International Journal of Cryptology Research 12(1): 46 - 66 (2022)

Survival of the First Practical Lattice-based Encryption Scheme: A Review

Arif Mandangan*[1], **Hailiza Kamarulhaili**[2], and **Muhammad Asyraf Asbullah**[3]

[1]Mathematics, Real-Time Graphics and Visualization Laboratory, Faculty of Sciences and Natural Resources, Universiti Malaysia Sabah
[2]School of Mathematical Sciences, Universiti Sains Malaysia
[3]Laboratory of Cryptography, Analysis and Structure, Institute for Mathematical Sciences, Universiti Putra Malaysia

Email: arifman@ums.edu.my
**Corresponding author*

ABSTRACT

Lattice-based cryptography is one of the most promising alternatives for the survival of cryptography against quantum computer attacks. The Goldreich-Goldwasser-Halevi Scheme (GGH Scheme) is the first practical lattice-based cryptographic scheme due to its simplicity which offers efficiency. The security of this scheme is based on the Closest Vector Problem (CVP) instance where the CVP is proven as NP-hard lattice problem. Unfortunately, there are some critical flaws on its design which make the scheme exposed to some security threats. The obvious way to avoid the attacks is by increasing the lattice dimension which makes the scheme suffered from some efficiency issues. In this paper, we revisit the early development of the scheme and then followed by the fatal attacks on it. Moreover, we address the current versions of the scheme and evaluate whether these versions can survive against the fatal attacks or not. We discuss the strength and weaknesses of the GGH Scheme in terms of its security which could be used as a guidance for further improvement on the scheme to make it better and ideal to be deployed in post-quantum era.

Keywords: lattice-based cryptosystem, post-quantum cryptography, embedded lattices, GGH cryptosystem

1 INTRODUCTION

One of the most promising post-quantum alternatives that immune to Shor's quantum attack is lattice-based cryptography which utilizes classical distance minimization problems in lattice to design one-way trapdoor function and later being used to develop cryptographic scheme. The first lattice-based cryptographic scheme was proposed by Ajtai and Dwork in 1987 (referred to as AD-Scheme) which is theoretically proven to be secure under worst-case lattice scenario (Ajtai and Dwork, 1987). Unfortunately, the AD Scheme is considered as impractical beside successful attack on it by Nguyen and Stern in 1998 (Nguyen and Stern, 1998).

The first practical scheme was proposed by Goldreich, Goldwasser and Halevi in 1997 (Goldreich *et. al*.,1997), referred to as GGH Scheme in this paper. The GGH Scheme is utilizing the Closest Vector Problem (CVP) and Shortest Basis Problem (SBP) as the underlying security core. Basically, the scheme is a lattice version of the famous McEliece's code-based cryptographic scheme proposed in 1978 (McEliece, 1978). Compared to the McEliece's Scheme, the GGH Scheme is considered practical due to its simplicity specially in key generation and encryption algorithms which could be done using simple algebraic operations. To build confidence on the security of the scheme, the inventors has published five challenges on the Internet, known as GGH Internet Challenges. In these challenges, five ciphertext that was encrypted using the GGH Scheme in the lattice dimensions of 200, 250, 300, 350 and 400 were published. By using the provided public basis and threshold parameter of each challenge, cryptanalysts are invited to decrypt the published ciphertexts.

The most devastating attack launched by Nguyen, P. Q (1999), referred to as Nguyen's attack. The attack completely decrypted all the challenges except for the largest dimension. Since that, Nguyen (1999) suggested that the lattice dimension to be applied should be larger than 400 for allowing the scheme to surpass the Nguyen's attack. However, the implementation of large lattice dimension consequently reduces the efficiency and practicality of the scheme. The final challenge for lattice dimension 400 has been solved by Lee and Hahn in 2010 (Lee, Hahn 2010). Since that, the GGH Scheme seems ready to be officially declared as a dead cryptographic scheme. However, a new hope was emerged in 2012 when Yoshino and Kunihiro proposed a novel improvement (referred to as GGH-YK Scheme) on the GGH Scheme (Yoshino and Kunihiro, 2012). Later in 2014, Barros and Schechter made some enhancement on the practicality part of the GGH-YK Scheme and proposed a new version that known GGH-YK-M Scheme (de Barros and Schechter, 2014). Since that, there is no more notable attack on the scheme can be found in the literature. From that, we consider that the GGH Scheme has survived and evolved to its new version GGH-YK-M Scheme.

This paper aims to review the early development of the GGH Scheme by focusing on the strength and weaknesses of its design. Then, we discuss the fatal attacks on the GGH Scheme by showing the weakness points that being exploited by the attacks. Furthermore, we address the corresponding improvement on the scheme. We evaluate whether the current version of the scheme is still prone to the fatal attack or not. However, efficiency issues of the GGH Scheme is not the main interest of this paper and will not be thoroughly discussed. Code-based improvement on the GGH Scheme also is beyond our scope. At the end of this paper, we summarize some important points related to the strength, weaknesses and challenges that need to be faced by the GGH-type scheme before it can be widely deployed in the post-quantum era.

2 MATHEMATICAL BACKGROUNDS

In this paper, all vectors will be considered as column vectors and represented by standard vector notation. For instance, $\vec{v} \in \mathbb{R}^m$ is a column vector with m real entries. All matrices will be represented by capital latter. Basically, lattice is a set of discrete vectors in $\mathbb{R}^m$. It can be defined as follow.

Definition 2.1. (*Hoffstein et. al., 2008*). Let $\vec{b}_1, \cdots, \vec{b}_n \in \mathbb{R}^m$ be a set of linearly independent vectors. A lattice $\mathcal{L}$ generated by the vectors $\vec{b}_1, \cdots, \vec{b}_n$ is the set of all vectors formed by linear combinations of the vectors $\vec{b}_1, \cdots, \vec{b}_n$ with integer coefficients. The lattice $\mathcal{L}$ can be denoted as $\mathcal{L}(\vec{b}_1, \cdots, \vec{b}_n) = \{\vec{v} = \sum_{i=1}^{n} \alpha_i \vec{b}_i \mid \forall \alpha_i \in \mathbb{Z}\}$.

