Malaysian Journal of Mathematical Sciences

Journal homepage: https://einspem.upm.edu.my/journal

An Upgrade on the Key Generation Algorithm of the GGH-MKA Lattice-Based Encryption Scheme

Mandangan, A.[1], Kamarulhaili, H.[2], and Asbullah, M. A. *[3]

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

E-mail: ma_asyraf@upm.edu.my
* *Corresponding author*

Received: 30 June 2021
Accepted: 8 October 2021

Abstract

This paper presents an upgrade on the key generation algorithm of a current variant of the Goldreich-Goldwasser-Halevi lattice-based encryption scheme, referred to as the GGH-MKA cryptosystem. The keys for this cryptosystem consisting of lattice bases where the private key is required to be a 'good' basis while the public key is required to be a 'bad' basis to ensure the cryptosystem works effectively. In the key generation algorithm of the GGH cryptosystem, the good and bad features of the lattice bases are measured by computing orthogonality-defect value. If the value is 'close to 1', the basis is considered as a good basis. On the contrary, the basis is considered as a bad basis if its orthogonality-defect value is 'far from 1'. Clearly, the consideration on various subjective terms could potentially trigger technical error during the key generation processes. In this paper, we proposed new conditions on the private and public bases of the GGH-MKA cryptosystem. Instead of depending solely on the orthogonality-defect values, the proposed conditions could make the measurement of good and bad bases in the key generation algorithm of the GGH-MKA cryptosystem becomes clearer and deterministic.

Keywords: closest-vector problem; smallest-basis problem; lattice; good basis; bad basis; GGH cryptosystem; lattice-based cryptography.

1 Introduction

Post-quantum cryptography (PQC) is a new direction of modern cryptography to cope with the emergence of quantum-based attacks. With the ability to solve hard computational problems underlying number theoretical-based cryptosystems, Shor's quantum algorithm [18] potentially made the Rivest-Shamir-Adleman (RSA) [16], El-Gamal [3] and Elliptic Curve [8] cryptosystems considered broken. One of the celebrated alternatives in the PQC is lattice-based cryptography which deploys lattice-based computational problems such as the Closest-Vector Problems (CVP) [12], Shortest-Vector Problems (SVP) [1], Smallest-Basis Problems (SBP) [7], and some variants of these problems as security backbone.

Since the 1990s, several lattice-based cryptosystems have been proposed with various security and efficiency features. For instance, the NTRU, which was initially published in [6]. The system continues to evolve as people release new versions and variations intended to make it more secure against attacks. One of NTRU's central claims to fame over systems like RSA is that NTRU is not known to be vulnerable against attacks mounted by quantum computers and is thus being studied for its application in a post-quantum-computer world. Another earlier significant lattice-based cryptosystem is the Goldreich-Goldwasser-Halevi encryption scheme commonly referred to as the GGH cryptosystem [5] was the first method judged viable. It was conjectured that the lattice problems underlying the GGH cryptosystem, known as GGH-CVP and GGH-SBP, were invulnerable in lattice dimensions 300 and beyond. However, a significant flaw in its design has allowed the simplification of the GGH-CVP. Consequently, embedding-based attacks launched by Nguyen [14] made the GGH cryptosystem in lattice dimensions of 200 up to 350 broken. Later, Lee and Hahn [9] launched another embedding-based attack and made the GGH cryptosystem in lattice dimension of up to 500 broken.

Various attempts have been proposed to make the GGH cryptosystem survives. Some of these attempts improved the efficiency of the GGH cryptosystem to make it remains practical in lattice dimensions beyond 500 where the Nguyen and Lee-Hahn embedding attacks defective as done in [13] and [15]. However, the flaw of the GGH cryptosystem remains unrepaired. It could be re-exploited by other embedding-based attacks in the future [10]. There are other attempts to repair the exploited flaw proposing some variants of GGH, such as the GGH-YK [19] and GGH-YK-M [2] cryptosystems. Despite repairing the exploited flaw, these variants majorly modified the original design of the GGH cryptosystem. On top of that, the security of these variants relies on new problems instead of the original GGH-CVP.

Recently, a new variant of the GGH cryptosystem is proposed in [11]. This variant is referred to as the GGH-MKA cryptosystem. Through minor modification on the original design of the GGH cryptosystem, this variant repaired the exploited flaw by the Nguyen's and Lee-Hahn's embedding attacks. This variant also maintains the security dependency on the GGH-CVP. By introducing a new set of entries for the error vector $\vec{e}$ and its distribution guideline, the simplification of the GGH-CVP as done by Nguyen's embedding attacks could be completely prevented. At the same time, the GGH-CVP distance also could be maintained as $\sigma\sqrt{n}$.

Security of the GGH-MKA cryptosystem relies on the hardness of the GGH-CVP and GGH-SBP. In both problems, quality measurement of the lattice bases is crucial. The private basis is required to be a good basis to ensure the correctness of the decryption. On the other hand, the public basis is required to be a bad basis to prevent attacks from an adversary. If the private basis is not good enough, then decryption would not recover the original message. If the public basis is not bad enough, it could be used for unauthorized decryption purposes by the adversary [11]. In the key generation algorithm of the GGH cryptosystem, the quality measurement of lattice bases

depends solely on the orthogonality-defect value of the bases. If the value is close to 1, the basis is considered as a good basis. On the contrary, it is considered a bad basis if the orthogonality-defect value is far from 1. The consideration for 'close to 1' and 'far from 1' is still too general and subjective. A technical error might occur due to confusion on these subjective terms. Any mistake in the key generation algorithm could trigger decryption error and security breach.

