Preventing Information Leakage from Encoded Data in Lattice Based Cryptography

Aarti Dadheech
Computer Science and Engineering Department
MBM Engineering College
Jodhpur, India
grtdadheech@gmail.com

***Abstract*—Lattice based Cryptography is an important sector which is ensuring cloud data security in present world. It provides a stronger belief of security in a way that the average-case of certain problem is akin to the worst-case of those problems. There are strong indications that these problems will remain safe under the availability of quantum computers, unlike the widely used issues like integer-factorization and discrete logarithm upon which most of the typical cryptosystems relies. In this paper, we tend to discuss the security dimension of Lattice based cryptography whose power lies within the hardness of lattice problems. Goldreich-Goldwasser-Halevi (GGH) public-key cryptosystem is an exemplar of lattice-based cryptosystems. Its security depends on the hardness of lattice issues. GGH is easy to understand and is widely used due to its straightforward data encoding and decoding procedures. Phong Nguyen, in his paper showed that there's a significant flaw within the style of the GGH scheme as ciphertext leaks information on the plaintext. Due to this flaw the practical usage of GGH cryptosystem is limiting to some extent. So as to enhance the safety and usefulness of the GGH cryptosystem, in this paper we proposed an improvised GGH encryption and decryption functions which prevented information leakage. We have implemented a package in MATLAB for the improvement of GGH cryptosystem. In our work we proposed some methods to improve GGH algorithm and make it more secure and information leakage resistant.**

Keywords— Cloud Data Security, Cryptography, Lattice based Cryptography, GGH Cryptography

I. Introduction

With the rise in cryptanalytic attacks and quantum computer generation, conventional cryptographic schemes will soon become obsolete. We need an alternate security mechanism which remains secure in presence of quantum computers and is as hard as the existing number theoretic approaches. Lattice-based-cryptography is an one in all the foremost promising candidates of post quantum cryptography [1]. There is a robust notion that lattices are able to resist the attacks performed by quantum computers. It's introduced by Ajtai in 1996, as a public-key cryptosystem using the lattice problem [2]. In explicit, two lattice-based public-key cryptosystems have accepted widely: the Ajtai-Dwork cryptosystem (AD) and also the Goldreich-Goldwasser-Halevi cryptosystem (GGH) [3, 4]. AD is mainly theoretical model and the GGH is suggested as practical alternative to number theory. In fact, it's the lattice version of the first code-based cryptosystem, designed by McEliece. Some major enhancements are suggested by Micciancio for the GGH cryptosystem related to its speed and the security. In scheme suggested by Micciancio, the public-key is the Hermite Normal Form (HNF) of the private-key, therefore, the size of the public-key in Micciancio scheme is abundantly smaller than the public-key size of the GGH cryptosystem [5]. At Crypto'99 Phong Nguyen showed that ciphertext of GGH leaks information on the plaintext and this can be a major flaw in the design of the GGH cryptosystem scheme [6]. In this paper we suggested some methods by which flaw of information leakage can be removed from GGH cryptosystem and system become more reliable to use.

The paper is organized as follows. Firstly, an overview about lattices is given in the coming section. After the overview in first section, we tend to, in short describe the GGH cryptosystem in Section 2 and justify how the encoding method leaks data, with observation. In Section 3 we tend to describe the proposed methodology to cure information leakage problem. In Section 4 experiments are done. And in last, we summarized our work.

A. Lattice Theory

Lattice-based-cryptography could be a new approach towards cryptological protection of data in computer systems and conjointly ensuring cloud data security. It's a counterpart of a lot of normally known, completely tested and smoothly-working traditional algorithms (such as RSA, DSA, AES), that appear so far to meet their purpose more than adequately. A most appealing property of lattice based cryptography schemes is that they're resilient to cryptanalysis even within the presence of quantum computers. Lattices were initially studied by mathematicians Joseph Louis Lagrange and Carl Friedrich Gauss and afterward in 1996; Miklos Ajtai and Micciancio mentioned the employment of lattices as cryptography primitive. Micciancio outlined lattices as general class of cyclic lattices (ideal lattice). A lattice could be a set of points within the *n* dimensional Euclidean space with a robust property of cyclicity. See Fig 1, here the set of vectors b_1, b_2 and e_1,e_2 are called a basis for the lattice. Here in below figure e_1,e_2 forms a "good basis" with nearly orthogonal vectors and b_1, b_2 a "bad basis".