The set of linearly independent vectors $(\vec{b}_1, \cdots, \vec{b}_n)$ is called a basis for the lattice $\mathcal{L}(\vec{b}_1, \cdots, \vec{b}_n)$ and the vectors $\vec{b}_i$ in the lattice basis are called basis vectors. The number of basis vectors in the lattice basis is called the dimension of the lattice and denoted as $\dim(\mathcal{L})$. For $\mathcal{L}(\vec{b}_1, \cdots, \vec{b}_n)$, the $\dim(\mathcal{L}) = n$. The number of entries in the basis vectors $\vec{b}_i$ is called the rank of the lattice. For basis vectors $\vec{b}_i \in \mathbb{R}^m$, each vector consists of m real entries. If $m = n$, then the lattice if referred to as a full-rank lattice. In this paper, we only consider full-rank lattices. For convenience, a lattice basis is always represented as a matrix where each basis vectors becomes the column of the basis matrix in the same order. For instance, the basis $(\vec{b}_1, \cdots, \vec{b}_n)$ can be represented as follows:

$$B = (\vec{b}_1, \vec{b}_2, \cdots, \vec{b}_n) = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & b_{n,n} \end{bmatrix}$$

where $b_{j,i} \in \vec{b}_i$ for all $i, j = 1, \ldots, n$. The lattice $\mathcal{L}(\vec{b}_1, \cdots, \vec{b}_n)$ now can be conveniently represented as $\mathcal{L}(B)$. Every lattice $\mathcal{L}$ with more than one basis vectors has infinitely many bases. Any two different bases have the same number of basis vectors.

Proposition 2.1. (*Hoffstein et. al., 2008*). Any two bases of lattice $\mathcal{L}$ are related by a unimodular matrix with integer coefficients and determinant equal to ± 1.

Definition 2.2. (*Goldreich et. al.,1997*). Let B be a real non-singular $n \times n$ matrix. The dual-orthogonality defect of B is defined as

$$\mathrm{orthdefect}(B)^* = \frac{\prod_i \|\vec{b}_i{}'\|}{|\det(B^{-1})|} = |\det(B)| \cdot \prod_i \|\vec{b}_i{}'\|$$

where $\vec{b}_i{}'$ is the i-th row in B^{-1}.

The orthogonality level of lattice basis can be measured by using dual-orthogonality defect. The $\mathrm{orthdefect}(B)^* = 1$ if and only if the columns of B are orthogonal to one another and $\mathrm{orthdefect}(B)^* > 1$ otherwise.

Definition 2.3. The i-th minimum of a lattice $\mathcal{L}$, denoted as $\lambda_i(\mathcal{L})$, is the radius of the smallest zero-centred ball containing at least i linearly independent lattice vectors.

Definition 2.4. Let $\mathcal{L}$ be a lattice. The lattice gap, denoted as $gap(\mathcal{L})$, is the real number such that

$$gap(\mathcal{L}) = \frac{\lambda_2(\mathcal{L})}{\lambda_1(\mathcal{L})}$$

Experimentally, the bigger the lattice gap, the easier to solve the underlying hard problem on it (Nguyen,1999). Most of the conjectured hard mathematical problems in lattice are related to distance minimization.

Definition 2.5. (*Micciancio, 2001*). The distance between $\vec{v} \in \mathbb{R}^m$ with a lattice $\mathcal{L}(B)$ is given as $dist(\vec{v}, \mathcal{L}(B)) = \min\{\|\vec{v} - \vec{w}\| \mid \vec{v} \in \mathbb{R}^n, \vec{w} \in \mathcal{L}(B)\}$.

Definition 2.6. (Shortest Vector Problem (SVP)). Given a lattice $\mathcal{L}$. Find a shortest nonzero vector $\vec{v} \in \mathcal{L}$ that minimizes the Euclidean norm $\|\vec{v}\|$.

Definition 2.7. (Closest Vector Problem (CVP)). Given a basis B for a lattice $\mathcal{L} \subset \mathbb{R}^n$ and a vector $\vec{w} \in \mathbb{R}^n$. Find a lattice vector $\vec{v} \in \mathcal{L}(B)$ that is closest to $\vec{w}$ which minimizes the Euclidean norm $\|\vec{v} - \vec{w}\|$.

Definition 2.8. (Shortest Basis Problem (SBP)). Given a basis B for a lattice $\mathcal{L} \subset \mathbb{R}^n$. Find the smallest basis B' for the same lattice $\mathcal{L}$.

The CVP is known to be NP-hard, and the SVP is NP-hard under a certain randomized reduction hypothesis (Hoffstein *et. al.*, 2008).

3 THE GGH SCHEME

The core idea behind the GGH Scheme is using the fact that a lattice may have infinitely many bases with different quality in terms of orthogonality. A basis with low orthogonality defect is considered as a good basis with reasonably orthogonal basis vectors. On the contrary, a basis with high orthogonality defect is considered as a bad basis with highly non-orthogonal basis vectors. Therefore, two different bases with different orthogonality defects will be used as private and public keys respectively. These bases generate the same lattice L, i.e., $\mathcal{L}(R) = L = \mathcal{L}(B)$.

Algorithm 1. The GGH Scheme (*Goldreich et. al.,1997*).

Parameter Setup by Alice.

Sets the value of the following parameters:

i) Lattice dimension n.
ii) Threshold parameter σ.
iii) Private key parameters $l \in \mathbb{Z}$ and $k = l\lceil\sqrt{n} + 1\rceil$.

