Efficient variants of the GGH-YK-M cryptosystem

João M. M. Barguil[1] **, Renan Yuri Lino**[1] **, Paulo S. L. M. Barreto**[1*]

[1] Escola Politécnica, University of São Paulo.

{jbarguil,rlino,pbarreto}@larc.usp.br

Abstract. *The Goldreich-Goldwasser-Halevi (GGH) public-key encryption scheme was deemed broken until recently proposed variants were shown to thwart all known attacks. However, the associated key sizes and generation times are notoriously inefficient. In this paper, we improve on the most promising such variant, proposed by Barros and Schechter and called GGH-YK-M, by reducing public key sizes rom $O(n^2 \lg n)$ down to $O(n \lg n)$ bits, and making key generation over 3 orders of magnitude faster than the results in the literature.*

Keywords: *lattice-based encryption.*

1. Introduction

There is a rising, medium to long term concern with the potential technological viability of quantum computers, because traditional cryptosystems based on the assumed hardness of integer factorization or discrete logarithm computation can be attacked with the help of this new kind of equipment [22]. New schemes based on different computational problems are thus necessary to address this concern, leading to the development of purely classical, but quantum-resistant constructions dubbed *post-quantum* cryptosystems [2].

The most popular family of post-quantum cryptosystems is that of schemes whose security relates to certain hard problems on lattices. Two examples of such problems are the *Shortest Vector Problem* (SVP) and *Closest Vector Problem* (CVP). The former consists of finding a certain approximation (to a factor $\gamma(n)$ where n is the lattice dimension) to the shortest vector in a given lattice, and the latter is to find a lattice vector that is closest to a given vector not necessarily in the lattice. Both of these problems are deemed hard to solve for the Euclidean norm and suitably chosen $\gamma(n)$, as there is no known method to solve them in polynomial time.

One of the pioneering lattice-based encryption schemes, proposed by Goldreich, Goldwasser and Halevi [10] and appropriately dubbed GGH, can be seen as a generalization of the McEliece scheme [16]. In this scheme, a message is translated to a vector in a given lattice and a small error is added. The message is recovered by solving the CVP in that lattice. Known algorithms for solving the CVP work well with short lattice bases, but not with long bases. GGH is a public-key encryption scheme that uses a good lattice basis as the private key and the corresponding Hermite normal form (HNF) [5, section 2.4.2] as the public key. Nguyen proved that the original GGH had inherent structural flaws [19]

*This research was supported by Intel Research grant "Energy-efficient Security for SoC Devices – Asymmetric Cryptography for Embedded Systems" 2012, and by the São Paulo Research Foundation (FAPESP) grant 2013/25977-7. P. Barreto is also supported by the Brazilian National Council for Scientific and Technological Development (CNPq) research productivity grant 306935/2012-0.

and was able to break typical, realistic GGH instances by using lattice reduction algorithms like LLL [12] and BKZ [21].

For several years the GGH scheme was deemed irretrievably broken, to the extent that other kinds of lattices stemming from the Learning with Errors (LWE) problem [20] have essentially dominated the research in the area. This situation began to change when Yoshino and Kunihiro [25] described a variant of GGH (aptly called GGH-YK) that thwarts all known attacks. However, their scheme was incomplete in the sense that, by blindly following their prescriptions, no proper parameter set can be feasibly constructed.

Recently, Barros and Schechter [6] revisited the GGH-YK construction, and proposed a surprising modification of that scheme (dubbed GGH-YK-M, from the fact that it makes essential use of M-matrices [1]) that effectively yields a suitable parametrization. The result is very promising, as it brings the simplicity of GGH and GGH-YK back to life.

The remaining aspect to address, therefore, is to circumvent the inherent high bandwidth occupation and computational cost incurred by all traditional variants of GGH, which make this family of schemes less competitive in practice with other lattice-based encryption methods like Lindner-Peikert [13]. The obvious way to obtain shorter keys in other lattice-based settings like LWE or NTRU [11], namely, resorting to certain rings of structured (e.g. circulant or negacyclic) matrices, fails for GGH because mapping the private key to a public key, that is, computing the HNF, ends up destroying the underlying structure that would enable the size reduction, and thus does not help in attaining that goal.

The technique proposed by Smart and Vercauteren [23] and perfected by Gentry and Halevi [9], targeted at homomorphic encryption, can be used to address this problem. However, the former depends on the lattice determinant to be prime, while the latter relies heavily on the special form of the ring $\mathbb{Z}[x]/(x^n + 1)$ where n is a power of 2. Besides, it requires the computation of resultants and the explicit extraction of the roots of polynomials modulo the lattice determinant, which is done through a quite complex modification of the extended Euclidean algorithm.