In this paper, we propose an upgrade on the key generation algorithm of the GGH-MKA cryptosystem by introducing new conditions on the generated bases. Instead of depending solely on the orthogonality-defect value, these conditions can be considered to ensure that the private basis would make decryption succeed without an error, and unauthorized decryption using the public basis would undoubtedly fail. Through the proposed upgrade, quality measurement of the private and public bases could be done deterministically. Clearer and deterministic mechanisms are demanded, making the key generation algorithm of the GGH-MKA cryptosystem free from any technical error.

This paper is organized in the following outline. Some mathematical foundations are provided in the next section. In Section 3, key generation, encryption, and decryption algorithms of the GGH-MKA cryptosystem are presented. Then, the proposed conditions on the private and public bases are presented in Section 4, and mathematical proofs to justify our proposal. Finally, this paper is concluded in Section 5.

2 Mathematical Foundation

Consider $m, n \in \mathbb{N}$. Entries of a vector $\vec{g} \in \mathbb{R}^m$ are represented as $\vec{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix}$ where $g_1, g_2, \ldots, g_m \in \mathbb{R}$. A set containing n vectors, $G = \{\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n\}$ where $\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n \in \mathbb{R}^m$ can be represented in matrix form, $G \in \mathbb{R}^{m \times n}$ as $G = \begin{bmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,n} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m,1} & g_{m,2} & \cdots & g_{m,n} \end{bmatrix}$. Consider the following definition:

Definition 2.1. [7] *Let $G = \{\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n\}$ where $\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n \in \mathbb{R}^m$. Then G is considered linearly independent if the only way to make the equation $\sum_{i=1}^{n} a_i \vec{g}_i = \vec{0}$ holds is when $a_i = 0$ for all $i = 1, 2, \ldots, n$. Otherwise, G is considered linearly dependent.*

Linearly independent G can be used to generate lattice $\mathcal{L}$, denoted as $L(G) = \mathcal{L}$. It is defined as follows:

Definition 2.2. [4] *Let $G = \{\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n\}$ where $\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n \in \mathbb{R}^m$ be linearly independent with $m \geq n$. A lattice $\mathcal{L} \subset \mathbb{R}^m$ that is spanned by G, denoted as $L(G) = \mathcal{L}$ is defined as,*

$$L(G) = \left\{ \vec{v} = \sum_{i=1}^{n} a_i \vec{g}_i \;\middle|\; \vec{g}_i \in G \text{ and } a_i \in \mathbb{Z}, \forall i = 1, 2, \ldots, n \right\}. \tag{1}$$

The set G that spans the lattice $\mathcal{L}$ is called lattice basis and the vectors $\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n \in G$ are called basis vectors. Since $G \in \mathbb{R}^{m \times n}$, the lattice $L(G)$ as defined in Definition 2.2 has dimension,

$dim(L(G)) = n$ and rank, $rank(L(G)) = m$. When $m = n$, the lattice $L(G)$ is referred to as full rank lattice. Beyond this point, all considered lattices are full-rank lattices unless stated otherwise.

For $n \geq 2$, the lattice $\mathcal{L}$ can be spanned by infinitely many bases. These bases are mathematically related as follows:

Proposition 2.1. [4] *Let $G, B \in \mathbb{R}^{n\times n}$ be linearly independent with columns $\vec{g}_i, \vec{b}_i \in \mathbb{R}^n$ respectively $\forall i = 1, \ldots, n$ and $U \in \mathbb{Z}^{n\times n}$ be a unimodular matrix with $det(U) = \pm 1$. If $G = BU$, then $L(G) = L(B) = \mathcal{L} \subset \mathbb{R}^n$.*

Proposition 2.2. [7] *Suppose that $G, B \in \mathbb{R}^{n\times n}$ be the bases of the lattice $\mathcal{L} \subset \mathbb{R}^n$ where $L(G) = L(B) = \mathcal{L}$. The value of the $det(\mathcal{L})$ is an invariant, i.e., $det(\mathcal{L}) = det\,(L(G)) = det\,(L(B))$ where $det\,(L(G)) = |det\,(G)\,|$ and $det\,(L(B)) = |det\,(B)\,|$.*

Quality of a lattice basis is determined by the norm (length) and orthogonality (angle) of its basis vectors.

Definition 2.3. [17] *Let $\vec{b} \in \mathbb{R}^n$. The Euclidean norm of the vector $\vec{b}$ is computed as $\|\vec{b}\| = \sqrt{\sum_{i=1}^{n} b_i^2}$ where $b_i \in \vec{b}$ for all $i = 1, 2, \ldots, n$.*

Definition 2.4. [7] *Let $G = \{\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n\}$ where $\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n \in \mathbb{R}^n$ be a basis for the lattice $L(G) = \mathcal{L} \subset \mathbb{R}^n$. For $k, l \in \mathbb{N}$ and $k, l \in [1, n]$, the dot product of $\vec{g}_k, \vec{g}_l \in G$ is computed as $\vec{g}_k \cdot \vec{g}_l = \sum_{i=1}^{n} g_{i,k} \cdot g_{i,l}$ where $g_{i,k} \in \vec{g}_k$ and $g_{i,l} \in \vec{g}_l$ for all $i = 1, 2, \cdots, n$. If $\vec{g}_k \cdot \vec{g}_l = 0$ for all $k, l \in [1, n]$ and $k \neq l$, then G is considered an orthogonal basis for the lattice $\mathcal{L}$. Otherwise, it is considered as non-orthogonal basis.*

The degree of non-orthogonality of lattice basis can be measured by computing the orthogonality-defect value of the basis as follows:

Definition 2.5. [5] *Let $G \in \mathbb{R}^{n\times n}$ with columns $\vec{g}_i \in \mathbb{R}^n$ for all $i = 1, 2, \cdots, n$ be a basis for the lattice $L(G) = \mathcal{L} \subset \mathbb{R}^n$. The orthogonality defect of the basis G is computed as,*

$$orth_{def}(G) = \frac{\prod_{i=1}^{n} \|\vec{g}_i\|}{|det\,(G)\,|}. \tag{2}$$

If the basis vectors $\vec{g}_1, \vec{g}_2, \ldots, \vec{g}_n$ are orthogonal to each other, then $orth_{def}(G) = 1$. Otherwise, $orth_{def}(G) > 1$.