Key Generation by Alice.

i) Generates an n-by-n perturbation matrix P with entries $p_{i,j} \in \{-l, \dots, l\}$.
ii) Computes $R = kI + P$ where I is an n-by-n identity matrix.
iii) Computes the dual orthogonality defect of R. If it is “close enough” to 1, then R is accepted as a private key. Otherwise, generate another R.
iv) Generates an n-by-n unimodular matrix T.
v) Computes $B = RT$.
vi) Computes the dual orthogonality defect of B. If it is “far enough” from 1, then B is accepted as a public key. Otherwise, compute another B.
vii) Sends the public basis B with the public parameters $\{n, \sigma\}$ to Bob.

viii) Keeps the other parameters with the private basis R secretly.

Encryption by Bob.

i) Encodes the message into a vector $\vec{m} \in \mathbb{Z}^n$.
ii) Generates an error $\vec{e} \in \mathbb{Z}^n$ with entries $\{\pm\sigma\}^n$.
iii) Computes the ciphertext $\vec{c} \in \mathbb{Z}^n$ as $\vec{c} = B\vec{m} + \vec{e}$.
iv) Sends the ciphertext $\vec{c}$ to Alice.

Decryption by Alice

i) Computes the vector $\vec{t} \in \mathbb{R}^n$ as $\vec{t} = R^{-1}\vec{c}$.
ii) Forms an integral vector $\lfloor\vec{t}\rceil$ by rounding each entry of the vector $\vec{t}$ as $\lfloor t_i \rceil$ for all $i = 1, \ldots, n$.
iii) Computes a lattice vector $\vec{v}' \in \mathbb{Z}^n$ as $\vec{v}' = T\lfloor\vec{t}\rceil$.
iv) Computes the error $\vec{e}$ as $\vec{e} = \vec{c} - \vec{v}'$.
v) Computes the message $\vec{m}$ as $\vec{m} = B^{-1}(\vec{c} - \vec{e})$.

In the encryption formula, $B\vec{m}$ is a lattice vector, denoted as $\vec{v}$, since B is a lattice basis and $\vec{m}$ is an integral vector. The vector $\vec{v} = B\vec{m}$ then being perturbed σ away by error $\vec{e}$ to form the ciphertext $\vec{c}$. In this case, the security of the scheme lies on the CVP. To recover the message $\vec{m}$, we need to solve the CVP by correcting the error $\vec{e}$ and getting the lattice vector $\vec{v} = B\vec{m}$ which is the closest lattice vector to the ciphertext $\vec{c}$. This task can be done by using Babai's round off method which works efficiently only if the basis being used is reasonably orthogonal.

Since Alice has the reasonably orthogonal private basis R, then she is able to run the Babai's round off method efficiently to get the correct closest vector $\vec{v}'$ which is implicitly the lattice vector $\vec{v} = B\vec{m}$ and proceeds to decode the secret message $\vec{m}$. The Babai's round off method would not be able to solve the CVP by using public basis B since it is a highly non-orthogonal basis. However, the orthogonality of this public basis can be improved by using lattice reduction algorithms.

In this case, the security of the scheme lies on the SBP. By solving the SBP, the public basis B can be transformed into its reduced form with better orthogonality. Then, it can be used in the Babai's round off method to solve the underlying CVP and eventually recover the message $\vec{m}$. To make the SBP harder, larger lattice dimension should be used to make the lattice reduction algorithms inefficient.

Another important factor which influences the successfulness of the decryption process is the threshold parameter σ which generates the entries of the error $\vec{e}$. If the value is too big, then the ciphertext vector $\vec{c}$ will be located too far from the lattice vector $\vec{v} = B\vec{m}$ which increases the probability of decryption failure. If the value is too small, then ciphertext vector $\vec{c}$ will be located too close to the lattice vector $\vec{v} = B\vec{m}$ and makes attempt to attack the scheme easier. Therefore, the inventors of the GGH Scheme suggested suitable bounds for the threshold parameter σ in their paper (Goldreich *et. al.*,1997).

To build confidence on the security of the scheme, the inventors published Internet GGH Challenges which invite any cryptanalyst to attack the scheme for lattice dimension $n = 200{,}250{,}300{,}350, 400$. Most of the launched attacks were used the embedding technique which is considered as the best way to solve the CVP. The only succeed attack was on the lattice dimension 200. That means, by applying lattice dimension more than 200, the scheme conjectured to be secure, until Nguyen successfully showed that the embedding technique still can attack the scheme up to lattice dimension 400 due to some major flaws on the design of the scheme which makes the underlying CVP instance can be reduced to SVP which is considered easier than the CVP.

4 FATAL ATTACKS ON THE GGH SCHEME

In 1999, Phong Q. Nguyen solved almost all the Internet GGH Challenges. We refer this attack as Nguyen's attack. For the lattice dimension $n = 400$, he discovered partial of the published ciphertext. He noticed that, the structure of the error $\vec{e} \in \{\pm\sigma\}^n$ could be exploited to get partial information of the message $\vec{m}$ from the corresponding ciphertext $\vec{c}$.

From the partial information, he is able to simplify the underlying CVP instance to an easier instance with smaller error vector. By setting a vector $\vec{s} \in \{\sigma\}^n$, the congruence $\vec{e} + \vec{s} \equiv \vec{0} (\text{mod } 2\sigma)$ holds. From the encryption formula, we have $\vec{c} = B\vec{m} + \vec{e}$. Then,

$$\vec{c} + \vec{s} = B\vec{m} + \vec{e} + \vec{s}$$

$$\vec{c} + \vec{s} \equiv B\vec{m} + \vec{e} + \vec{s} \ (\text{mod } 2\sigma)$$

$$\vec{c} + \vec{s} \equiv B\vec{m} \ (\text{mod } 2\sigma) \tag{1}$$

Basically, the congruence (1) is a linear congruence with single unknown value, $\vec{m}$ and has very few solutions. Assume that the congruence (1) is solved which means the value of $\vec{m} \ (\text{mod } 2\sigma) = \vec{m}_{2\sigma}$ is known. The known $\vec{m}_{2\sigma}$ be inserted into the encryption formula as follows

$$\vec{c} - B\vec{m}_{2\sigma} = B\vec{m} - B\vec{m}_{2\sigma} + \vec{e}$$

$$\vec{c} - B\vec{m}_{2\sigma} = B(\vec{m} - \vec{m}_{2\sigma}) + \vec{e} \tag{2}$$