Contributions: In this paper we describe an efficient key generation technique that reduces public key bandwidth occupation by an order of complexity, specifically, from $O(n^2 \lg n)$ down to $O(n \lg n)$, while avoiding the need to resort to a full-fledged HNF algorithm, in the same way as the Smart-Vercauteren and Gentry-Halevi methods[1]. Our work extends their technique to any value of n and also for the circulant ring $\mathbb{Z}[x]/(x^n - 1)$, for which we also provide a structural security analysis. In particular, and surprisingly, prime values of n are observed to lead to faster key generation, despite the unavailability of fast Fourier transform techniques to speed up the computations. Our technique only requires a straightforward application of the usual extended Euclidean algorithm, coupled with the

[1]Note added in revision: we were first made aware of the Smart-Vercauteren and Gentry-Halevi key generation techniques after this paper was written. We missed them apparently because of our different target (conventional rather than homomorphic encryption). However, as we explicitly indicate, our proposal is more general, arguably simpler, and empirically more efficient than those methods.

Chinese remainder theorem and the fast Fourier transform.

Our proposal attains much faster processing in all operations involved in a GGH-style cryptosystem, that is, key generation, encryption, and decryption. By far the most pronounced improvement is in key generation, which becomes more than 3 orders of magnitude faster than published results, while encryption becomes almost 2 orders of magnitude faster (our implementation is twice as fast as the literature for decryption). Although our goal was to optimize the GGH-YK-M scheme, it may turn out that our proposal is useful for other scenarios as well, like the somewhat homomorphic encryption scheme of Loftus *et al.* [14] which is the only such scheme so far that resists key recovery attacks [4].

The remainder of this paper is organized as follows. Section 2 introduces basic concepts and notation. We describe the GGH-YK-M scheme in Section 3. Our proposed improvements are put forward in Section 4. In Section 5 we make some security considerations on the improved scheme. The results of experimental assessment and comparisons with the previous state of the art are detailed in Section 6. We conclude in Section 7.

2. Preliminaries

Vector and matrix indices are numbered starting from 0 throughout this paper. We denote by $M_{(i)}$ the i-th row of a matrix M, and by M_j the j-th element on its first row, i.e. $M_j := M_{(0],j}$. We also denote by $x \xleftarrow{\$} U$ the uniformly random sampling of variable x from set U.

Definition 1. *Let $P \in \mathbb{C}^{n\times n}$. The* spectral radius *of P is the quantity $\rho(P) := \max\{|\lambda| : \lambda \text{ is an eigenvalue of } P\}$.*

Definition 2. ([1, Definition 1.2]) *Let $P \in \mathbb{Z}^{n\times n}$ such that $P_{ij} \leqslant 0$ for all $0 \leqslant i, j < n$. A (nonsingular) M-*matrix *is a matrix of form $A = \gamma I + P$ for some $\gamma > \rho(P)$.*

Definition 3. ([5, section 2.4.2]) *A matrix $H \in \mathbb{Z}^{n\times n}$ is said to be in* Hermite normal form *(HNF) if it is upper triangular, all its elements are non-negative and the entries on the diagonal are positive and are the largest entries in their respective columns.*

Definition 4. *A matrix $H \in \mathbb{Z}^{n\times n}$ in HNF is said to be* minimal *if it has the form*

$$H = \left[\begin{array}{c|c} I_{n-1} & v^T \\ \hline 0^{n-1} & d \end{array}\right],$$

where $v \in \mathbb{Z}^{n-1}$ and $d \in \mathbb{Z}$.

One can check by direct inspection that the inverse (over $\mathbb{Q}$) of a matrix H in minimal HNF is

$$H^{-1} = \left[\begin{array}{c|c} I_{n-1} & -(1/d)v^T \\ \hline 0^{n-1} & 1/d \end{array}\right].$$

Thus a matrix H in minimal HNF can be conveniently represented by $(v, d) \in \mathbb{Z}^n$ alone. Also, it is clear that $\det(H) = d$.

3. The GGH-YK-M scheme

We now summarize the intriguing GGH variant proposed by Barros and Schechter [6], which itself improves on the GGH-YK scheme by Yoshino and Kunihiro [25], and was called GGH-YK-M by virtue of resorting to M-matrices [1] to complete the specification of that scheme.

For simplicity and efficiency, in our description of GGH-YK-M we explicitly require that the private lattice basis A be such that its HNF is minimal.