3 GGH-MKA Cryptosystem

Key generation, encryption and decryption algorithms of the GGH-MKA cryptosystem are given as the following [11]. In these algorithms, consider a scenario where Bob wants to send a secret message $\vec{m} \in \mathbb{Z}^n$ to Alice by using the GGH-MKA cryptosystem.

Algorithm 1 Key generation algorithm of the GGH-MKA cryptosystem is done by Alice as the recipient.

Input: Parameter $\delta \in \mathbb{N}$ where $\sigma > 2$.
Output: Public key (B, σ, n), private key (G, U), set of entries E and its distribution guideline.

1: Decide the lattice dimension n as $n = (4\sigma - 2)k$ where $k \in \mathbb{N}$.
2: Generate a good basis $G \in \mathbb{Z}^{n\times n}$ such that $orth_{def}(G) \approx 1$.
3: Generate a unimodular matrix $U \in \mathbb{Z}^{n\times n}$.
4: Compute a bad basis $B \in \mathbb{Z}^{n\times n}$ as $B = GU^{-1}$ and $orth_{def}(B)$ far from 1.
5: Set the set $E = \{(2-\sigma), (1-\sigma), \sigma, (\sigma+1)\}$ and distribution guideline as follows:

$$e_i = \begin{cases} (2-\sigma) & \text{for } \frac{n}{4\sigma-2} \text{ number of entries,} \\ (1-\sigma) & \text{for } \frac{\sigma n - n}{2\sigma-1} \text{ number of entries,} \\ \sigma & \text{for } \frac{n}{4\sigma-2} \text{ number of entries,} \\ (\sigma+1) & \text{for } \frac{\sigma n - n}{2\sigma-1} \text{ number of entries.} \end{cases} \tag{3}$$

6: Keep the private key (G, U) secretly and send the public key (B, σ, n) together with the set E and distribution guideline in equation (3) to the sender.

Algorithm 2 Encryption algorithm of the GGH-MKA cryptosystem is done by Bob as the sender.

Input: Public key (B, σ, n), set of entries E and its distribution guideline.
Output: Ciphertext $\vec{c} \in \mathbb{R}^n$.

1: Generate plaintext $\vec{m} \in \mathbb{Z}^n$.
2: For all $i = 1, \ldots, n$, generate the error vector $\vec{e} \in \mathbb{Z}^n$ with entries $e_i \in \vec{e}$ that are randomly selected from the set $E = \{(2-\sigma), (1-\sigma), \sigma, (\sigma+1)\}$ based on the given distribution guideline in equation (3) from the recipient.
3: Perform the encryption as $\vec{c} = B\vec{m} + \vec{e}$.
4: Send the ciphertext $\vec{c}$ to the recipient and keep the error vector $\vec{e}$ secretly.

Algorithm 3 Decryption algorithm of the GGH-MKA cryptosystem is done by Alice as the recipient.

Input: Ciphertext $\vec{c} \in \mathbb{R}^n$ from the sender and private key (G, U).
Output: Plaintext $\vec{m} \in \mathbb{Z}^n$.

1: Compute vector $\vec{x} \in \mathbb{R}^n$ as $\vec{x} = G^{-1}\vec{c}$.
2: Round each entry $x_i \in \vec{x}$ to form an integer vector $\lfloor \vec{x} \rceil \in \mathbb{Z}^n$ with entries $\lfloor x_i \rceil \in \mathbb{Z}$ such that

$$|x_i - \lfloor x_i \rceil| < \frac{1}{2},$$

for all $i = 1, \ldots, n$.
3: Perform the decryption as $\vec{m} = U\lfloor \vec{x} \rceil$.

Effective decryption yields the original message as proven in the following proof of correctness:

Proposition 3.1. *For $n, \sigma \in \mathbb{N}$, let $G, B \in \mathbb{R}^{n\times n}$ be bases for $\mathcal{L}$, i.e., $L(G) = L(B) = \mathcal{L} \subset \mathbb{R}^n$ such that $G = BU$ where $U \in \mathbb{Z}^{n\times n}$ is a unimodular matrix. Then, let $\vec{c} = B\vec{m} + \vec{e}$ be a ciphertext vector where $\vec{m} \in \mathbb{Z}^n$ is a plaintext vector and $\vec{e} \in \mathbb{Z}^n$ is an error vector. If $\lfloor G^{-1}\vec{e} \rceil = \vec{0}$, then $U\lfloor G^{-1}\vec{c} \rceil = \vec{m}$ which indicates that decryption successful.*

Proof.

$$\begin{aligned} U\lfloor G^{-1}\vec{c}\rceil &= U\lfloor G^{-1}(B\vec{m}+\vec{e})\rceil, since\ \vec{c}=B\vec{m}+\vec{e} \\ &= U\lfloor G^{-1}B\vec{m}+G^{-1}\vec{e}\rceil \\ &= \lfloor UG^{-1}B\vec{m}\rceil + U\lfloor G^{-1}\vec{e}\rceil \\ &= \lfloor B^{-1}GG^{-1}B\vec{m}\rceil + U\lfloor G^{-1}\vec{e}\rceil, since\ U=B^{-1}G \\ &= \lfloor \vec{m}\rceil + U\lfloor G^{-1}\vec{e}\rceil \\ &= \vec{m}+U\lfloor G^{-1}\vec{e}\rceil, since\ \vec{m}\in\mathbb{Z}^n. \end{aligned}$$

