Cryptanalysis of Homomorphic Encryption Schemes based on the Aproximate GCD Problem

Tikaram Sanyashi[a], Darshil Desai[b] and Bernard Menezes[c]
Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Mumbai, India

Keywords: Cloud Computing, Symmetric Key Encryption, Homomorphic Encryption, Approximate Greatest Common Divisor Problem, Lattices, Orthogonal Lattice Attack.

Abstract: Economies of scale make cloud computing an attractive option for small and medium enterprises. However, loss of data integrity or data theft remain serious concerns. Homomorphic encryption which performs computations in the encrypted domain is a possible solution to address these concerns. Many partially homomorphic encryption schemes that trade off functionality for lower storage and computation cost have been proposed. However, not all these schemes have been adequately investigated from the security perspective. This paper analyses a suite of such proposed schemes based on the hardness of the Approximate GCD problem. We show that two of these schemes are vulnerable to the Orthogonal Lattice attack. The execution time of the attack is a function of various parameters including message entropy. For the recommended set of parameters, the execution time of the attack is no greater than 1 day on a regular laptop.

1 INTRODUCTION

Large scale cloud computing obviates the need for significant investment in local computing resources. However, storing sensitive information such as financial or medical data in the cloud environment may result in loss of data integity and/or data theft. One possible solution to maintaining data confidentiality is to store user data in encrypted form. To perform operations on the data, it would need to be decrypted in the cloud. This would expose the plaintext to the cloud service provider and could compromise the privacy of the data. Another possibility is to communicate the encrypted data to the client who would decrypt it and then perform the necessary computations on the data. But that would defeat the goal of outsourcing computation to the cloud. An attractive alternative is to perform the operations on the cloud in the encrypted domain. Cryptography offers a solution to this problem in the form of homomorphic encryption.

The notion of homomorphic encryption dates back to 1978 when Rivest et al. (Rivest and Dertouzos, 1978) proposed it and called it "Privacy Homomorphism". Through time, several encryption schemes were introduced. These were partially homomorphic - some were homomorphic with respect to addition and some with respect to multiplication. Examples include (Goldwasser and Micali, 1982), (ElGamal, 1985) (Benaloh, 1994) and (Paillier, 1999).

In 2009, the problem was finally addressed and solved by Craig Gentry (Gentry et al., 2009) using ideal lattices. Implementation of the solution required very high storage and computation time. Even for a single bit encryption, 2.3 Gigabytes and a bootstrapping time of 30 minutes were required with lattice dimension over 32,000 making it impractical.

Many homomorphic schemes were proposed based on the hardness of the Learning With Error (LWE) and Ring-Learning With Error (R-LWE) problems. These include (Fan and Vercauteren, 2012; Brakerski, 2012; Bos et al., 2013; Brakerski and Vaikuntanathan, 2014; Brakerski et al., 2014; Costache and Smart, 2016). Also, some schemes were constructed based on the hardness assumption of integer GCD problem (Van Dijk et al., 2010; Coron et al., 2011).

A drawback of many homomorphic encryption schemes is the bit-by-bit encryption - each block of ciphertext corresponds to a single bit of plaintext. Moreover, computation tasks need to be converted to binary addition and multiplication making the encryption scheme even more complicated. To reduce com-

[a] https://orcid.org/0000-0002-7434-0468
[b] https://orcid.org/0000-0003-3283-3560
[c] https://orcid.org/0000-0003-2997-9286

Sanyashi, T., Desai, D. and Menezes, B.
Cryptanalysis of Homomorphic Encryption Schemes based on the Aproximate GCD Problem.
DOI: 10.5220/0008071605170522
In *Proceedings of the 16th International Joint Conference on e-Business and Telecommunications (ICETE 2019)*, pages 517-522
ISBN: 978-989-758-378-0

plexity ciphertext packing techniques were studied. For example, Peikert's SIMD technique (Peikert et al., 2008), Yasuda's ideal lattice technique (Yasuda et al., 2013), Cheon's preprocessing of binary vector before encryption technique (Cheon et al., 2013), Smart and Vercuteren's polynomial-CRT technique (Smart and Vercauteren, 2014) etc. However, these schemes are still not efficient enough to be acceptable in practice.