Let n be an integer (usually, but not necessarily, a power of 2), let γ be a multiple of n by some small factor (i.e. $\gamma = \alpha n$ for some small integer α), let σ be an even integer, and let h and k be integers such that $h + k < \gamma < 2h$. The GGH-YK-M encryption scheme [6] was designed to thwart all known attacks applicable against the GGH scheme [10], and consists of the following three algorithms:

- Keygen: Sample $P \xleftarrow{\$} \{-1, 0\}^{n \times n}$, compute $A \leftarrow \gamma I + P$ and its HNF $H := \mathrm{HNF}(A)$ until $\rho(P) < \gamma$, $1/\gamma < \left|(A^{-1})_{ii}\right| \leqslant 2/\gamma$ for $0 \leqslant i < n$, $\left|(A^{-1})_{ij}\right| < 2/\gamma^2$ for $i \neq j$, and H is in minimal form. Empirically, taking α in the definition $\gamma = \alpha n$ to be as small as 2 is usually enough to ensure that these conditions hold with high probability. The private key is A, and the public key is $(v, d) \in \mathbb{Z}^n$. Since $v_i < d$ from the definition of the HNF (see Definition 3), and $d = O(\gamma^n)$ by virtue of the Hadamard bound on the size of the determinant of a matrix [8], it follows that the public key has size $O(n^2 \lg \gamma)$ or simply $O(n^2 \lg n)$ bits, while the private key, which is essentially P, has size n^2 bits.

- Encrypt: Let $m \in \{0, 1\}^{n-k}$ be the plaintext. Select a random subset $S \subset \{1 \ldots n\}$ with k elements. The encoding of m is a vector $r \in \mathbb{Z}^n$ such that $r_i = h$ for $i \in S$, otherwise $r_i \xleftarrow{\$} \{1 \ldots \sigma/2\}$ if $m_j = 0$, and $r_i \xleftarrow{\$} \{\sigma/2 + 1 \ldots \sigma\}$ if $m_j = 1$, where i corresponds to the j-th index not in S. Compute $r - \lfloor rH^{-1} \rfloor H$, which, because of the particularly simple structure of the minimal HNF (see Definition 4), has the form $(0, \ldots, 0, c)$. The cryptogram is $c \in \mathbb{Z}$, the only nonzero coefficient thereof.

- Decrypt: Let $c \in \mathbb{Z}$ be the ciphertext. Compute $c' \leftarrow (0, \ldots, 0, c)A^{-1} \in \mathbb{Q}^n$, which means simply $c' \leftarrow cA^{-1}_{(n-1)}$, and let $r' \leftarrow (c' - \lfloor c' \rceil)A$. Compute the error vector $e \in \{0, 1\}^n$ by letting $e_i \leftarrow 1$ whenever $r'_i < 0$, otherwise $e_i \leftarrow 0$, for all $0 \leqslant i < n$. Compute the recovered message encoding as $r \leftarrow r' + eA$. Let $S := \{i \mid r_i = h\}$ (this is the same set S chosen during encryption). For all $0 \leqslant i < n$ such that $i \notin S$, extract $m_j \leftarrow 0$ if $0 < r_i \leqslant \sigma/2$, and $m_j \leftarrow 1$ if $\sigma/2 < r_i \leqslant \sigma$, where i corresponds to the j-th index not in S.

Notice that, strictly speaking, this is only a trapdoor one way function, not a full semantically secure encryption scheme. To attain semantic security, a suitable transform like Fujisaki-Okamoto [7] should be used.

4. Improvements

The usual technique adopted to reduce space requirements and bandwidth occupation in lattice-based cryptosystems is to resort to certain structured matrices that correspond to ideals in polynomial rings [15, 17, 18].

The most popular choices are circulant matrices, associated to the polynomial ring $\mathbb{Z}[x]/(x^n - 1)$, and negacyclic matrices, which correspond to the polynomial ring $\mathbb{Z}[x]/(x^n + 1)$. Due to security concerns with the idea of working on a ring (where not all nonzero elements have inverses), Bernstein [3] suggests adopting a number field instead, specifically a field of form $\mathbb{Z}[x]/(x^n - x - 1)$ because of the very simple form of the irreducible polynomial $x^n - x - 1$, which yields nearly circulant matrices and fairly efficient arithmetic. More generally, one could consider the $n \times n$ matrices whose i-th row contains the coefficients of $a(x)x^i \bmod p(x)$ for some $a(x)$ and a fixed but arbitrary monic polynomial $p(x)$ of degree n without multiple roots (and preferably small coefficients). Such matrices correspond to the ideals of a polynomial ring $\mathbb{Z}[x]/p(x)$.