Assume that, $\lfloor G^{-1}\vec{e}\rceil = \vec{0}$. Therefore,

$$\begin{aligned} U\lfloor G^{-1}\vec{c}\rceil &= \vec{m}+U(\vec{0}) \\ &= \vec{m}. \end{aligned}$$

□

4 Upgrade on the Key Generation Algorithm of the GGH-MKA Cryptosystem

In the GGH-MKA cryptosystem, the threshold parameter σ is required as $\sigma > 2$ to ensure that the entry $(2-\sigma) \neq 0$ and this allows each of the entries $(2-\sigma), (1-\sigma), \sigma$ and $(\sigma+1)$ appears in the error vector $\vec{e}$. Furthermore, the lattice dimension n is determined as $n = (4\sigma-2)k$ where $\sigma > 2$ and $k \in \mathbb{N}$. Since n represents the dimension of the bases $G, B \in \mathbb{R}^{n\times n}$, then the selection of the threshold parameter σ is done prior the generation of the private basis G. Recall the communication scenario between Alice and Bob in the previous section. In the following discussion, consider Eve as an unauthorized party who wants to recover the secret message $\vec{m} \in \mathbb{Z}^n$ that is sent from Bob to Alice.

4.1 Condition on the Private Basis

To be considered as a private basis, it is required that $orth_{def}(G) \approx 1$ to ensure that G is a good basis. In addition, the generated G is also required to fulfill the condition that the rounded vector $\lfloor G^{-1}\vec{e}\rceil = \vec{0}$ in order to avoid decryption error, as proven in Proposition 3.1. Although the threshold parameter σ belongs to Alice, the arrangement of the entries $(2-\sigma), (1-\sigma), \sigma$ and $(\sigma+1)$ in the error vector $\vec{e}$ is fully determined by Bob. Without knowing the exact entries of the error vector $\vec{e}$, how could Alice check whether the generated G satisfies the condition $\lfloor G^{-1}\vec{e}\rceil = \vec{0}$ or not? To address this issue, consider the following proposition:

Proposition 4.1. *For $n, \sigma, k \in \mathbb{N}$ where $n = (4\sigma-2)k$ and $\sigma > 2$, let $G \in \mathbb{Z}^{n\times n}, \vec{t} \in \{\sigma+1\}^n$ and $\vec{e} \in \mathbb{Z}^n$ where the entries $e_i \in \vec{e}$ are selected randomly from $E = \{(2-\sigma), (1-\sigma), \sigma, (\sigma+1)\}$ based on the distribution guideline in equation (3) for all $i = 1, \ldots, n$. If $\lfloor G^{-1}\vec{t}\rceil = \vec{0}$, then $\lfloor G^{-1}\vec{e}\rceil = \vec{0}$.*

Proof. Denote the entries of the inverse matrix $G^{-1} \in \mathbb{R}^{n\times n}$ as $g'_{i,j} \in G^{-1}$ for all $i, j = 1, \ldots, n$.

Suppose that $\lfloor G^{-1}\vec{t} \rceil = \vec{0}$. Note that,

$$\begin{aligned} G^{-1}\vec{t} &= \begin{bmatrix} g'_{1,1} & g'_{1,2} & \cdots & g'_{1,n} \\ g'_{2,1} & g'_{2,2} & \cdots & g'_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g'_{n,1} & g'_{n,2} & \cdots & g'_{n,n} \end{bmatrix} \begin{bmatrix} \sigma+1 \\ \sigma+1 \\ \vdots \\ \sigma+1 \end{bmatrix} \\ &= \begin{bmatrix} (\sigma+1)\left(g'_{1,1}+g'_{1,2}+\cdots+g'_{1,n}\right) \\ (\sigma+1)\left(g'_{2,1}+g'_{2,2}+\cdots+g'_{2,n}\right) \\ \vdots \\ (\sigma+1)\left(g'_{n,1}+g'_{n,2}+\cdots+g'_{n,n}\right) \end{bmatrix}. \end{aligned}$$

Since $\lfloor G^{-1}\vec{t} \rceil = \vec{0}$, this implies that

$$\left| t_i \left(g'_{i,1}+g'_{i,2}+\cdots+g'_{i,n}\right) \right| < \frac{1}{2},$$

where $t_i \in \vec{t}$ and $g'_{i,j} \in G^{-1}$ for all $i, j = 1, \ldots, n$. On the other hand,

$$G^{-1}\vec{e} = \begin{bmatrix} g'_{1,1} & g'_{1,2} & \cdots & g'_{1,n} \\ g'_{2,1} & g'_{2,2} & \cdots & g'_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ g'_{n,1} & g'_{n,2} & \cdots & g'_{n,n} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} e_1\left(g'_{1,1}+g'_{1,2}+\cdots+g'_{1,n}\right) \\ e_2\left(g'_{2,1}+g'_{2,2}+\cdots+g'_{2,n}\right) \\ \vdots \\ e_n\left(g'_{n,1}+g'_{n,2}+\cdots+g'_{n,n}\right) \end{bmatrix}.$$

For any $\sigma \in \mathbb{N}$ and $\sigma > 2$,

$$|\sigma+1| > |\sigma| > |1-\sigma| > |2-\sigma|.$$

This implies that,

$$\left| t_i \left(g'_{i,1}+g'_{i,2}+\cdots+g'_{i,n}\right) \right| \geq \left| e_i \left(g'_{i,1}+g'_{i,2}+\cdots+g'_{i,n}\right) \right|,$$

for any i-th row of the vectors $\lfloor G^{-1}\vec{t} \rceil$ and $\lfloor G^{-1}\vec{e} \rceil$ for all $i = 1, \ldots, n$. Since

$$\left| t_i \left(g'_{i,1}+g'_{i,2}+\cdots+g'_{i,n}\right) \right| < \frac{1}{2},$$

then

$$\left| e_i \left(g'_{i,1}+g'_{i,2}+\cdots+g'_{i,n}\right) \right| < \frac{1}{2},$$

as well for all $i = 1, \ldots, n$. This implies that, $\lfloor G^{-1}\vec{e} \rceil = \vec{0}$. □