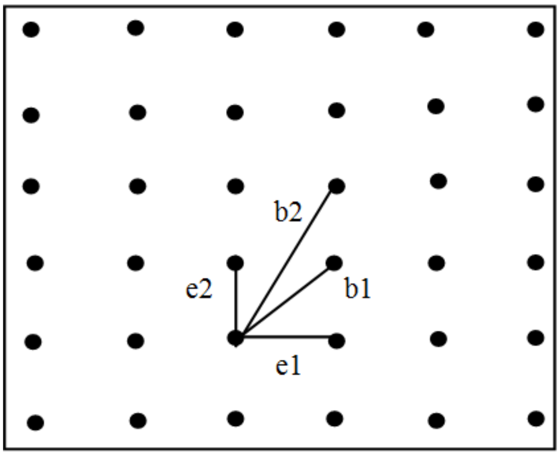

Fig. 1. Lattice generated by basis [$e_{1,}e_2$] and [$b_{1,}b_2$]

978-1-5386-5314-2/18/$31.00 ©2018 IEEE

There are two kinds of lattice-based hard mathematical problems. The first is Shortest Vector Problem (SVP) and the second one is Closest Vector Problem (CVP). The problem statement of SVP is that a basis of a lattice is given, find the shortest vector in the lattice. The problem statement of CVP states that given a basis of a lattice and a vector *c* find a vector v ∈ L that is closest to *c* among all points of L. Both of these problems are easy to solve with "good basis" and hard to solve with "bad basis". The Goldreich–Goldwasser–Halevi (GGH) cryptosystem is designed on the basis of the fact of hard lattice problem to find closest vector in lattice.

B. Pre-requisite

This section covers the necessary background required throughout this work. First, we give some general notations; define matrices, since it is a fundamental building block in lattice-based GGH cryptosystem.

Rank

The rank of a lattice is the number of linearly independent vectors in any basis for that lattice. The lattice L(B) is full rank $n=m$ if, i.e. if B spans the entire vector space IR_m.

Determinant

For a rank n lattice L, its determinant denoted by det(L) is defined as the n-dimensional volume of L(B). Mathematically, $\det(L)=\sqrt{\det(B^TB)}$ .When L is full rank, $\det(L) = |\det(B)|$.

Unimodular Matrices

A unimodular transformation matrix is defined as an integer matrix, whose inverse is also integer. This implies the following properties:

1. U must be integral.
2. U must be square.
3. det(U) must be exactly ±1.

Orthogonality defect

Let B be a non-singular n x n matrix. Then the orthogonality defect of lattice B can be defined as the product of the length of the basis vector and the ratio of this product with the parallelepiped volume of basis. It is a measure of nearly orthogonal.

$$\text{Orth-defect}(B) =||b_i|| /(\det(B)\Pi)$$

The good basis have relatively short vectors and have low orthogonality defect i.e. vectors are appreciably orthogonal to each other. The bad basis has long skewed vectors and have high orthogonality defect.

II. GGH CRYPTOSYSTEM

The Goldreich–Goldwasser–Halevi (GGH) cryptosystem is an equated asymmetric public-key cryptosystem based on lattices. GGH cryptosystem makes use of the very fact that the nearest vector problem is often a tough problem with a "bad basis" however it is easy with a "good basis". This scheme uses a one-way trapdoor function and the thought enclosed in this function is that given any basis for a lattice, it's simple to come up with a vector that is near to a lattice point, as an instance taking a lattice point and adding a tiny low error vector. But to return from this incorrect vector to the initial lattice purpose a special basis is required. In 1999, Nguyen attacked the hypothesis of GGH cryptosystem and broke the sensible proposition of it. Moreover, GGH algorithm is difficult to be broken even in its average case.

Since the complexity of solving CVP has been proved to be NP-hard on average cases, the GGH cryptosystem is designed to be a novel encryption and decryption mechanism based on hardness of solving CVP.