Unfortunately, this technique does not seem to improve the space requirements of GGH, nor, for that matter, those of GGH-YK-M. This is because the HNF is usually *not* in the same (structured) ring as the original matrix. Thus, for instance, HNF(A) in general is *not* circulant or negacyclic even though A displays such symmetries (except if A is a scalar matrix). Therefore, by resorting to circulant or similarly structured matrices one would apparently be able at most to reduce the size of private keys from n^2 down to n bits, but not that of public keys, which stay at $O(n^2 \lg \gamma)$ bits.

Contrary to this intuitive observation, one can still benefit from an underlying structure in the private key to reduce the size of the public key in a nontrivial way. This was first indicated by Smart and Vercauteren [23], but it seems to require computing the HNF of the lattice basis. Gentry and Halevi [9, Lemma 1] offer a proof of this property that avoids computing the HNF for the case $p(x) = x^n + 1$ (where n is a power of 2). We show that, in fact, it holds for any ideal matrix, regardless of the choice of $p(x)$, even though some choices may be more efficient (and possibly more secure) than others.

If matrix P in the Keygen algorithm is associated to a polynomial ring $\mathbb{Z}[x]/p(x)$, then matrix A is associated to a polynomial in the same ring, and although $H := \text{HNF}(A)$ does not display the ring symmetry (i.e. H is not circulant, etc), its rows still correspond to elements of that ring. Thus, if $a(x)$ is the polynomial associated to any row of H, then $xa(x) \bmod p(x)$ and $x^{-1}a(x) \bmod p(x)$ are two other (independent) vectors on the same lattice.

Given that $H_{(n-2)} = (0, \ldots, 0, 1, u)$ for some $u \in \mathbb{Z}$ (because H is assumed to be minimal), the polynomial associated to it is $ux^{n-1} + x^{n-2} = (ux + 1)x^{n-2}$, and hence $(ux + 1)x^i = (x^{-1})^{i-(n-2)}(ux + 1)x^{n-2}$ stands for yet another vector on that lattice for every $0 \leqslant i < n - 1$. Collecting all of these vectors together with $H_{(n-1)}$, one gets

$$H' = \begin{bmatrix} 1 & u & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & u & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & u \\ 0 & 0 & 0 & \ldots & 0 & 0 & d \end{bmatrix}, \tag{1}$$

which is an alternative basis for the same lattice, since all of its rows are linearly independent vectors from that lattice, and H' shares the same determinant d as H (and A). But because the HNF is unique, it also follows that $\text{HNF}(H') = H$, and by applying a straightforward Gaussian elimination on H', namely by changing $H'_{(n-1-j)} \leftarrow H'_{(n-1-j)} - uH'_{(n-j)}$

successively for $2 \leqslant j < n$ and then reducing modulo d, one gets

$$H = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & -(-u)^{n-1} \bmod d \\ 0 & 1 & \dots & 0 & 0 & -(-u)^{n-2} \bmod d \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & -u^2 \bmod d \\ 0 & 0 & \dots & 0 & 1 & u \\ 0 & 0 & \dots & 0 & 0 & d \end{bmatrix}, \tag{2}$$

and by comparing the result with the definition of minimal H (see Definition 4) yields $v_i = -(-u)^{n-1-i} \bmod d$ for $0 \leqslant i < n-1$.

Therefore, H (and its inverse) can be efficiently represented simply by $(u, d) \in \mathbb{Z}^2$. Because $0 < u < d$ and d satisfies the Hadamard bound for A, which is $d < \gamma^n$, it follows that H can be represented with only $2n \lg \gamma$ and hence $O(n \lg n)$ bits, a vast improvement over the naive $O(n^2 \lg \gamma)$ or $O(n^2 \lg n)$ size of the whole (v, d) for practical values of n (typically in the hundreds).

The remaining tasks are computing d and u from A. We now address these tasks individually. Our approach avoids both the computation of resultants and complex modifications of the extended Euclidean algorithm.

Computing the determinant d is accomplished by diagonalizing the projections of A onto a number of finite fields $\mathbb{F}_{q_0}, \dots, \mathbb{F}_{q_t-1}$ such that $d < \prod_k q_k$, since this enables computing $d \bmod q_k$ for each q_k, and then recovering d by means of the Chinese remainder theorem. This is possible as long as the polynomial $p(x)$ splits completely into n distinct linear factors over each of those fields. If that is the case, let $V \in \mathbb{F}_{q_k}^{n \times n}$ be the Vandermonde matrix built from the n distinct roots of $p(x)$ over $\mathbb{F}_{q_k}$, i.e. $V_{ij} := z_j^i$ with $p(z_j) = 0$ and $z_j \in \mathbb{F}_{q_k}$. Then V is invertible, and the diagonal form of A is $V^{-1}AV$ (the eigenvalues themselves are just the sequence of components of $A_{(0)}V$).