Note that

$$\vec{m} \equiv \vec{m}_{2\sigma} \ (\text{mod } 2\sigma)$$

which means that

$$\frac{\vec{m} - \vec{m}_{2\sigma}}{2\sigma} = \vec{m}' \in \mathbb{Z}^n$$

$$\vec{m} - \vec{m}_{2\sigma} = 2\sigma\vec{m}' \tag{3}$$

By inserting equation (3) into equation (2), we have

$$\vec{c} - B\vec{m}_{2\sigma} = 2\sigma B\vec{m}' + \vec{e}$$

$$\frac{\vec{c} - B\vec{m}_{2\sigma}}{2\sigma} = \frac{2\sigma B\vec{m}'}{2\sigma} + \frac{\vec{e}}{2\sigma}$$

$$\frac{\vec{c} - B\vec{m}_{2\sigma}}{2\sigma} = B\vec{m}' + \frac{\vec{e}}{2\sigma} \tag{4}$$

All values in the left side of the equation (4), are known. Therefore, the equation (4) is a new CVP instance where the left side of the equation is a known non-lattice vector, $B\vec{m}'$ is an unknown lattice vector, and the right-most of the equation is a new error vector. Since $\vec{e} \in \{\pm\sigma\}^n$, therefore

$$\frac{\vec{e}}{2\sigma} = \pm\frac{\sigma}{2\sigma} = \pm\frac{1}{2}$$

which indicates that the new error vector is $\frac{\vec{e}}{2\sigma} \in \left\{\pm\frac{1}{2}\right\}^n$.

The new error vector is much smaller compared to previous error vector. This makes the new CVP instance much easier to solve. Furthermore, Nguyen used the embedding technique to reduce the new CVP instance to SVP. In the embedding technique, a new $n+1$-dimensional basis B' will be formed as $B' = \begin{pmatrix} \vec{c} & \vec{b}_1 & \cdots & \vec{b}_n \\ 1 & 0 & \cdots & 0 \end{pmatrix}$.

From this basis, a new lattice $\mathcal{L}(B')$ can be generated with dimension almost the same with the lattice $\mathcal{L}(B)$. Increasing the length of the shortest vector of $\mathcal{L}(B)$ makes the lattice gap of $\mathcal{L}(B')$ larger, which results in an easier lattice reduction (Lee and Hahn, 2010). With this large lattice gap, Nguyen used lattice reduction algorithms such as the LLL, BKZ and pruning technique to solve the SVP and eventually solve the Internet GGH Challenges for lattice dimension $n = 200{,}250{,}300, 350$.

In (Goldreich *et. al.*, 1997), the authors had earlier assessed the security of the GGH Scheme against the embedding attack. They showed that the attack failed for dimension larger than 120. However, with simpler CVP instance Nguyen can launch the same attack even for larger lattice dimensions. That means, Nguyen's attack succeeds only when the CVP instance is being simplified into its easier form.

If we can prevent the simplification process, then the embedding attack should be conjectured as fail as well. To do so, we need to ensure that the partial information $\vec{m}_{2\sigma}$ could not be derived from the ciphertext $\vec{c}$ as done by Nguyen. The structure of the error $\vec{e}$ should be modified by making the entries no longer uniformly distributed as $\{\pm\sigma\}^n$. For that purpose, Nguyen proposed a remedy to his own attack by using an error $\vec{e} \in \{\pm\sigma, \pm(\sigma-1)\ \}^n$ which could avoid modulo reduction on the encryption formula (Nguyen,1999). However, he conjectured that this modification is still insecure since the structure of $\vec{e}$ is still in a particular form which could be exploited by other kind of attack. His conjecture was true.

In 2010, Lee and Hahn showed that the improvement proposed by Nguyen is insecure. Not only that, they also have successfully solved the last Internet GGH Challenge for $n = 400$ by combining their technique with Nguyen's attack. We refer the attack as Lee-Hahn's attack (Lee and Hahn, 2010). The Nguyen's attack was launched by reducing the size of the error $\vec{e}$ for simplifying the underlying CVP instance. Lee and Hahn used different approach by enlarging the error $\vec{e}$ for the same purpose as well. Besides that, the Lee-Hahn's attack also requires some knowledge on the secret message $\vec{m}$ to succeed. The more the information they gain, than the more effective their attack can perform.

Without loss of generality, assume that the first k entries of the message $\vec{m}$ are known. The message $\vec{m}$ now represented as $\vec{m} = \begin{pmatrix} \vec{m}_1 \\ \vec{m}_2 \end{pmatrix} \in \mathbb{Z}^n$ where $\vec{m}_1$ represents the known first k entries and $\vec{m}_2$ represents the remaining unknown entries. Similarly, the public basis B also has a new representation $B = (B_1 \quad B_2)$ where $\vec{b}_i \in B_1$ for $i = 1, \dots, k$ and $\vec{b}_i \in B_2$ for $i = k+1, \dots, n$. From the encryption formula, we have

$$\vec{c} = B\vec{m} + \vec{e}$$

$$\vec{c} = (B_1 \quad B_2)\begin{pmatrix} \vec{m}_1 \\ \vec{m}_2 \end{pmatrix} + \vec{e}$$