Somewhat Homomorphic Encryption (SHE) schemes support a limited set of homomorphic computations. One such SHE scheme was proposed by Dyer et al. (Dyer et al., 2017). They presented a suite of four symmetric key encryption schemes - HE1 and HE2 are suitable for large entropy data while their variants, HE1N and HE2N are appropriate for small entropy data. The entropy represents the number of bits present in the message. HE2N also has the flexibility to be generalized to l-dimensions with an increased level of security.

The encryption scheme is based on the hardness of the Approximate Greatest Common Divisors problem (AGCD) (Howgrave-Graham, 2001). Given m approximate multiples of p, $c_i = pq_i + r_i$ with small r_i, the approximate greatest common divisor p needs to be recovered. The usual way to recover p is to guess any two of $r_1, \cdots, r_m$ then compute greatest common divisor of them as GCD($c_i - r_i$,$c_j - r_j$). If the r_i's are sufficiently small then the secret p can be recovered easily using brute force search. However, if the perturbation r_i is in the vicinity of p, then it is clearly impossible to reconstruct p from the given information.

The main contribution of this work is to demonstrate that two of the four proposed schemes, HE1 and HE1N are insecure. Even with a fairly small number of ciphertexts, the plaintext and secret key can be deduced using the Orthogonal Lattice attack. The execution time of the attack is a function of two main parameters - the entropy of the message, ρ and the maximum number of homomorphic operations, d that can be performed on a given message before decryption fails. For small parameter settings, the execution time is a few minutes and increases to a few hours for larger parameter values.

Notations: In this paper, uppercase bold letters are used to represent matrices, lowercase bold are for vectors and regular lowercase are for constants.

$x \overset{\$}{\leftarrow} Q$ denotes x is chosen uniformly at random from space Q. lg denotes log base 2.

The paper is organized as follows. Section 2 contains background material related to lattices. Section 3 summarizes the encryption schemes presented in (Dyer et al., 2017) including the variants targeted in our attack. In section 4, we present the cryptanalytic attack on two of those schemes. We also include the time to execute the attack. Section 5 concludes the paper.

2 BACKGROUND

A lattice is a discrete (additive) subgroup of $\mathbb{R}^n$. In particular, any subgroup of $\mathbb{Z}^n$ is a special kind of lattice referred to as an integer lattice. Let $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_m \in \mathbb{Z}^n$, $n \geq m$ be linearly independent. The lattice, $\mathcal{L}$, spanned by integer linear combinations of $\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m$ is

$$\mathcal{L}(\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_m) = \left\{ \sum_{i=0}^{m} x_i \boldsymbol{b}_i \; : \; x_i \in \mathbb{Z} \right\}$$

The set of vectors $\boldsymbol{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2 \cdots, \boldsymbol{b}_m)$ is called a basis of lattice $\mathcal{L}$, n, rank of lattice and m, dimension of lattice. A full-rank lattice is one for which $m = n$.

The determinant of a lattice is n-dimensional volume of its fundamental parallelepiped, computed as det($\boldsymbol{L}$)= $\sqrt{det(\boldsymbol{B}\boldsymbol{B}^T)}$, where $\boldsymbol{B}^T$ is the transpose of $\boldsymbol{B}$. A lattice can have multiple bases spanning the same lattice. The determinant of a lattice is independent of the choice of basis.

Lattice reduction is often a key step in solving problems based on lattices. It is used to find a basis with short and nearly orthogonal vectors. The quality of the basis obtained from a reduction algorithm is determined by the Hermite factor δ_0^n where

$$||b_0|| = \delta_0^n vol(\boldsymbol{L})^{\frac{1}{n}}$$

Here, b_0 represents the shortest non-zero vector after lattice reduction. The smaller the Hermite factor, the higher is the quality of the reduced basis.

The two main lattice reduction algorithms are LLL and BKZ. BKZ2.0 is an optimized version of BKZ. BKZ behaves differently based on block size k. For $k = 2$ the algorithm runs in polynomial time and outputs a basis equivalent to an LLL-reduced basis. An increase in block size improves the quality of the reduced basis but takes more time. In practice, the run time of BKZ increases rapidly with block size and becomes practically infeasible for $k > 30$ or so. BKZ2.0 can handle a much larger block size and results in a greatly reduced basis compared to BKZ.