The obstacle to this approach is finding the fields $\mathbb{F}_{q_k}$ such that $p(x)$ splits in the required form over all of them. Exhaustive search via the factorization of an arbitrary $p(x)$ over candidate fields is far too expensive, even for fairly small n. One could reverse the reasoning and choose the roots of $p(x)$ first, but this only enables the computation of a single field $\mathbb{F}_q$ over which $p(x)$ splits, and because the coefficients of such a $p(x)$ are expected to be rather large, any private basis is usually very large as well, yielding an even larger determinant d which is likely to exceed q by a factor exponentially large in n, and hence precluding the recovery of d from its value mod q alone.

However, the circulant and negacyclic cases offer a much better prospect, since all that is required for $p(x)$ to split over $\mathbb{F}_{q_k}$ is that $n \mid q-1$ in the former case, and $2n \mid q-1$ in the latter. When n is a power of 2, the computation of the diagonal form of A amounts to a fast Fourier transform (more precisely, a fast number theoretic transform), which takes time $O(n \lg n)$ products by certain fixed roots of unity in $\mathbb{F}_{q_k}$. However, computation of the eigenvalues is fairly efficient even for general n, and as we shall see this extra flexibility in the choice of n tends, a bit surprisingly, to offer better key generation performance.

Assuming that the fields $\mathbb{F}_{q_k}$ are available and that the determinant d has been computed, the value of u, if it exists, can be computed as follows. The first row of H is expected to have the form $(1, u, 0, \dots, 0)$, associated to the polynomial $ux + 1$ in the

underlying polynomial ring. The rows of the matrix H^* corresponding to this polynomial spell the coefficients of $(ux + 1)x^j \bmod p(x)$. Thus H^* differs from H only in its last row, and it defines a sub-lattice of the lattice defined by H or, equivalently, by A.

Therefore, there must exist a matrix $M \in \mathbb{Z}^{n\times n}$ (actually in the same ring as A and H^*) such that $MA = H^*$. Let $A^\dagger$ be the classical adjoint (or adjugate) of A, i.e. $AA^\dagger = dI$. Then $dM = H^*A^\dagger$, and the peculiar structure of H^* reduces this to the Diophantine equation $dM_j - A^\dagger_{j-1}u = A^\dagger_j$ for all j. Thus, if any solution to this equation exists, it is $u = -A^\dagger_j/A^\dagger_{j-1 \pmod n} \bmod d$ for any j, which requires all $A^\dagger_j$ to be invertible mod d. However, this in turn actually requires only that $A^\dagger_0$ and $A^\dagger_1$ be invertible mod d, since then $A^\dagger_j = A^\dagger_0(A^\dagger_1/A^\dagger_0)^j = A^\dagger_0(-u)^j \bmod d$ as one can check by induction.

This provides a simple algorithm to determine at once whether HNF(A) is minimal, and if so, what the value of u in Equation 1 is. Indeed, $A^\dagger$ can be computed via the Chinese remainder theorem from $A^\dagger = dA^{-1} \bmod q_k$, and the extended Euclidean algorithm then yields $u \leftarrow -A^\dagger_1/A^\dagger_0 \bmod d$ or proves that no such u exists.

The efficient key pair generation this process enables, without a full HNF algorithm, arguably outweighs the practical restriction for $p(x) = x^n \pm 1$. This method works for any choice of n. Processing times are much smaller for this compact representation than they are for unstructured matrices. We report on experimental results in Section 6.

5. Security considerations

Adopting a structured matrix as the private key must be made carefully to avoid introducing weaknesses. The particular case $p(x) = x^n + 1$ where n is a power of 2 has received a considerable amount of attention in the literature. We now analyze how circulant lattices, corresponding to $p(x) = x^n - 1$, have the drawback of leaking a small amount of information on the private key, specifically $O(\lg n)$ bits thereof. As always, our analysis does not require n to be a power of 2. Admittedly, the security level attainable when generalizing n is less clear, though it seems unlikely that this would introduce any weakness that is not already present in the more extensively analyzed NTRU scenario, where prime n is the usual choice.

We begin by noticing that the sum $\lambda := \sum_j A_j$ is bound between $\gamma - n$ (when P is the all-one ring element) and γ (when P is zero). Let $\Lambda := \sum_j A^\dagger_j$. The following property holds:

Lemma 1. $d = \lambda\Lambda$.