$$\vec{c} = B_1\vec{m}_1 + B_2\vec{m}_2 + \vec{e}$$

$$\vec{c} - B_1\vec{m}_1 = B_2\vec{m}_2 + \vec{e} \tag{5}$$

Since the message $\vec{m}_1$ is assumed as known, then the left side of the equation (5) is known. Therefore, the equation (5) is a new CVP instance with the same error $\vec{e}$ but with different lattice vector $B_2\vec{m}_2$ which is obviously smaller than the lattice vector $\vec{v} = B\vec{m}$ in the GGH Scheme. In this case, $\mathcal{L}(B_2)$ is a sublattice of the lattice $\mathcal{L}(B)$ since the basis B_2 is derived from B and contains $(n - k)$ basis vectors. The shortest vector in $\mathcal{L}(B_2)$ also could be larger than the shortest vector in $\mathcal{L}(B)$, which increases the lattice gap in the corresponding SVP instance. As done by Nguyen, the embedding technique can be used to reduce the new CVP instance to SVP and later can be solved by using lattice reduction methods. The solution will reveal the value of $\vec{m}_2$

and later can be combined with the known $\vec{m}_1$ to form the whole secret message $\vec{m}$.

To solve the challenge for $n = 400$, Lee and Hahn combined their technique with Nguyen's attack. In line with the Nguyen's attack, assume that

$$\vec{c} + \vec{s} \equiv B\vec{m} \ (\text{mod } 2\sigma)$$

is solved and the value of $\vec{m}\ (mod\ 2\sigma) = \vec{m}_{2\sigma}$ is known. The known $\vec{m}_{2\sigma}$ can be represented as $\vec{m}_{2\sigma} = \begin{pmatrix} \vec{m}_{2\sigma,1} \\ \vec{m}_{2\sigma,2} \end{pmatrix} \in \mathbb{Z}^n$ and inserted into the equation (5) as,

$$\vec{c} - B_1\vec{m}_1 - B_2\vec{m}_{2\sigma,2} = B_2\vec{m}_2 - B_2\vec{m}_{2\sigma,2} + \vec{e}$$
$$\vec{c} - B_1\vec{m}_1 - B_2\vec{m}_{2\sigma,2} = B_2\left(\vec{m}_2 - \vec{m}_{2\sigma,2}\right) + \vec{e} \qquad (6)$$

Dividing equation (6) by 2σ yields

$$\frac{\vec{c} - B_1\vec{m}_1 - B_2\vec{m}_{2\sigma,2}}{2\sigma} = \frac{B_2\left(\vec{m}_2 - \vec{m}_{2\sigma,2}\right)}{2\sigma} + \frac{\vec{e}}{2\sigma} \qquad (7)$$

The values of $\vec{m}_1$ and $\vec{m}_{2\sigma}$ are assumed as known. Therefore, the left-side of the equation (7) is known and eventually makes the equation (7) as a new CVP instance.

Lee and Hahn used the partial decrypted ciphertext by Nguyen for the $n = 400$ GGH challenge as the vector $\vec{m}_1$. In that challenge, the message $\vec{m} \in \mathbb{Z}^{400}$ has integer entries $m_i \in [-128,127]$. Nguyen discovered $\vec{m}_{2\sigma} \in \mathbb{Z}^{400}$ where $2\sigma = 2(3) = 6$ (Nguyen, 1999). The first entry of $\vec{m}_{2\sigma}$ is 5 which indicates that $m_1\ (mod\ 6) = 5$. From the interval $[-128,127]$, there are 43 integers can be the first entry $m_1 \in \vec{m}$ such that m_1 (mod 6) $= 5$. By trying all the possible 43 candidates and applying the Lee-Hahn's attack, the challenge for $n = 400$ has been successfully solved.

It seems the right time to officially declare the GGH Scheme as a dead scheme. However, the general idea behind the scheme is still viable and worth to study for further improvement since the one-way function of the scheme is still merits due to its simplicity and practicality. On top of that, the scheme also considered as the only lattice-based cryptographic scheme that works explicitly with lattices compared to the other lattice-based schemes. Therefore, the remedy to heal the scheme is still worthy to be explored for keeping the scheme alive and survive. The hope is still there.

5 NEW HOPES FOR THE GGH SCHEME

A new hope emerges in 2012, when Yoshino and Kunihiro presented a new design of the GGH Scheme with large error vector (referred to as the GGH-YK Scheme). The security of the scheme is based on a variant of lattice problem which defined as follow (Yoshino and Kunihiro, 2012).

Definition 5.1. For given lattice basis B, a vector $\vec{c}$, set of integers I_1 and I_2, and an integer k, find an error $\vec{e}$ with entries classified as

$$e_i \in I_1 \text{ for } i = 1, \dots, k$$

$$e_i \in I_2 \text{ for } i = k + 1, \dots, n$$

where $\vec{e} = \vec{c} - \vec{v}$ with some lattice vector $\vec{v}$.

This indicates that the entries of the error $\vec{e}$ are no longer selected from the small set $\{-\sigma, \sigma\}$ where σ is the perturbation parameter. From the given ciphertext $\vec{c}$ and lattice vector $\vec{v}$, the task of this problem is to find the whole entries of the error $\vec{e}$. To guarantee that the decryption succeeds, the GGH Scheme uses the property of a rounding vector $\vec{w} = \lfloor R^{-1}\vec{e} \rceil = \vec{0}$ which implies that all elements of $\vec{e}$ are short. In the GGH-YK Scheme, the rounding vector $\vec{w} = \lfloor R^{-1}\vec{e} \rceil \neq \vec{0}$. The entries of $\vec{w}$ are classified as follows

$$w_i = \begin{cases} 0 \text{ for at least } (n - k\ ) \text{ values of } i, \\ 1 \text{ or } -1 \text{ for at most } k \text{ values of } i \end{cases}$$

for $i = 1, \dots, n$ and $k \in \mathbb{Z}^+_{<n}$. If $\|\vec{w}\| > 0$ holds, the new error $\vec{e}$ is thus expected to be larger than the original error $\vec{e}$ in GGH Scheme.