A. Algorithm

The cryptosystem depends on two parameters, the dimension n and the security parameter σ. Given a target point and a lattice with two different bases, namely the *Private-Key* and the *Public-Key.* The Private Key is a approximately orthogonal basis $B_{good\cdot}$ and the Public Key B_{bad} is a bad basis that is far away from being orthogonal. According to the Babai's rounding off algorithm, for the target point, the good basis B_{good} can find the correct closest lattice point with a high possibility, but B_{bad} cannot solve CVP in the lattice. In such way, the data will be transferred successfully yet keep security [7].

1) Key Generation: Party A begins with constructing a Private Key,

$$R=k.I_n+ Q \quad (1)$$

Here $k=\sqrt{n}\ .l$. Value of $l=4$, and Q is a random perturbation matrix with entries from $\{-l,...,l\}$. A public key B can be created by computing

$$B =U.R \quad (2)$$

Here U is "random" unimodular matrix with det(U) = ±1.

2) Encryption: Compute ciphertext by adding randomly chosen error matrix e ∈ {±σ} a,

$$c=m.B+e \quad (3)$$

3) Decryption: To decrypt this cipher c compute:

$$\text{round}(c.R^{-1}) \quad (4)$$

Here *round()* is Babai's rounding technique. It will be used to remove the error term as it is a small value.

B. Security

Firstly, if a third person listened the ciphertext c, the only basis available to him is the Public-key B, which is a hardly orthogonal basis for the lattice L. But there is no known polynomial time algorithm to solve CVP exactly, or to approximate it to within a polynomial factor using bad basis. Hence, the plaintext decrypted by is incorrect. It is easy to find the closest vector with a "good basis" but difficult to do so with a "bad basis".

Second note, that by construction, the public basis B has high orthogonality defect and the private basis R is far away from orthogonality. Thus in order to determine the private key given only the public key, an eavesdropper would again need to solve an instance of hard problem. It is easier computing a "bad basis" out of a "good basis" than computing a "good" out of "bad" one [8].

C. Weakness

Phong Nguyen, in his paper showed that there are some major flaws in the design of the GGH scheme. His "leaky remainder" attack exploited the specific form of the error vector, allowing recovering the plaintext remainders: $c + \sigma \equiv$ mB-1 (mod 2σ). This simplified the CVP instance, making plaintext recovery tractable and ciphertext can be easily tested for various plaintexts. Apart from the various attacks like the Round-off Attack, the Nearest-plane Attack and the embedding Attack, the following are also the weaknesses discovered by Nguyen in his work:

- **Short error vector**
  GGH error vectors are significantly shorter than the lattice vectors, making GGH CVP instances much easier than that of general Closest Vector instances.
- **Special form of error vector**
  GGH cryptosystem selects the value of the error vector e as either +σ or -σ, where σ is security parameter. This makes the original extremely vulnerable to various attacks. This σ = (σ,σ,…,σ) ϵ Z^n will reveal partial data regarding the plaintext.
- **"Leaky Remainder" Attack**
  The error vector from the ciphertext could be removed with a well-chosen modulus. By adding σ to every element of the ciphertext, the error vector changes from $\{-\sigma, \sigma\}^n$ to $\{0, 2\sigma\}^n$. The error vector can then be completely removed by reducing modulo 2σ.

III. Proposed Methodology

In this section we are proposing two functions which may be embedded with GGH encoding formula and GGH decipherment formula to stop information leakage from GGH ciphered text. The *pixelPermutation()* function is included to the GGH encryption algorithm and *reconstruct()* function is appended to GGH decryption algorithm. Thenew improvised algorithm is shown in Table I.

TABLE I. New Proposed Algorithm

Party A	Party B
Key Generation	
1. Choose a nearly orthogonal basis **R** as Private Key; **2.** Compute Bad basis **B=U*R** 3. Publish **B** as the Public Key	
Encryption	
	1. Choose a message **m** **2.** Choose an error vector **e** **3.** Use A's Public Key to encrypt m: **c=mB+e** **4.** Compute: **c_{out}=pixelPermutation(c)** **5.** Send the ciphertext c_{out} to Party A
Decryption	
1. Reconstruct the cipher **c_{rec}=reconstruct(c_{out})** 2. Compute round(c_{rec}*R^{-1}) to find the original message	

The new ciphertext generated by *pixelPermutation()* when leaked to third party then it will not reveal any information about the plaintext because pixels of ciphertext c are shuffled using random shuffling in MATLAB. Now $c_{out.}B^{-1}$ reveals no information about plaintext. The working of both functions is explained in coming lines.