Implementations of LLL, BKZ and BKZ 2.0 are available in many software packages. In our implementations we have used SageMath and the fplll libraries.

3 ENCRYPTION SCHEMES TARGETED FOR ATTACK

The homomorphic encryption scheme presented in (Dyer et al., 2017) is a 5-tuple (KeyGen, Enc, Dec, Add, Mult). Based on its entropy, ρ, there are two ways to encrypt a message, m, - HE1 and HE2. If ρ is less than 32 bits, then HE1N or its more secure version, HE2N, is used. A message with entropy greater than 32 bits is encrypted with HE1 or its more secure version HE2.

In the KeyGen step, two distinct large primes p and q are employed. A security parameter, $\lambda = \frac{3d\rho}{2}$, (measured in bits) is selected and η is calculated as $\eta = \frac{3d\lambda}{2} - \lambda$. The ranges of p and q are determined by $p \in [2^{\lambda-1}, 2^{\lambda}]$ and $q \in [2^{\eta-1}, 2^{\eta}]$. pq is publicly known while p is the secret key. To prevent the factorization of pq via Coppersmith's method (Coppersmith, 1997) or any other, a large value of λ is used.

The ciphertext, c of an integer, m, in HE1 is obtained by computing

$$Enc(m,p) = m + rp \ (mod\ pq) \tag{1}$$

where $r \overset{\$}{\leftarrow} [1,q)$ is an ephemeral key.

Decryption involves a simple modulo p operation

$$Dec(c,p) = c \ (mod\ p)$$

It is clear from 1 that a message with small entropy, ρ, is vulnerable to a brute force attack. Therefore to encrypt such a message, a modified scheme, HE1N, is used - this adds an extra noise term during encryption to increase entropy and thereby thwart a possible brute force attack. The secret key for the encryption scheme is now the pair, (k, p). Here, k is an integer and its size is a security parameter. Key generation for HE1N is as in HE1 but with the following modification

$$\lambda = \frac{3d\rho'}{2} \qquad \eta = \frac{3d\lambda}{2} - \lambda \quad \text{where } \rho' = \rho + lg\, k.$$

Encryption of an integer, m using HE1N is computed using

$$c = Enc(m,p,k) = m + [p,k]\begin{bmatrix} r \\ s \end{bmatrix} \ (mod\ pq)$$

where $r \overset{\$}{\leftarrow} [1,q)$ and $s \overset{\$}{\leftarrow} [0,k)$.

Message m is obtained from the ciphertext c by performing the following two modulo computations in sequence

$$Dec(c,sk) = (c \ mod\ p) \ mod\ k$$

HE2 and HE2N are the more secure counterparts of HE1 and HE1N respectively. Encryption of messages with HE2 and HE2N is similar to that with HE1 and HE1N but the new ciphertext is a vector of two components with added randomness to make it more secure.

The key generation process of HE2 and HE2N is also similar to that of HE1 and HE1N. It involves sampling of two prime numbers, $a_i \overset{\$}{\leftarrow} [1, pq)$, $i \in \{1,\ 2\}$ with a_1, a_2, $(a_1 - a_2) \neq 0$ (mod p and mod q). It computes a re-encryption matrix $\mathbf{R}$ (for homomorphic multiplication) as

$$\begin{bmatrix} 1-2\alpha_1 & \alpha_1 & \alpha_1 \\ -\alpha_2 & \alpha_2+1 & \alpha_2 \end{bmatrix}$$

where,
$\alpha_1 = \beta^{-1}(\sigma a_1 + \rho p - a_1^2)$,
$\alpha_2 = \beta^{-1}(\sigma a_2 + \rho p - a_2^2)$,
with $\beta = 2(a_2 - a_1)^2$, $\rho \overset{\$}{\leftarrow} [0,q]$ and $\sigma \overset{\$}{\leftarrow} [0,pq)$

The ciphertext of an integer m in HE2 is a vector of two components calculated as

$$\mathbf{c} = Enc(m,sk) = m\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} p & a_1 \\ p & a_2 \end{bmatrix}\begin{bmatrix} r \\ s \end{bmatrix} \ (mod\ pq)$$

Here $r \overset{\$}{\leftarrow} [1,q)$ and $s \overset{\$}{\leftarrow} [1,pq)$ are independent ephemeral secrets for each encryption resulting in different ciphertexts for the same message m.