Proof. By definition of adjugate matrix, $AA^\dagger = dI$. Then $A_{(j)}A^\dagger = dI_{(j)}$ and hence $\sum_j A_{(j)}A^\dagger = \sum_j dI_{(j)}$, which yields $(\lambda, \ldots, \lambda)A^\dagger = (d, \ldots, d)$, since the elements on each column of A are the same except for a circular permutation, and thus all columns of $\sum_j A_{(j)}$ take the value $\sum_j A_j = \lambda$. Now $(\lambda, \ldots, \lambda)A^\dagger = \lambda(1, \ldots, 1)A^\dagger$, which is simply $(\lambda\Lambda, \ldots, \lambda\Lambda)$ because $(1, \ldots, 1)A^\dagger = (\sum_j A^\dagger_j, \ldots, \sum_j A^\dagger_j) = (\Lambda, \ldots, \Lambda)$. Therefore $(\lambda\Lambda, \ldots, \lambda\Lambda) = (d, \ldots, d)$ which repeats the claim n times, i.e. $d = \lambda\Lambda$. □

Lemma 2. $(-u)^n - 1 \equiv 0 \bmod d$.

Proof. We show that $(0, \ldots, -(-u)^n + 1)$ is a lattice vector in the subspace generated by $H_{(n-1)} = (0, \ldots, d)$. Consider the lattice generated by

$$C^{(0)} = \begin{bmatrix} 1 & u & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & u & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & u & 0 & 0 \\ 0 & 0 & 0 & \ldots & 1 & u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & u \\ u & 0 & 0 & \ldots & 0 & 0 & 1 \end{bmatrix},$$

where the superscript denotes a stage in the Gaussian elimination process described below, with (0) indicating the original matrix. This is a sublattice of the original lattice, since it only involves rotations of the first row of H' defined by Equation 1. Applying Gaussian elimination to the last row as $C^{(j+1)}_{(n-1)} \leftarrow C^{(j)}_{(n-1)} + (-u)^{j+1} C_{(j)}$ for $j = 0, \ldots, n-1$, we get

$$C^{(n)} = \begin{bmatrix} 1 & u & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & u & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & u & 0 & 0 \\ 0 & 0 & 0 & \ldots & 1 & u & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 & u \\ 0 & 0 & 0 & \ldots & 0 & 0 & -(-u)^n + 1 \end{bmatrix}.$$

Thus $C^{(n)}_{(n-1)}$ is in the subspace spanned by $H'_{(n-1)}$, i.e. $C^{(n)}_{(n-1)} = \kappa H'_{(n-1)}$ for some κ. Thus $-(-u)^n + 1 = \kappa d$, i.e. $(-u)^n - 1 \equiv 0 \bmod d$ as claimed. □

Let $Z := \sum_j (-u)^j \bmod d$. Given that $A^\dagger_j = A^\dagger_0 (-u)^j \bmod d$, it follows that $\sum_j A^\dagger_j = A^\dagger_0 \sum_j (-u)^j \bmod d$ and thus $\Lambda = A^\dagger_0 Z \bmod d$. At first glance this equation might seem to provide a means to recover the full $A^\dagger_0$ by inverting $Z \bmod d$. That this cannot actually happen is established by the following property:

Lemma 3. $\Lambda \mid \gcd(Z, d)$*, and hence* Z *is not invertible* mod d.

Proof. From $A^\dagger_0 Z = \Lambda \bmod d$ and Lemma 2 it follows that $\lambda A^\dagger_0 Z = \lambda\Lambda = 0 \bmod d$ and since, by the key generation requirement of Section 4, $A^\dagger_0$ itself is invertible mod d, then $\lambda Z = 0 \bmod d$, i.e. $\lambda Z = Z' d = Z' \lambda \Lambda$ for some integer Z', meaning that $Z = Z'\Lambda$, i.e. Z itself is a multiple of Λ, and hence cannot be invertible mod d by virtue of having the common factor Λ with d. □

However, equation $\Lambda = A^\dagger_0 Z \bmod d$ does reveal a small piece of information on $A^\dagger_0$. Indeed, $\Lambda = A^\dagger_0 Z + \kappa d = A^\dagger_0 Z' \Lambda + \kappa\lambda\Lambda$ for some κ, and hence $1 = A^\dagger_0 Z' + \kappa\lambda$ by removing the common factor Λ, or simply $1 = A^\dagger_0 Z' \bmod \lambda$. This reveals $A^\dagger_0 \bmod \lambda = Z'^{-1} \bmod \lambda$

as long as Z' is invertible mod λ. However, this amounts to revealing only $O(\lg n)$ bits of the private value $A_0^\dagger$.