Lemma 4.1. *For $n, \sigma, k \in \mathbb{N}$ where $n = (4\sigma - 2)\,k$ and $\sigma > 2$, let $G, B \in \mathbb{Z}^{n\times n}$ where G and B be bases for the lattice $\mathcal{L} \subset \mathbb{R}^n$ and U is a unimodular matrix such that $G = BU$. Let $\vec{c} = B\vec{m} + \vec{e}$ be a ciphertext vector where $\vec{m} \in \mathbb{Z}^n$ is a plaintext vector and $\vec{e} \in \mathbb{Z}^n$ is an error vector with entries $e_i \in \vec{e}$ are randomly selected from $E = \{(2-\sigma), (1-\sigma), \sigma, (\sigma+1)\}$ based on the distributions guideline in equation (3) for all $i = 1, \ldots, n$. If $\lfloor G^{-1}\vec{t} \rceil = \vec{0}$, then $U\lfloor G^{-1}\vec{c} \rceil = \vec{m}$ where $\vec{t} \in \{\sigma+1\}^n$.*

Proof. Given that $G = BU, \vec{c} = B\vec{m} + \vec{e}$ and $\vec{m} \in \mathbb{Z}^n$. Thus,

$$\begin{aligned} U\lfloor G^{-1}\vec{c} \rceil &= U\lfloor G^{-1}(B\vec{m}+\vec{e}) \rceil, \text{since } \vec{c} = B\vec{m}+\vec{e} \\ &= U\lfloor G^{-1}B\vec{m} + G^{-1}\vec{e} \rceil \\ &= \lfloor B^{-1}GG^{-1}B\vec{m} \rceil + U\lfloor G^{-1}\vec{e} \rceil, \text{since } U = B^{-1}G \\ &= \lfloor \vec{m} \rceil + U\lfloor G^{-1}\vec{e} \rceil \\ &= \vec{m} + U\lfloor G^{-1}\vec{e} \rceil, \text{since } \vec{m} \in \mathbb{Z}^n. \end{aligned}$$

Suppose that $\lfloor G^{-1}\vec{t} \rceil = \vec{0}$. Based on Proposition 4.1, $\lfloor G^{-1}\vec{e} \rceil = \vec{0}$ as well. Hence,

$$\begin{aligned} U\lfloor G^{-1}\vec{c} \rceil &= \vec{m} + U(\vec{0}) \\ &= \vec{m}. \end{aligned}$$

□

By satisfying the condition $\lfloor G^{-1}\vec{t} \rceil = \vec{0}$ in her key generation algorithm, Alice could ensure that $\lfloor G^{-1}\vec{e} \rceil = \vec{0}$. This indicates that, decryption error could be prevented and decryption yields the plaintext vector $\vec{m} \in \mathbb{Z}^n$. Therefore, the generated basis G is not only required to satisfy $orth_{def}(G) \approx 1$ condition, it is also required to satisfy the condition that $\lfloor G^{-1}\vec{t} \rceil = \vec{0}$ in order to be classified as a good basis and selected as a private basis in the GGH-MKA cryptosystem.

4.2 Conditions on the Public Basis

To be considered as a public basis, the $orth_{def}(B)$ is required to be as far as possible from 1 to ensure that the basis B is a bad basis. The purpose is to ensure that the basis B could not be used to perform the decryption algorithm. Without the private basis G, Eve could try to perform decryption using the public basis B since the bases G and B are spanning the same lattice $\mathcal{L}$, i.e., $L(G) = L(B) = \mathcal{L}$. Suppose that decryption is done using the public basis B as follows,

$$\vec{c} = B\vec{y}, \tag{4}$$

where $\vec{y} \in \mathbb{R}^n$ is an unknown vector. Compute the vector $\vec{y}$ as follows,

$$\vec{y} = B^{-1}\vec{c}. \tag{5}$$

The vector $\vec{y}$ is then rounded as $\lfloor \vec{y} \rceil \in \mathbb{Z}^n$ such that

$$|y_i - \lfloor y_i \rceil| < \frac{1}{2},$$

where $y_i \in \vec{y}$ and $\lfloor y_i \rceil \in \lfloor \vec{y} \rceil$ for all $i = 1, \ldots, n$. Consider the following lemma:

Lemma 4.2. *For $n \in \mathbb{N}$, let $\vec{c} = B\vec{m} + \vec{e}$ be a ciphertext vector where $B \in \mathbb{Z}^{n\times n}$ be a basis for the lattice $L(B) = \mathcal{L} \subset \mathbb{R}^n, \vec{m} \in \mathbb{Z}^n$ is a plaintext vector and $\vec{e} \in \mathbb{Z}^n$ is an error vector. Suppose that, $\vec{y} \in \mathbb{R}^n$ where $\vec{y} = B^{-1}\vec{c}$. Then, $\lfloor \vec{y} \rceil = \vec{m}$ if and only if $\lfloor B^{-1}\vec{e} \rceil = \vec{0}$.*

Proof. $\Rightarrow$ From $\vec{c} = B\vec{m} + \vec{e}$, it implies that $\vec{m} = B^{-1}(\vec{c} - \vec{e})$. Suppose that $\lfloor \vec{y} \rceil = \vec{m}$. Since $\vec{m} \in \mathbb{Z}^n$, then $\vec{m} = \lfloor \vec{m} \rceil$. Thus,

$$\begin{aligned} \vec{y} &= \lfloor \vec{m} \rceil \\ \vec{y} &= \lfloor B^{-1}(\vec{c} - \vec{e}) \rceil \\ \vec{y} &= \lfloor B^{-1}\vec{c} \rceil - \lfloor B^{-1}\vec{e} \rceil \\ \lfloor B^{-1}\vec{e} \rceil &= \lfloor B^{-1}\vec{c} \rceil - \vec{y}. \end{aligned}$$