Decryption of ciphertext $\mathbf{c} = (c_1, c_2)$ involves computation of

$$Dec(\mathbf{c},sk) = \gamma^T * \mathbf{c} \ (mod\ p),$$

where $\gamma^T = (a_2 - a_1)^{-1}[a_2\ , -a_1]$.

The ciphertext with HE2N is a vector of two components computed as

$$\mathbf{c} = Enc(m,sk) = m\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} p & k & a_1 \\ p & k & a_2 \end{bmatrix}\begin{bmatrix} r \\ s \\ t \end{bmatrix} \ (mod\ pq)$$

Here $r \overset{\$}{\leftarrow} [1,q)$, $s \overset{\$}{\leftarrow} [1,k)$ and $t \overset{\$}{\leftarrow} [1,pq)$.

Decryption of ciphertext $\mathbf{c} = (c_1, c_2)$ involves computation of

$$Dec(\mathbf{c},sk) = \gamma^T \mathbf{c} \ (mod\ p) \ mod\ k$$

where γ is defined as in the case of HE2. The encryption scheme can be generalized to $l-$dimensions as

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_l \end{bmatrix} = \begin{bmatrix} m \\ m \\ \vdots \\ m \end{bmatrix} + \begin{bmatrix} p & a_{11} & \cdots & a_{1(l-1)} \\ p & a_{21} & \cdots & a_{2(l-1)} \\ \vdots & \vdots & \cdots & \vdots \\ p & a_{l1} & \cdots & a_{l(l-1)} \end{bmatrix}\begin{bmatrix} r \\ s_1 \\ \vdots \\ s_{(l-1)} \end{bmatrix} \ (mod\ pq)$$

$$\boldsymbol{c} = \boldsymbol{m} + \boldsymbol{A}\boldsymbol{s} \ (mod\ pq)$$

The above equality resembles LWE but there are some major differences. In LWE, the cipher $\boldsymbol{c}$ and matrix $\boldsymbol{A}$ are known but here $\boldsymbol{A}$ itself is a secret and $\boldsymbol{s}$ is an ephemeral key which makes it even harder.

The hardness increases as dimension l increases but increasing l progressively makes the encryption scheme impractical. Therefore there is a tradeoff between hardness and practicality. When $l = 1$ the problem converges to AGCD problem which can be solved for the parameter setting provided in (Dyer et al., 2017). In the next section, we show how HE1 and its low entropy version HE1N may be compromised.

4 CRYPTANALYSIS

In this section, we focus on attacking HE1 and HE1N using the Orthogonal Lattice Attack.

4.1 Attacking HE1 and HE1N

We explored two lattice-based techniques to recover the secret key of the encryption scheme viz. Orthogonal Lattice attack (OLA) and Simultaneous Diophantine Approximation Attack (SDA). In each case, we recovered the plaintext and this further leads to recovery of the secret key p.

Both of these attacks perform well in practice though OLA runs comparatively faster compared to SDA. In the interest of brevity, we focus on the OLA Attack here.

The notion of the Orthogonal Lattice was first introduced by Nguyen and Stern (Nguyen and Stern, 2001) to crack the Qu-Vanstone cryptosystem. Since then it has been used for cryptanalysis of various other cryptosytems. Appendix B.1 of (Van Dijk et al., 2010) also mentioned a way to recover vectors orthogonal to cipher text $(c_1, \cdots, c_z)$, where $(c_1, \cdots, c_z)$ represents ciphertext corresponding to message $(m_1, \cdots, m_z)$. The idea is that a lattice orthogonal to $(c_1, \cdots, c_z)$ contains a sublattice orthogonal to both $(m_1, \cdots, m_z)$ and $(q_1, \cdots, q_z)$. We used techniques of (Van Dijk et al., 2010) to break the encryption schemes, HE1 and HE1N.