Algorithm 2. The GGH-YK Scheme (*Yoshino and Kunihiro, 2012*).

Parameter Setup by Alice

Sets the value of the following parameters,

i) Lattice dimension n.

ii) Private key parameters $\gamma \in \mathbb{Z}$ and $\gamma > n$.

iii) Public parameters (σ, h, k), with $h > \sigma$ and σ is an even number. The selected parameters must satisfy the following conditions:

Condition 1:$\frac{\sigma}{\gamma} + \frac{2kh}{\gamma^2} + \frac{2n\sigma}{\gamma^2} < \frac{1}{2}$

Condition 2:$\frac{h-\sigma}{\gamma} + \frac{2h}{\gamma^2} < 1$

Condition 3:$2k + 2h < \gamma$

Key Generation by Alice

i) Generates an n-by-n perturbation matrix P with entries $p_{i,j} \in \{-1,0,1\}$.

ii) Computes a private basis R as $R = \gamma I + P$ where I is an n-by-n identity matrix.

iii) Computes $Q = R^{-1}$ with entries must satisfy the following conditions,

Condition 4: For diagonal entries where $i = j, |q_{i,j}| \leq \frac{1}{\gamma}$.

Condition 5: For non-diagonal entries where $i \neq j, |q_{i,j}| \leq \frac{2}{\gamma^2}$.

iv) Generate a public basis B as $B = HNF(R)$ such that $\mathcal{L}(R) = L = \mathcal{L}(B)$.

v) Sends the public basis B with the public parameters $\{n, \sigma\}$ to Bob.

vi) Keeps the other parameters with the private basis R secretly.

Encryption by Bob

i) Sets the message as a binary set $\{0,1\}^l$ where $l \leq n$.

ii) Randomly chooses two secret sets $S, T \subset \{1, \dots, n\}$ such that $|S| = k, |T| = n - l$ and $S \cap T = \emptyset$.

iii) The bits of the message $\{0,1\}^l$, are encoded into the non-zero entries of $\vec{e}$ with entries generated according to the following rules:

a. if $i \in S$, then $e_i = \pm h$

b. if $i \in \{1, \dots, n\} \backslash (S \cup T)$, then $e_i \in \{-\sigma, \dots, -1\} \cup \{1, \dots, \sigma\}$

c. if $i \in T$, then $e_i = 0$

iv) Computes the ciphertext $\vec{c} \in \mathbb{Z}^n$ as $\vec{c} = B\vec{x} + \vec{e}$ where $\vec{x} = -\lfloor B^{-1}\vec{e} \rceil$.

v) Sends the ciphertext $\vec{c}$ to Alice.

Decryption by Alice

i) Computes $\vec{u} = R^{-1}\vec{c} - \lfloor R^{-1}\vec{c} \rceil$
ii) Computes $\vec{e}' = R\vec{u}$
iii) Determine the entries of the rounding vector $\vec{w} = \lfloor R^{-1}\vec{e} \rceil$ as follows:
   a. If $e_i' < -h - k$, set $w_i = 1$,
   b. If $e_i' > h + k$, set $w_i = -1$,
   c. Otherwise, set $w_i = 0$.
iv) Compute the error $\vec{e}$ as $\vec{e} = \vec{e}' + R\vec{w}$ and decode the message bits.

From efficiency perspective, the decryption is very efficient since it is done without using the Babai's round off method. Moreover, the process is deterministic without any decryption error compared to GGH Scheme where the decryption is still done in probabilistic way. From security perspective, the scheme is claimed as resistant to lattice reduction attacks. The used underlying lattice-problem instance is conjectured as harder to solve.

After implementing the GGH-YK Scheme, Barros and Schechter in 2014 realized that the generation of realistic parameters in the GGH-YK Scheme is almost impossible. On top of that, they also noticed that the scheme is still behave in the same way as the GGH Scheme since the rounding vector $\vec{w}$ is always a null vector, $\vec{0}$. Inspired by these issues, they proposed an improvement on the GGH-YK Scheme, we refer as GGH-YK-M Scheme.

Algorithm 3. The GGH-YK-M Scheme (*de Barros and Schechter, 2014*).

Parameter Setup by Alice

Sets the value of the following parameters,

i) Lattice dimension n.
ii) Private key parameter $\gamma \in \mathbb{Z}$ and $\gamma > \rho P$ where
$$\rho P = \max\{|\lambda_i| : \lambda_i \text{ is an eigenvalue of } P\}$$
is the spectral radius of the perturbation matrix P.
iii) Public parameters (σ, h, k), with $h > \sigma$ and σ is an even number. The selected parameters must satisfy the following conditions,

Condition 1: $\frac{2\sigma}{\gamma} + \frac{2kh}{\gamma^2} + \frac{2n\sigma}{\gamma^2} < \frac{1}{2} + \frac{2\sigma(k+1)}{\gamma^2}$

Condition 2: $2(h - \sigma)\left(\frac{1}{\gamma} - \frac{1}{\gamma^2}\right) < 1$

Condition 3: $\frac{h}{\gamma} > \frac{1}{2}$

Condition 4: $h + k < \gamma$

Key Generation by Alice

i) Generates an n-by-n perturbation matrix P with entries $p_{i,j} \in \{-1,0\}$

ii) Computes $R = \gamma I + P$ where I is an n-by-n identity matrix.

iii) Computes $Q = R^{-1}$ with entries must satisfy the following conditions,

Condition 5: For diagonal entries where $i = j$,

$$\frac{1}{\gamma} < |q_{i,j}| \leq \frac{2}{\gamma}$$

Condition 6: For non-diagonal entries where $i \neq j$,

$$|q_{i,j}| \leq \frac{2}{\gamma^2}$$

iv) Generate a public basis B as $B = HNF(R)$ such that,

$$\mathcal{L}(R) = L = \mathcal{L}(B)$$

v) Sends the public basis B with the public parameters $\{n, \sigma\}$ to Bob.

vi) Keeps the other parameters with the private basis R secretly.