On the constructive side, $(u+1)\Lambda = A_0^\dagger(u+1)\sum_j(-u)^j \bmod d = -A_0^\dagger((-u)^n - 1) \bmod d = 0$, $(u+1)\Lambda = \xi\lambda\Lambda$ for some ξ, and hence $\lambda \mid u+1$. Thus λ is a common factor between d and $u+1$, and can be factored out by publishing the public key as the triple $(d/\lambda, (u+1)/\lambda, \lambda)$ instead of the pair (u, d), saving $O(\lg n)$ bits.

This also shows that the attack cannot be extended to recover the whole matrix $A \bmod \lambda$ (from which A could be extracted immediately) from $A^\dagger \bmod \lambda$. Because $u+1 = 0 \bmod \lambda$ and hence $-u = 1 \bmod \lambda$, it follows that $A_j^\dagger \bmod = A_0^\dagger(-u)^j \bmod \lambda$ (this equality holds because $\lambda \mid d$) and hence $A_j^\dagger = A_0^\dagger \bmod \lambda$ for all j, so that $A^\dagger = A_0^\dagger U \bmod \lambda$ where U is the (singular) all-one matrix. Therefore the adjugate mapping mod λ cannot be inverted to recover A from $A^\dagger \bmod \lambda$.

Interestingly, this attack does not apply to negacyclic lattices (or, for that matter, most or perhaps all other ideal lattices), because Lemma 1 does not hold, i.e. the determinant, in general, is not the product of a linear combination of the components of A and a linear combination of the components of $A^\dagger$.

6. Experimental results

We implemented the improved encryption scheme in Java running on an Intel i5-3210M 2.5 GHz platform under 64-bit Windows 7.

To facilitate comparison with the literature [6], where timings, obtained from an implementation in C/C++, are only available on an AMD 1.6 GHz platform, our speeds are shown scaled down by a factor 1.6/2.5 on Table 1. Performance turned out to be already highly competitive with the prior state of the art, in spite of the adoption of Java rather than C/C++.

We disregard lattice dimensions smaller than 350, since they are susceptible to attacks [6], and we set n to be either a prime or a power of 2. We provide data for dimensions around 512 as well, going somewhat beyond the dimensions found in that reference. The times needed to gather suitable primes for the Chinese remainder theorem are not included since they are precomputed only once and stored. For simplicity, we only consider the circulant ring $\mathbb{Z}[x]/(x^n - 1)$, since the times corresponding to negacyclic ring $\mathbb{Z}[x]/(x^n + 1)$ would be very similar to those corresponding to the circulant case.

Table 1. Timings (in seconds)

source	(n, σ, h, k)	keygen (s)	encrypt (ms)	decrypt (ms)
[6]	(350, 256, 526, 64)	1662.55	60.0	170.0
ours	(353, 256, 526, 64)	0.48	0.7	88.5
[6]	(400, 256, 601, 64)	3127.17	70.0	270.0
ours	(401, 256, 601, 64)	0.65	0.8	132.3
ours	(509, 256, 769, 80)	1.30	1.2	448.0
ours	(512, 256, 769, 80)	4.30	1.3	278.8

By design, we only consider private keys whose HNF is minimal. To this end, we adopted a rejection sampling strategy, generating uniformly random private keys and

discarding those that do not satisfy the desired property, until finding one that does.

Interestingly, prime values of n tend to yield lattices with minimal HNF far more often than composite n. Empirically, the probability that a random circulant matrix A has a minimal HNF is heavily affected by the choice of lattice parameters, particularly its dimension n, being roughly $O(1/D)$ where D is the number of irreducible factors of $x^n - 1$. Tourloupis [24] addresses this issue (for a generic matrix A, not necessarily circulant) by sieving the randomly sampled A to have prime or near-prime determinant, thus ensuring that it has a 99% probability of sporting a minimal HNF. However, choosing n itself to be prime increases that probability to the same level (since the number of irreducible factors of $x^n - 1$ coincide with the number of factors of n), without having to resort to primality testing during key generation. This behavior is only counterbalanced for composite n when the FFT is available, in which case processing is fast enough to roughly compensate for the rejection sampling overhead.

Key sizes are essentially the same in our proposal for a given dimension n regardless of the choice of $p(x)$. Sample public key sizes are listed on Table 2.

Table 2. Public key sizes (in bits)

source	(n, σ, h, k)	$\lvert pk \rvert$
[6]	$(350, 256, 526, 64)$	1157800
ours	$(353, 256, 526, 64)$	6682
[6]	$(400, 256, 601, 64)$	1543200
ours	$(401, 256, 601, 64)$	7738
[6]†	$(512, 256, 769, 80)$	2621440
ours	$(512, 256, 769, 80)$	10240

† Inferred.