- *pixelPermutation():* The function will take input a ciphertext *c* generated by GGH encryption algorithm and output a new ciphertext c_{out} with shuffled pixels values.
- *reconstruct():* It reconstruct the ciphertext *c* (here symbolize as c_{rec}) from $c_{out.}$

IV. Experimental Result and Analysis

For experimental purpose we have choose a image of size 100x100 as input message i.e. the value of n=100 and apply both GGH encryption algorithm and New proposed encryption algorithm on this same message and analyze their respective ciphertexts against "information leakage" attack. Fig. 2 below shows what happen after ciphertext generated by both the algorithms are attacked by third person.

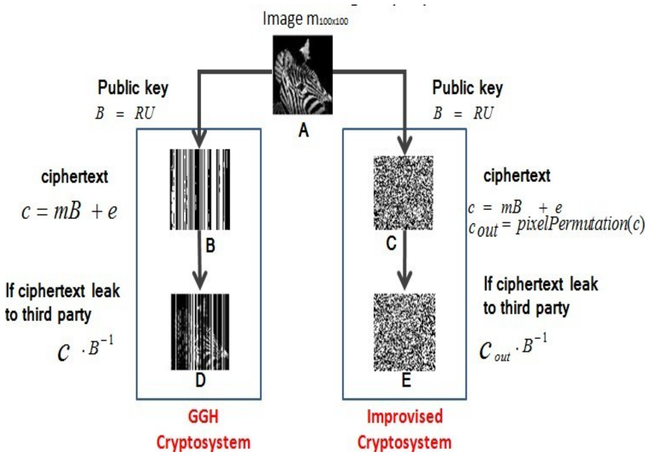

Fig. 2. Comparison of GGH and Proposed scheme's ciphertext i.e. comparing *c* and c_{out}

The person tries to get information out of ciphertext by multiplying the cipher with inverse of Public key. In this case the GGH's ciphertext reveals the information where as ciphertext generated by proposed scheme is robust against such attacks and reveals no information.

Conclusion and Future Work

In our work, we implemented GGH cryptosystem using some advanced function by which the problem of information leakage can be prevented. The improved efficiency allows using wider range of security parameter while maintaining the scheme reasonably practical. This new scheme is presented for grayscale images, work can also be done on text and colored images in future. Also scheme can be improve in terms of space complexity as the size of public key and their corresponding ciphertext is much larger.

Acknowledgment

I owe a debt of deepest gratitude to Dr. Anil Gupta, Professor and Head, Department of Computer Science & Engineering, M.B.M. Engineering College, Jodhpur, for his guidance, support, motivation and encouragement throughout my work.

References

[1] Aarti Dadheech,Study of Lattice based FHE for Cloud data Security. International Journal of Advanced Research in Computer Science,Volume 8, Issue 7, July-August 2017.

[2] M. Ajtai, Generating hard instances of lattice problems.In Proceedings of 28th STOC, Philadelphia,(1996), pp. 99–108.

[3] M. Ajtai, and C. Dwork, A public-key cryptosystem with worst-case/averagecase equivalence.In Proceedings of 29th STOC, Texas, (1997), pp. 284–293.

[4] Goldreich O., Goldwasser S., Halevi S. Public-key cryptosystems from lattice reduction problems. Proceedings of 17th Annual International Cryptology Conference; Santa Barbara, California, USA, Springer Berlin / Heidelberg. (1997).

[5] D. Micciancio, Improving lattice based cryptosystems using the Hermite normal form. Proceedings Cryptography and Lattices Conference, pp. 126-145, (2001)

[6] P Nguyen, P. Q, Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97. Crypto 1999, LNCS 1666, pp 288-304. Springer-Verlag (1999).

[7] L. Babai, On Lovaśz lattice reduction and the nearest lattice point problem. Combinatorica, vol. 6, no. 1, pp. 1-13, Mar. (1986).

[8] Massoud Sokouti, Ali Zakerolhosseini and Babak Sokouti, Medical Image Encryption: An Application for Improved Padding Based GGH Encryption Algorithm. The Open Medical Informatic Journal, 2016, pp. 11–22.