Encryption by Bob

i) Set the message as a binary set $\{0,1\}^l$ where $l \leq n$.

ii) Randomly choose a secret set $S \subset \{1, \dots, n\}$.

iii) The bits of the message $\{0,1\}^l$, are encoded into the non-zero entries of $\vec{e}$ with entries generated according to the following rules:

a. The "0" bits are encoded as entries randomly chosen from the integer set $\left\{1, \dots, \frac{\sigma}{2}\right\}$

b. The "1" bits are encoded as entries randomly chosen from the integer set $\left\{\frac{\sigma}{2} + 1, \dots, \sigma\right\}$.

iv) Computes the ciphertext $\vec{c} \in \mathbb{Z}^n$ as $\vec{c} = B\vec{x} + \vec{e}$ where $\vec{x} = -\lfloor B^{-1}\vec{e} \rceil$.

v) Sends the ciphertext $\vec{c}$ to Alice.

Decryption by Alice

i) Computes $\vec{u} = R^{-1}\vec{c} - \lfloor R^{-1}\vec{c} \rceil$

ii) Computes $\vec{e}' = R\vec{u}$

iii) Determine the entries of the rounding vector $\vec{w} = \lfloor R^{-1}\vec{e} \rceil$ as follows:

a. If $e_i' < 0$, set $w_i = 1$,

b. Otherwise, set $w_i = 0$.

iv) Compute the error $\vec{e}$ as $\vec{e} = \vec{e}' + R\vec{w}$ and decode the message bits.

The letter M in the name of GGH-YK-M Scheme is come from the M-matrix that being used as the private key R. The inverse of this M-matrix has only positive entries which is expected to be advantageous in the efficiency of decryption process. The major improvement proposed by Barros and

Schechter was on the parameter generation processes where some conditions have been changed, maintained and there are new conditions have been introduced. The security of the GGH-YK-M Scheme has been assessed by launching lattice reduction attacks using LLL and BKZ algorithms. Experimentally, these attacks failed for lattice dimension more than 300.

6 DISCUSSIONS

The effectiveness of the Nguyen and Lee-Hahn attacks are based on the same concept, which is simplifying the underlying CVP instance to easier instance then use the embedding technique to reduce the simplified instance to SVP. With large lattice gaps, lattice reduction methods can efficiently solve the SVP and eventually solve the easier CVP instance. If the underlying CVP instance can be prevented from being simplified, then both attacks may be bypassed.

In this section, we evaluate the GGH-YK-M Scheme whether it is secure against the Nguyen and Lee-Hahn attacks or not. From the encryption formula, we have $\vec{c} = B\vec{x} + \vec{e}$. In this scheme, the message bits are encoded in the vector $\vec{e}$. Thus, the scheme can be considered broken if we can reveal the vectors $\vec{e}$ or $\vec{x}$ where $\vec{x} = -\lfloor B^{-1}\vec{e} \rceil$. In Nguyen's attack, the vector $\vec{e}$ can be eliminated by using modulo reduction on the encryption formula. We try do the same task on the GGH-YK-M Scheme. By setting a vector $\vec{s} \in \{\sigma\}^n$, the following congruence

$$\vec{e} + \vec{s} \equiv \vec{0} (\text{mod } 2\sigma) \tag{8}$$

does not hold since the vector $\vec{e}$ now no longer uniformly selected from the set $\{\pm\sigma\}$. The entries of $\vec{e}$ now generated from the integer set $\{1, \ldots, \sigma\}$ where the parameter σ is an even number. For large dimension, it is computationally inefficient to try all possible entries of the vector $\vec{e}$ exhaustively, but it is still possible. Thus, we assume that the congruence (8) holds, then we have

$$\vec{c} + \vec{s} \equiv B\vec{x} \ (\text{mod } 2\sigma) \tag{9}$$

where $\vec{x}$ is the unknown vector. Solving this congruence yields the vector $\vec{x}_{2\sigma}$ and be inserted into the encryption formula as follows

$$\vec{c} - B\vec{x}_{2\sigma} = B\vec{x} - B\vec{x}_{2\sigma} + \vec{e}$$

$$\vec{c} - B\vec{x}_{2\sigma} = B(\vec{x} - \vec{x}_{2\sigma}) + \vec{e} \tag{10}$$

Since $\vec{x} \equiv \vec{x}_{2\sigma}$ (mod 2σ), then we have

$$\vec{x} - \vec{x}_{2\sigma} = 2\sigma\vec{x}' \in \mathbb{Z}^n \tag{11}$$

By inserting equation (11) into equation (10), we have

$$\vec{c} - B\vec{x}_{2\sigma} = 2\sigma B\vec{x}' + \vec{e}$$

$$\frac{\vec{c} - B\vec{x}_{2\sigma}}{2\sigma} = B\vec{x}' + \frac{\vec{e}}{2\sigma} \tag{12}$$

All values in the left side of the equation (12), are known. Therefore, the equation (12) is a new CVP instance where the left side of the equation is a known non-lattice vector, $B\vec{x}'$ is an unknown lattice vector, and the right-most of the equation is a new error vector. But then, the question now is back to the entries of the vector $\vec{e}$ which are now generated from the integer set $\{1, \ldots, \sigma\}$, and no longer uniformly selected from the small set $\{\pm\sigma\}$. Clearly, the GGH-YK-M Scheme has successfully eliminated the particular form of the vector $\vec{e}$ which was previously exploited by Nguyen to simplify the underlying CVP instance. At this point, we consider that the Nguyen's attack has been bypassed by the GGH-YK-M Scheme.