7. Conclusion

We have shown how to enhance the GGH-YK-M scheme by Barros and Schechter, reducing its public key size by an order of complexity from $O(n^2 \lg n)$ down to $O(n \lg n)$ bits. The bandwidth savings stem from the technique first put forward by Smart and Vercauteren technique, which we optimize in a simpler and more efficient way than the Gentry-Halevi method. As a result, key generation times decrease as compared to the Barros-Schechter variant by more than 3 orders of magnitude. Besides the key generation speedup, encryption becomes almost 2 orders of magnitude faster; decryption is about twice as fast though the reason for the improvement in this particular operation could be simply related to different implementation details. Our benchmarks were obtained using Java; a C/C++ implementation is likely to improve the timings even more.

An intriguing line for follow-up research is to assess the impact of extending the proposed method to lattices whose HNF is only near-minimal, say, having the two rightmost columns in nontrivial form. This is observed far more often than a minimal HNF when n is composite and might reduce key generation times considerably. The key and cryptogram sizes remain the same, because the sizes of the elements on each row of those nontrivial columns are bound by complementary factors of the determinant. While there are more factors to tackle for encryption and decryption, they are also smaller than in the minimal HNF case, so processing might end up being faster for those operations as well.

Our results show that the proposed techniques constitute a viable option to help minimize the cost of the GGH-YK-M scheme, and possibly for other lattice-based protocols, regarding both key size and processing times. We leave the application of the proposed techniques to somewhat (levelled) homomorphic encryption as a further research problem.

References

[1] Abraham Berman and Robert J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*, volume 9. Society for Industrial and Applied Mathematics (SIAM), 1994.

[2] D. J. Bernstein, J. Buchmann, and E. Dahmen. *Post-Quantum Cryptography*. Springer, Heidelberg, Deutschland, 2008.

[3] Daniel J. Bernstein. A subfield-logarithm attack against ideal lattices. Blog entry, February 2014. `http://blog.cr.yp.to/20140213-ideal.html`.

[4] Massimo Chenal and Qiang Tang. On key recovery attacks against existing somewhat homomorphic encryption schemes. In *International Conference on Cryptology and Information Security in Latin America – Latincrypt 2014*, Lecture Notes in Computer Science. Springer, 2014. To appear.

[5] Henri Cohen. *A course in computational algebraic number theory*, volume 138. Springer, 1993.

[6] Charles F. de Barros and L. Menasché Schechter. GGH may not be dead after all. In *XXXV Congresso Nacional de Matemática Aplicada e Computacional – CNMAC 2014*. Sociedade Brasileira de Matemática Aplicada e Computacional – SBMAC, 2014.

[7] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology – Crypto 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 537–554, Santa Barbara, USA, 1999. Springer.

[8] D. J. H. Garling. *Inequalities: A Journey into Linear Analysis*. Cambridge, 2007.

[9] Craig Gentry and Shai Halevi. Implementing Gentry's fully-homomorphic encryption scheme. In *Advances in Cryptology – Eurocrypt 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer, 2011.

[10] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology – CRYPTO '97*, pages 112–131. Springer, 1997.

[11] J. Hoffstein, J. Pipher, and J. Silverman. NTRU: A ring-based public key cryptosystem. In J. P. Buhler, editor, *Algorithmic Number Theory*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer Berlin Heidelberg, Oregon, USA, 1998.

[12] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[13] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In *Topics in Cryptology – CT-RSA 2011*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339, San Francisco, CA, USA, 2011. Springer.

[14] Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren. On CCA-secure somewhat homomorphic encryption. In *International Conference on Selected Areas in Cryptography – SAC 2011*, volume 7118 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2012.

[15] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2006.

[16] Robert J McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, 42(44):114–116, 1978.

[17] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 356–365. IEEE, 2002.

[18] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

[19] Phong Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from CRYPTO '97. In *Advances in Cryptology – CRYPTO '99*, pages 288–304. Springer, 1999.

[20] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[21] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical computer science*, 53(2):201–224, 1987.

[22] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, 1995.

[23] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In P. Q. Nguyen and D. Pointcheval, editors, *Public Key Cryptography – PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.

[24] Vasilios Evangelos Tourloupis. Hermite normal forms and its cryptographic applications. Master's thesis, University of Wollongong, 2013.

[25] M. Yoshino and N. Kunihiro. Improving GGH cryptosystem for large error vector. In *International Symposium on Information Theory and its Applications – ISITA 2012*, pages 416–420. IEEE, 2012.