Since $\vec{y} = B^{-1}\vec{c}$, then

$$\begin{aligned} \lfloor B^{-1}\vec{e} \rceil &= \lfloor B^{-1}\vec{c} \rceil - \lfloor B^{-1}\vec{c} \rceil \\ \lfloor B^{-1}\vec{e} \rceil &= \vec{0}. \end{aligned}$$

$\Leftarrow$ Given that $\vec{y} = B^{-1}\vec{c}, \vec{c} = B\vec{m} + \vec{e}$ and $\vec{m} \in \mathbb{Z}^n$. Thus,

$$\begin{aligned} \lfloor \vec{y} \rceil &= \lfloor B^{-1}\vec{c} \rceil \\ &= \lfloor B^{-1}\left(B\vec{m} + \vec{e}\right) \rceil \\ &= \lfloor B^{-1}B\vec{m} \rceil + \lfloor B^{-1}\vec{e} \rceil \\ &= \vec{m} + \lfloor B^{-1}\vec{e} \rceil. \end{aligned} \tag{6}$$

Suppose that $\lfloor B^{-1}\vec{e} \rceil = \vec{0}$. Hence, $\lfloor \vec{y} \rceil = \vec{m}$.

□

As proven in Lemma 4.2, the plaintext $\vec{m}$ can be recovered by executing the decryption algorithm using the public basis B if and only if $\lfloor B^{-1}\vec{e} \rceil = \vec{0}$. To prevent this from happening, the public basis B is necessary to satisfy the condition that $\lfloor B^{-1}\vec{e} \rceil \neq \vec{0}$. Without knowing the error vector $\vec{e}$, how could Alice check whether the generated B fulfils the condition $\lfloor B^{-1}\vec{e} \rceil \neq \vec{0}$ or not? To address this issue, consider the following proposition:

Proposition 4.2. *For $n, \sigma, k \in \mathbb{N}$ where $n = (4\sigma - 2)\,k$ and $\sigma > 2$, let $B \in \mathbb{Z}^{n\times n}, \vec{u} \in \{2-\sigma\}^n$ and $\vec{e} \in \mathbb{Z}^n$ where the entries $e_i \in \vec{e}$ are selected randomly from $E = \{(2-\sigma), (1-\sigma), \sigma, (\sigma+1)\}$ based on the distribution guideline in equation (3) for all $i = 1, \ldots, n$. If $\lfloor B^{-1}\vec{u} \rceil \neq \vec{0}$, then $\lfloor B^{-1}\vec{e} \rceil \neq \vec{0}$.*

Proof. Denote the entries of the inverse matrix $B^{-1} \in \mathbb{R}^{n\times n}$ as $b'_{i,j} \in B^{-1}$ for all $i, j = 1, \ldots, n$. Suppose that $\lfloor B^{-1}\vec{u} \rceil \neq \vec{0}$. Note that,

$$\begin{aligned} B^{-1}\vec{u} &= \begin{bmatrix} b'_{1,1} & b'_{1,2} & \cdots & b'_{1,n} \\ b'_{2,1} & bg'_{2,2} & \cdots & b'_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b'_{n,1} & b'_{n,2} & \cdots & b'_{n,n} \end{bmatrix} \begin{bmatrix} 2-\sigma \\ 2-\sigma \\ \vdots \\ 2-\sigma \end{bmatrix} \\ &= \begin{bmatrix} (2-\sigma)\left(b'_{1,1} + b'_{1,2} + \cdots + b'_{1,n}\right) \\ (2-\sigma)\left(b'_{2,1} + b'_{2,2} + \cdots + b'_{2,n}\right) \\ \vdots \\ (2-\sigma)\left(b'_{n,1} + b'_{n,2} + \cdots + b'_{n,n}\right) \end{bmatrix}. \end{aligned}$$

Since $\lfloor B^{-1}\vec{u} \rceil \neq \vec{0}$, this implies that

$$\left|u_i\left(b'_{i,1} + b'_{i,2} + \cdots + b'_{i,n}\right)\right| \geq \frac{1}{2},$$

for some $i = 1, \ldots, n$ where $u_i \in \vec{u}$ and $b'_{i,j} \in B^{-1}$ for all $i, j = 1, \ldots, n$. On the other hand,

$$B^{-1}\vec{e} = \begin{bmatrix} b'_{1,1} & b'_{1,2} & \cdots & b'_{1,n} \\ b'_{2,1} & b'_{2,2} & \cdots & b'_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b'_{n,1} & b'_{n,2} & \cdots & b'_{n,n} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{bmatrix} = \begin{bmatrix} e_1\left(b'_{1,1} + b'_{1,2} + \cdots + b'_{1,n}\right) \\ e_2\left(b'_{2,1} + b'_{2,2} + \cdots + b'_{2,n}\right) \\ \vdots \\ e_n\left(b'_{n,1} + b'_{n,2} + \cdots + b'_{n,n}\right) \end{bmatrix}.$$

For any $\sigma \in \mathbb{N}$ and $\sigma > 2$,

$$|2-\sigma| < |1-\sigma| < |\sigma| < |\sigma+1|.$$

This implies that,

$$\left|u_i\left(b'_{i,1} + b'_{i,2} + \cdots + b'_{i,n}\right)\right| \leq \left|e_i\left(b'_{i,1} + b'_{i,2} + \cdots + b'_{i,n}\right)\right|,$$

for any i-th row of the vectors $\lfloor B^{-1}\vec{u} \rceil$ and $\lfloor B^{-1}\vec{e} \rceil$ for all $i = 1, \ldots, n$. Since

$$\left| u_i \left(b'_{i,1} + b'_{i,2} + \cdots + b'_{i,n} \right) \right| \geq \frac{1}{2},$$

then

$$\left| e_i \left(b'_{i,1} + b'_{i,2} + \cdots + b'_{i,n} \right) \right| \geq \frac{1}{2},$$

as well for some $i = 1, \ldots, n$. This implies that, $\lfloor B^{-1}\vec{e} \rceil \neq \vec{0}$. □