4.2 Attack Details

Construct a lattice, $\mathcal{L}$ spanned by the rows of following $z \times (z+1)$ basis matrix $\boldsymbol{T}$.

$$\boldsymbol{T} = \begin{bmatrix} c_1 & M_1 & & & \\ c_2 & & M_2 & & \\ \vdots & & & \ddots & \\ c_z & & & & M_z \end{bmatrix}$$

c_i is the ciphertext corresponding to message m_i and M_i is an upper bound on the value of m_i. Any vector $\boldsymbol{u} = (u_0, u_1, \cdots, u_z)$ in the lattice $\boldsymbol{T}$ can be represented by

$$\begin{aligned}\boldsymbol{u} &= (\alpha_1, \cdots, \alpha_z)\boldsymbol{T} \\ &= (\sum_{i=1}^{z} \alpha_i c_i, \alpha_1 M_1, \cdots, \alpha_z M_z)\end{aligned}$$

for some integer values α_i. The main observation is that

$$\begin{aligned} u_0 - \sum_{i=1}^{z} \frac{u_i}{M_i}.m_i &= \sum_{i=1}^{z} \alpha_i.c_i - \sum_{i=1}^{z} \frac{\alpha_i M_i}{M_i}.m_i \pmod p \\ &= \sum_{i=1}^{z} \alpha_i(c_i - m_i) \pmod p \\ &= 0 \end{aligned}$$

As the value of p is not known beforehand, we want a vector that satisfies

$$\begin{aligned} u_0 - \sum_{i=1}^{z} \frac{u_i}{M_i}.m_i &\le |u_0| + |\sum_{i=1}^{z} \frac{u_i}{M_i}.m_i| \\ &\le \sum_{i=0}^{z} |u_i| \le \frac{p}{2} \end{aligned}$$

The determinant of the Gram matrix $\boldsymbol{T}\boldsymbol{T}^T$ is bounded by product of the norms of the columns of matrix $\boldsymbol{T}$, which is $M^z.\sqrt{c_1^2 + c_2^2 + \cdots + c_z^2}$. According to Gaussian heuristic the shortest vector is as short as $\sqrt{z}M.\sqrt[z]{c_1^2 + c_2^2 + \cdots + c_z^2}$. To obtain the shortest vector $\boldsymbol{u}$, we need this to be as short as p. The condition to obtain such a vector can be formulated as

$$M.\sqrt[z]{c} < p$$

where $c = c_1^2 + c_2^2 + \cdots + c_z^2$,

The above gives a lower bound on the number of ciphertexts, z, needed for plaintext recovery as $z > \frac{\gamma}{\lambda - \rho}$. Here, γ, λ, ρ represent the number of bits in ciphertext c secret key p and message m respectively. When $z > \frac{\gamma}{\lambda - \rho}$, then lattice reduction is able to recover the short vectors in $\boldsymbol{T}$ subject to constraints on computation time.

Lattice reduction is performed on the basis matrix, $\boldsymbol{T}$. The first row of $\boldsymbol{T}$ is $(1, -\frac{m_1}{M_1}, \cdots, -\frac{m_z}{M_z})$. From this we recovered the plaintexts, m_i. The secret key, p, is obtained by computing $GCD(c_i - m_i, c_j - m_j)$.

4.3 Results

We implemented the attack of the previous section and ran it for several parameter sets including those mentioned in (Dyer et al., 2017). All experiments were performed on Intel i5 Gen 4, with 3.5 GHz clock and 8 GB DRAM running Ubuntu 16.04 64-bit LTS.

We were successful in recovering the message and later the secret key itself for all parameter values in

Table 1. The total time is dominated by the time for lattice reduction and increases greatly with message size and d. For the highest parameter setting in (Dyer et al., 2017), we could recover the message and the key in about 18 hours.

Table 1: Attack time for recovering plaintext in HE1 encryption scheme.

d	ρ	Attack time(Min)
2	32	.25
2	64	3.5
2	128	75
3	32	.25
3	64	4
3	128	430
4	32	.5
4	64	20
4	128	1070

For attacking HE1N, we followed the same technique as for HE1. We first used the Orthogonal Lattice attack to recover $m+ks$ from $c=m+ks+pr$ (Attack 1) and later the same technique to recover m from $m+ks$ (Attack 2).

For the highest parameter setting mentioned in (Dyer et al., 2017) we could recover the message and hence the secret key p in about 20 hours as shown in Table 2.