For the Lee-Hahn attack, assume that the first k entries of the vector $\vec{x}$ are known. The vector $\vec{x}$ now represented as

$$\vec{x} = \begin{pmatrix} \vec{x}_1 \\ \vec{x}_2 \end{pmatrix} \in \mathbb{Z}^n$$

where $\vec{x}_1$ represents the known first k entries and $\vec{x}_2$ represents the remaining unknown entries. Similarly, the public basis B also has a new representation,

$$B = (B_1 \quad B_2)$$

where $\vec{b}_i \in B_1$ for $i = 1, \ldots, k$ and $\vec{b}_i \in B_2$ for $i = k + 1, \ldots, n$. From the encryption formula, we have

$$\vec{c} = B\vec{x} + \vec{e}$$

$$\vec{c} = (B_1 \quad B_2)\begin{pmatrix}\vec{x}_1\\ \vec{x}_2\end{pmatrix} + \vec{e}$$
$$\vec{c} = B_1\vec{x}_1 + B_2\vec{x}_2 + \vec{e}$$
$$\vec{c} - B_1\vec{x}_1 = B_2\vec{x}_2 + \vec{e} \tag{13}$$

Since the message $\vec{x}_1$ is assumed as known, then the left side of the equation (13) is known. Therefore, the equation (13) is a new CVP instance with the same vector $\vec{e}$ but with different lattice vector $B_2\vec{x}_2$ which is obviously smaller than the lattice vector $B\vec{x}$ in the encryption formula. Note that, the Lee-Hahn attack succeed since Nguyen previously solved the Internet GGH Challenge for dimension $n = 400$. They used the partial solution to run their attack to fully solve the challenge for dimension $n = 400$. Without the partial solution provided by Nguyen, their attack could fail. The same case with the GGH-YK-M Scheme as well. If the partial information cannot be gained from the vector $\vec{x}$, the attack also fails to proceed. To reveal the vector $\vec{x}$ is equivalent as revealing the vector $\vec{e}$ since $\vec{x} = -\lfloor B^{-1}\vec{e}\rceil$. Therefore, we conclude that the GGH-YK-M Scheme also bypassed the Lee-Hahn's attack.

7 CONCLUSION AND FUTURE WORKS

Is the GGH-YK-M Scheme can be considered as the secure version of the GGH Scheme? The accurate answer is remains unknown until a thorough analysis on the security of the scheme being carried out which is not the aim of this paper. We put this as task as one of our future works. From literature, the scheme is remains untouched by any notable attack like Nguyen and Lee-Hahn attacks. We consider that the GGH Scheme as survive since the current variant of the scheme, which are the GGH-YK and GGH-YK-M schemes can bypass the Nguyen and Lee-Hahn attacks. However, there are some important issues surrounding the scheme. The public and private bases are still in matrices form. For large lattice dimension, their sizes still give significant effect to the efficiency and practicality the scheme.

In addition, the public basis B is in Hermite Normal Form (HNF) as deployed by Micciancio in his GGH-Micciancio Scheme (Micciancio, 2001).

Even though the HNF is easier to store and harder to reduce, it comes with additional cost in transforming the private basis R into its HNF, especially in terms of space complexity. Moreover, the simple formula of the encryption process is still exposed to security threats like Nguyen and Lee-Hahn attacks. Perhaps, some better formulization is required specially to prevent the CVP instance from being simplified. Furthermore, the encoding of the message into the vector $\vec{e}$ is done bit-by-bit. For large capacity of data, this process could be time consuming. Last but not least, the underlying lattice-problem of the GGH-YK-M Scheme also need to be properly formulized and assessed. We will address all these issues as our future works to improve the scheme for making it better, stronger, and ideal for wide adoption in post quantum era.

REFERENCES

Ajtai, M., & Dwork, C. (1997). A Public-key Cryptosystem with Worst-case/Average-case Equivalence. Paper presented at the *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*.

de Barros, C. F., & Schechter, L. M. (2015). GGH May Not Be Dead After All. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, 3*(1).

Goldreich, O., Goldwasser, S., & Halevi, S. (1997). Public-key cryptosystems from lattice reduction problems. Paper presented at the Advances in Cryptology-CRYPTO'97: *17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 1997. Proceedings*.

Goldreich, O., Goldwasser, S., & Halevi, S. (1997). Challenges for the GGH-Cryptosystem.
Available at http://theory.lcs.mit.edu/~shaih/challenge.html

Hoffstein, J., Pipher, J., & Silverman, J. H. (2008). Lattices and Cryptography. In S. Axler & K. A. Ribet (Eds.), *An Introduction to Mathematical*

Cryptography (pp. 349-422). Spring Street, New York: Springer Science+Business Media, LLC.

Lee, M. S. & Hahn, S. G. (2010). Cryptanalysis of the GGH Cryptosystem. *Mathematics in Computer Science* Vol 3(2): pp 201-208.

McEliece, R. J. (1978). A Public-key Cryptosystem based on Algebraic Coding Theory. The Deep Space Network Progress Report, DSN PR 42-44: pp. 114-116.

Micciancio, D. (2001). Improving Lattice based Cryptosystems using the Hermite Normal Form. *Cryptography and Lattices*. Springer: pp. 126-145.

Nguyen P., Stern J. (1998) Cryptanalysis of the Ajtai-Dwork Cryptosystem. In: Krawczyk H. (eds) Advances in Cryptology – CRYPTO '98. CRYPTO 1998. *Lecture Notes in Computer Science, vol 1462. Springer, Berlin, Heidelberg*.

Nguyen P. (1999) Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97. In: Wiener M. (eds) Advances in Cryptology – CRYPTO '99. CRYPTO 1999. *Lecture Notes in Computer Science, vol 1666. Springer, Berlin, Heidelberg*.

Yoshino, M., & Kunihiro, N. (2012). Improving GGH cryptosystem for large error vector. Paper presented at the *Information Theory and its Applications (ISITA), 2012 International Symposium on Information Theory and its Applications, Honolulu, Hawaii Island, USA.*