Lemma 4.3. *For $n, \sigma, k \in \mathbb{N}$ where $n = (4\sigma - 2)\, k$ and $\sigma > 2$, let $G, B, U \in \mathbb{Z}^{n \times n}$, where G and B be bases for the lattice $\mathcal{L} \subset \mathbb{R}^n$ and U is a unimodular matrix such that $G = BU$. Then, let $\vec{c} = B\vec{m} + \vec{e}$ be a ciphertext vector where $\vec{m} \in \mathbb{Z}^n$ is a plaintext vector and $\vec{e} \in \mathbb{Z}^n$ is an error vector with entries $e_i \in \vec{e}$ are selected randomly from $E = \{(2-\sigma), (1-\sigma), \sigma, (\sigma+1)\}$ based on the distribution guideline in equation (3) for all $i = 1, \ldots, n$. If $\lfloor B^{-1}\vec{u} \rceil \neq \vec{0}$, then $\lfloor \vec{y} \rceil \neq \vec{m}$ where $\vec{u} \in \{2-\sigma\}^n$ and $\vec{y} = B^{-1}\vec{c}$.*

Proof. Given that $\vec{y} = B^{-1}\vec{c}$ where $G = BU$, $\vec{c} = B\vec{m} + \vec{e}$ and $\vec{m} \in \mathbb{Z}^n$. Thus,

$$\begin{aligned} \lfloor \vec{y} \rceil &= \lfloor B^{-1}\vec{c} \rceil \\ &= \lfloor B^{-1}\left(B\vec{m} + \vec{e}\right) \rceil \\ &= \lfloor B^{-1}B\vec{m} \rceil + \lfloor B^{-1}\vec{e} \rceil \\ &= \vec{m} + \lfloor B^{-1}\vec{e} \rceil. \end{aligned} \tag{7}$$

Suppose that $\lfloor B^{-1}\vec{u} \rceil \neq \vec{0}$. Based on Proposition 4.2, $\lfloor B^{-1}\vec{e} \rceil \neq \vec{0}$ as well. Hence,

$$\lfloor \vec{y} \rceil = \vec{m} + \lfloor B^{-1}\vec{e} \rceil \neq \vec{m}.$$

□

Other than measuring the value of $orth_{def}(B)$, Alice also needs to ensure that the basis B satisfies $\lfloor B^{-1}\vec{e} \rceil \neq \vec{0}$ in order to be classified as a bad basis and selected as the public basis. Although the error vector $\vec{e}$ is privately generated by Bob, Lemma 4.3 proves that it is sufficient for Alice to check whether $\lfloor B^{-1}\vec{u} \rceil \neq \vec{0}$ to ensure that the condition $\lfloor B^{-1}\vec{e} \rceil \neq \vec{0}$ is satisfied.

4.3 The Upgraded Key Generation Algorithm of the GGH-MKA Cryptosystem

By considering the proposed conditions on the private and public bases, a new key generation algorithm of the GGH-MKA cryptosystem is developed as the following:

Algorithm 4 New key generation algorithm of the GGH-MKA cryptosystem is done by Alice as the recipient.

Input: Parameter $\delta \in \mathbb{N}$ where $\sigma > 2$.
Output: Public key (B, σ, n), private key (G, U), set of entries E and its distribution guideline.

1: Decide the lattice dimension n as $n = (4\sigma - 2)k$ where $k \in \mathbb{N}$.
2: Generate a good basis $G \in \mathbb{Z}^{n\times n}$. If

i) $orth_{def}(G) \approx 1$, and

ii) $\lfloor G^{-1}\vec{t} \rceil = \vec{0}$ where $\vec{t} \in \{\sigma + 1\}^n$,

then G is accepted as a private basis. Otherwise, repeat Step 2.
3: Generate a unimodular matrix $U \in \mathbb{Z}^{n\times n}$.
4: Compute a bad basis $B \in \mathbb{Z}^{n\times n}$ as $B = GU^{-1}$. If

i) $orth_{def}(B)$ far from 1, and

ii) $\lfloor B^{-1}\vec{u} \rceil \neq \vec{0}$ where $\vec{u} \in \{2 - \sigma\}^n$,

then B is accepted as a public basis. Otherwise, repeat Step 3 and Step 4.
5: Setup the set $E = \{(2 - \sigma), (1 - \sigma), \sigma, (\sigma + 1)\}$ and its' distribution guideline as follows:

$$e_i = \begin{cases} (2-\sigma) & \text{for } \frac{n}{4\sigma-2} \text{ number of entries,} \\ (1-\sigma) & \text{for } \frac{\sigma n-n}{2\sigma-1} \text{ number of entries,} \\ \sigma & \text{for } \frac{n}{4\sigma-2} \text{ number of entries,} \\ (\sigma+1) & \text{for } \frac{\sigma n-n}{2\sigma-1} \text{ number of entries.} \end{cases} \tag{8}$$

6: Keep the private key (G, U) secretly and send the public key (B, σ, n) together with the set E and distribution guideline in equation (8) to the sender.

In two respects, the new key creation Algorithm 4 is distinct from the original key generation Algorithm 1, Step 2 and Step 4. The original GGH cryptosystem and its current variant, the GGH-MKA cryptosystem, use the key generation Algorithm 1. In both cryptosystems, the orthogonality of the produced basis G and the computed basis B is exclusively determined by their orthogonality defect values, as mentioned in Steps 2 and 4 of Algorithm 1. If $orth_{def}(G) \approx 1$, the generated G is accepted as a private basis, whereas the computed B is accepted as a public basis if $orth_{def}(B)$ is far than 1. In Algorithm 4, new requirements are proposed in Steps 2(ii) and Step 4(ii) to make the process of determining whether the generated G and computed B are accepted or not as private and public bases, respectively, more transparent, and deterministic.