Table 2: Attack time for recovering plaintext in HE1N encryption scheme.

d	ρ	ρ'	Attack1(Min)	Attack2(Sec)
2	1	32	.25	.0106
2	1	64	13	.0103
2	1	128	460	.0359
2	8	32	.30	.0122
2	8	64	16	.0134
2	8	128	470	.0133
2	16	64	20	.0122
2	16	128	490	.0125
3	1	32	.28	.0126
3	1	64	22	.0105
3	1	128	510	.0125
3	8	32	.32	.0125
3	8	64	22	.0142
3	8	128	520	.0144
3	16	64	30	.0502
3	16	128	1195	.0508

5 CONCLUSION

In this paper, we studied the security of the encryption schemes presented in (Dyer et al., 2017). These schemes are based on the hardness of the AGCD problem. We investigated the possible application of the Orthogonal Lattice attack to crack these schemes. Our experiments indicate that the parameter values recommended for the HE1 and HE1N schemes are too small to resist attack. The plaintext and key for these schemes may be recovered within a day on a regular laptop even for the highest parameter settings.

Encryption schemes HE2 and HE2N are also based on the hardness of the AGCD problem. However, as of now, we are unable to find any attack techniques to crack these and its subsequent versions.

REFERENCES

Benaloh, J. (1994). Dense probabilistic encryption. In *Proceedings of the workshop on selected areas of cryptography*, pages 120–128.

Bos, J. W., Lauter, K., Loftus, J., and Naehrig, M. (2013). Improved security for a ring-based fully homomorphic encryption scheme. In *IMA International Conference on Cryptography and Coding*, pages 45–64. Springer.

Brakerski, Z. (2012). Fully homomorphic encryption without modulus switching from classical gapsvp. In *Annual Cryptology Conference*, pages 868–886. Springer.

Brakerski, Z., Gentry, C., and Vaikuntanathan, V. (2014). (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):13.

Brakerski, Z. and Vaikuntanathan, V. (2014). Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871.

Cheon, J. H., Coron, J.-S., Kim, J., Lee, M. S., Lepoint, T., Tibouchi, M., and Yun, A. (2013). Batch fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 315–335. Springer.

Coppersmith, D. (1997). Small solutions to polynomial equations, and low exponent rsa vulnerabilities. *Journal of Cryptology*, 10(4):233–260.

Coron, J.-S., Mandal, A., Naccache, D., and Tibouchi, M. (2011). Fully homomorphic encryption over the integers with shorter public keys. In *Annual Cryptology Conference*, pages 487–504. Springer.

Costache, A. and Smart, N. P. (2016). Which ring based somewhat homomorphic encryption scheme is best? In *Cryptographers' Track at the RSA Conference*, pages 325–340. Springer.

Dyer, J., Dyer, M., and Xu, J. (2017). Practical homomorphic encryption over the integers for secure computation in the cloud. In *IMA International Conference on Cryptography and Coding*, pages 44–76. Springer.

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472.

Fan, J. and Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144.

Gentry, C. et al. (2009). Fully homomorphic encryption using ideal lattices. In *Stoc*, volume 9, pages 169–178.

Goldwasser, S. and Micali, S. (1982). Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 365–377. Academic Press, New York.

Howgrave-Graham, N. (2001). Approximate integer common divisors. In *International Cryptography and Lattices Conference*, pages 51–66. Springer.

Nguyen, P. Q. and Stern, J. (2001). The two faces of lattices in cryptology. In *International Cryptography and Lattices Conference*, pages 146–180. Springer.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer.

Peikert, C., Vaikuntanathan, V., and Waters, B. (2008). A framework for efficient and composable oblivious transfer. In *Annual international cryptology conference*, pages 554–571. Springer.

Rivest, R., A. L. and Dertouzos, M. (1978). Paper. In *On Data Banks and Privacy Homomorphisms. Foundations of Secure Computation, ed. by RA DeMillo, et. al.* Academic Press, New York.

Smart, N. P. and Vercauteren, F. (2014). Fully homomorphic simd operations. *Designs, codes and cryptography*, 71(1):57–81.

Van Dijk, M., Gentry, C., Halevi, S., and Vaikuntanathan, V. (2010). Fully homomorphic encryption over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 24–43. Springer.

Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., and Koshiba, T. (2013). Packed homomorphic encryption based on ideal lattices and its application to biometrics. In *International Conference on Availability, Reliability, and Security*, pages 55–74. Springer.