from w. Hence, we find e by solving the SVP problem in the lattice L′. One can then solve for the CVP by subtracting e from w.

3.2 Key Generation

This thesis adopts the key generation of PRS because of the security they achieved by making improvement on the private basis of GGH construction and Micciancio's public basis.

Private basis use GGH's private basis construction, namely $\mathbb{R} \leftarrow bI + M$ (as it's shown in Algorithm 4.1, line 9).

It allows generalizations about the bound on the size $||R||\infty$ which allows for faster key generation and it provides a more orthogonal basis with which to perform decryption, which in turn decreases the size necessary to ensure correct decryption using CVP embedded technique. This in turn allow us to decrease the size of the coefficients while keeping the same security parameter, saving storage and transmission space and increasing efficiency.

Public Basis use Micciancio's method of applying a HNF reduction on the private basis (as it's shown in Algorithm 4.1, line 10).

This provided a greater level of security, simplified key storage and much smaller public keys.

Algorithm 1: Key-Generate

```
Input:   $n \in \mathbb{N}$ the security parameter.
Output : $B \in \mathbb{Z}^{n,n}$ the public key, $\mathbb{R} \in \mathbb{Z}^{n,n}$ the private key.
begin
  1:   $M \leftarrow 0$
  2:   for $i, j \leftarrow 0$ to $n - 1$ do
  3:       $M_{i,j} \leftarrow \mathbb{R}\mathrm{and}(-1, 1)$
  4:   end for
  5:   $b \leftarrow \lceil 2\sqrt{2n/3} \rceil$
  6:   repeat
  7:       $b \leftarrow b + 1$
  8:   until $\|(bI + M)^{-1}\|\infty \leq 1/2$
  9:   $\mathbb{R} \leftarrow bI + M$
  10:  $B \leftarrow \mathrm{HNF}(\mathbb{R})$
End
```

3.3 Encryption/ Decryption

For *encryption /decryption* we improved on the vector reduction basis. For our trapdoor function, we added a small error vector to a lattice point since given any basis of a lattice, it is easy to generate a vector which is close to a lattice point (considering our CVP). From this one-way, it is hard to return from the closest lattice vector to the original lattice point since we added the small error vector. Thus we used two different bases of the same lattice so a basis allows computing the function and the other the inverse function by permitting good approximation to the CVP.

To *encrypt* a message, we first map it to a lattice point by taking the integer combinations specified by the message of the public basis vectors and then add to the lattice point a small error vector chosen at random. To *decrypt*, we look for a lattice point which is close to the ciphertext. By using the private basis which is a reduced basis, the correct decryption is obtained with high probability.

3.4 Test and Results

The dimension *n* which is the main security parameter was tested for various values but we base our work on dimension 400 and 800 for comparison with the work of PRS. The parameters used for this thesis are influenced by security considerations, application platform, constraints of the particular computing environment, and constraints of the particular communications environment hence, it is difficult to decide on a single "best" set of choices. Though it has been proved that the larger the value of *n*, the more secure the system will be.

Table 1: Speeds and key sizes

Dimension	Enc.	Dec.	Pub. Key	Priv. Key
400	0.03s	0.05s	313.3KB	123.2 MB
800	0.1s	0.3s	917.1 KB	412.3 MB

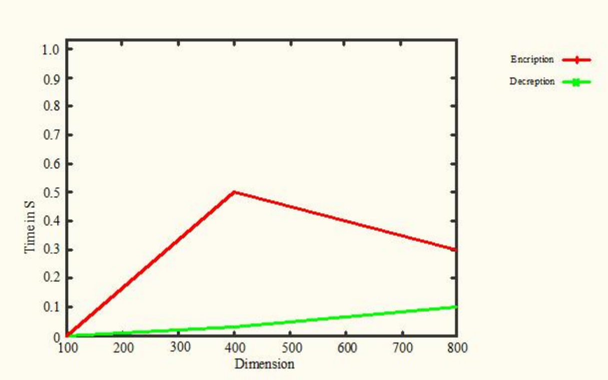

Fig. 4: Performance result

4. CONCLUSION

The fact that the security of many existing asymmetric key cryptosystems has been demonstrated to be broken in the theoretical sense with the continuous advancements in the field of quantum computing reveals the need for technological advancements that is, lattice based cryptography. Also there is the need for the existence of large-scale quantum computers to implement on. Unlike a classical computer, in which a bit can represent either 1 or 0, in a quantum computer a bit can represent 1 or 0 or a mixture of the two at the same time, letting the computer perform many computations simultaneously and that would shorten the time needed to break a strong 1024-bit RSA code from billions of years to a matter of minutes. The fundamental problem in lattice-based cryptography is that there exist a widely used efficiency improvement which entails the use of newer security assumptions and however, analyzing these thoroughly is still an open problem.

5. REFERENCES

[1] Plantard T., Rose M. and Willy S. (2009). Improvement of Lattice-based Cryptography using CRT. School of Computer and Software Engineering, University of Wollongong NSW, Australia.

[2] Ajtai, M. (1996). Generating Hard Instances of Lattice Problems. In Proceedings of the 28th annual ACM Symposium on Theory of Computing, New York, USA.

[3] Alese, B. K. (2000). Vulnerability Analysis of Encryption/Decryption Techniques of Computer Network Security. Master's Thesis, Department of Computer Science, Federal University of Technology, Akure, Nigeria.

[4] Edward, P. (2011). Comparative Analysis of Public-Key Encryption Schemes. Master's Thesis, Department of Computer Science, Federal University of Technology, Akure, Nigeria.

[5] Diffie and Hellman (1976). New Directions in Cryptography. IEEE Transactions on Information Theory, Vol. IT-22, pp 644 – 654.

[6] Barreno, M. A. (2002). The Future of Cryptography Under Quantum Computers. Department of Computer Science. Dartmouth College. Technical Report TR'02 – 425.

[7] Micciancio, D. (2003). Lattice-Based Cryptography. University of California, San Diego.

[8] Micciancio, D. and O. Regev (2008). Lattice-Based Cryptography. In Proceedings of Crypto 2009, pages 577 – 594, Springer.

[9] Ruckert, M. and M. Schneider (2010). Estimating the Security of Lattice-Based Cryptosystems. Technischen Universitat Darmstadt, Department of Computer Science, Cryptography and Computer algebra, Germany.

[10] Cai J. and H. Zhu (2005). Progress in Computational Complexity Theory: Lattice Problems. In Journal of Computer Science and Technology, Vol. 20, No. 6, pages 735 – 750, Springer.

[11] Peikert, C. (2009). Some Recent Progress in Lattice-Based Cryptography. TCC'09 (ppt).

[12] Galbraith S. (2011). Mathematics of Public Key Cryptography. Department of Math, University of AucMand, New Zealand. http://www.isg.rhul.ac.uk/sdg/crypto-book