Nevertheless, how near is it close to 1, and how distant is it far from 1? These measurements are far too subjective and ad hoc. Hence, there is no measurable range that can be used to address this subject. While the conditions in Step 2(i) and Step 4(i) are too subjective to determine, the conditions in 2(ii) and 4(ii) may be determined by computing the vectors $\lfloor G^{-1}\vec{t} \rceil$ and $\lfloor B^{-1}\vec{u} \rceil$ and comparing those vectors to the vector $\vec{0}$. Instead of relying entirely on subjective conditions in Step 2(i) and Step 4(i), the GGH-MKA cryptosystem's upgraded key generation algorithm employs deterministic conditions in Step 2(ii) and Step 4(ii).

5 Conclusion

This paper presents some additional conditions to be considered in the key generation algorithm of the GGH-MKA cryptosystem. Using the proposed conditions, the generated bases G and B by Alice as the recipient become more apparent and more specific. Alice could ensure that the generated G would make the decryption free from error once the proposed condition as stated in Lemma 4.1 for the basis G is satisfied. At the same time, Alice also could ensure that the basis B that is derived from the basis G would make decryption attempt by Eve using the basis B would fail once the proposed conditions as stated in Lemma 4.3 for the basis B is satisfied. Instead of relying solely on subjective measurement and general terms, the upgraded classification process of the lattice bases G and B in the key generation algorithm of the GGH-MKA cryptosystem becomes more precise and deterministic compared to the key generation algorithm of the GGH cryptosystem.

The GGH cryptosystem and the fatal attacks on it are developed and executed using early 2000s computing technology. Thus, all the security and efficiency issues are discussed and concluded based on experimental results conducted using those technologies. We are currently expanding the breadth of our experimental results to resolve the security analysis and decryption failure rate. Additionally, we analyse the efficiency issue in terms of computing time comparison. It is worth highlighting the issue of decryption failure probabilities and their improvement when circumstances are incorporated. These experimental comparisons address the topic of security measurement using the public key in conjunction with the likelihood of decryption error using the private key. We regard it as study material deserving of our future work.

Acknowledgement The authors would like to express their gratitude to the anonymous reviewers for their valuable comments and suggestions for a betterment of this paper. This research was sponsored by Fundamental Research Grant Scheme (203.PMATHS.6711941) from Ministry of Higher Education Malaysia.

Conflicts of Interest The authors declare no conflict of interest.

References

[1] M. Ajtai (1999). Generating hard instances of the short basis problem. In *Automata, Languages and Programming*, pp. 1–9. Springer, Berlin, Heidelberg.

[2] C. F. de Barros & L. M. Schechter (2015). GGH may not be dead after all. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, *3*(1). https://doi.org/10.5540/03.2015.003.01.0095.

[3] T. Elgamal (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transaction on Information Theory*, *4*(31), 469–472.

[4] S. D. Galbraith (2012). *Mathematics of Public Key Cryptography*. Cambridge University Press, Cambridge.

[5] O. Goldreich, S. Goldwasser & S. Halevi (1997). Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology - CRYPTO'97*, pp. 112–131. Springer, Berlin, Heidelberg.

[6] J. Hoffstein, J. Pipher & J. H. Silverman (1998). NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory*, pp. 267–288. Springer, Berlin, Heidelberg.

[7] J. Hoffstein, J. Pipher & J. H. Silverman (2008). *An Introduction to Mathematical Cryptography*. Springer-Verlag New York, New York, NY.

[8] N. Koblitz (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, *177*(48), 203–209.

[9] M. S. Lee & S. G. Hahn (2010). Cryptanalysis of the GGH cryptosystem. *Mathematics in Computer Science*, *3*(2), 201–208.

[10] A. Mandangan, H. Kamarulhaili & M. Asbullah (2019). On the smallest-basis problem underlying the GGH lattice-based cryptosystem. *Malaysian Journal of Mathematical Sciences*, *13*, 1–11.

[11] A. Mandangan, H. Kamarulhaili & M. Asbullah (2020). A security upgrade on the GGH lattice-based cryptosystem. *Sains Malaysiana*, *49*(6), 1471–1478.

[12] D. Micciancio & S. Goldwasser (2002). Closest vector problem. In *The Springer International Series in Engineering and Computer Science*, pp. 45–68. Springer, Boston, MA.

[13] D. Micciancio (2001). Improving lattice based cryptosystems using the Hermite normal form. In *Cryptography and Lattices*, pp. 126–145. Springer, Berlin, Heidelberg.

[14] P. Nguyen (1999). Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from CRYPTO 97. In *Advances in Cryptology - CRYPTO'99*, pp. 288–304. Springer, Berlin, Heidelberg.

[15] S.-H. Paeng, B. E. Jung & K.-C. Ha (2003). A lattice based public key cryptosystem using polynomial representations. In *Public Key Cryptography - PKC 2003*, pp. 292–308. Springer, Berlin, Heidelberg.

[16] R. L. Rivest, A. Shamir & L. Adleman (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *2*(21), 120–126.

[17] D. Serre (2010). *Matrices - Theory and Applications*. Springer, New York, NY.

[18] P. W. Shor (1994). Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science*, pp. 124–134. IEEE, Santa Fe, NM.

[19] M. Yoshino & N. Kunihiro (2012). Improving GGH cryptosystem for large error vector. In *2012 International Symposium on Information Theory and its Applications*, pp. 416–420. IEEE, Honolulu, HI.