A Lattice Attack on Homomorphic NTRU with Non-invertible Public Keys

Soyoung Ahn[1], Hyang-Sook Lee[1](✉), Seongan Lim[2](✉), and Ikkwon Yie[3](✉)

[1] Department of Mathematics, Ewha Womans University, Seoul, Korea
syahn921@gmail.com, hsl@ewha.ac.kr

[2] Institute of Mathematical Sciences, Ewha Womans University, Seoul, Korea
seongannym@ewha.ac.kr

[3] Department of Mathematics, Inha University, Incheon, Korea
ikyie@inha.ac.kr

Abstract. In 2011, Stehlé and Steinfeld modified the original NTRU to get a provably IND-CPA secure NTRU under the hardness assumption of standard worst-case problems over ideal lattices. In 2012, López-Alt et al. proposed the first multikey fully homomorphic encryption scheme based on the IND-CPA secure NTRU. Interestingly, this homomorphic NTRU and subsequent homomorphic variants of NTRU removed the condition 'invertible public key' of the underlying IND-CPA secure NTRU. In this paper, we investigate the security influence of using non-invertible public key in the homomorphic NTRU. As a result, we present how to mount a lattice attack to message recovery for the homomorphic NTRU when the public key is non-invertible. Our result suggests that using invertible public keys in the homomorphic NTRU is necessary for its security.

Keywords: NTRU · Homomorphic NTRU · IND-CPA security
Lattices · LLL algorithm

1 Introduction

The NTRU encryption scheme designed by Hoffstein et al. [6] is considered as a reasonable alternative to the public key encryption schemes based on either integer factorization or discrete logarithm. Since its first introduction, minor changes of the parameter to avoid known attacks have been added. Even with its computational efficiency and standardization of the NTRU [11], a provably secure version was not known until Stehlé et al. proposed a modification of the original NTRU in the year 2011 [10]. The IND-CPA security of their modification is proven in the standard model under the hardness assumption of standard worst-case problems over ideal lattices [10]. Reflecting the continued progress in the research on quantum computing, researches on transitioning to quantum resistant algorithms become very active. Moreover, NIST has initiated a standardization process in post-quantum cryptography. The IND-CPA secure version of NTRU could be a strong candidate for the standardization of post-quantum public key encryption. The security proof of the IND-CPA secure NTRU was

S. Qing et al. (Eds.): ICICS 2017, LNCS 10631, pp. 248–254, 2018.
https://doi.org/10.1007/978-3-319-89500-0_22

given in [10] under the assumption that the public key is an invertible polynomial in $R_q = \mathbb{Z}[x]/\langle q, x^n+1\rangle$, however, no such result is known for 'non-invertible' public key. López-Alt et al. observed that the IND-CPA secure NTRU can be made fully homomorphic and proposed the first multikey homomorphic encryption scheme for a bounded number of users [8]. Notably, the homomorphic NTRU [8] and its subsequent versions [3,9] do not assume invertible public keys. If q is a prime number and n is a power of 2 with $q = 1 \bmod 2n$, then there is a ring isomorphism between R_q and $\mathbb{Z}_q^n$ and the number of non-invertible elements in R_q is $q^n - (q-1)^n$.

In this paper, we investigate the security influence of using non-invertible public key in the homomorphic NTRU. We present a very effective lattice attack for message recovery on the homomorphic NTRU when the public key is not invertible. The message space of the homomorphic NTRU is $\{0, 1\}$ which implies that the IND-CPA security is equivalent to the security against the message recovery attack. We interpret the message recovery attack as solving a system of linear equations under some condition over a finite field $\mathbb{Z}_q$ using $\beta(x) = \frac{x^n+1}{\gcd(h(x), x^n+1)} \in \mathbb{Z}_q[x]$ for any non-invertible public key $pk = h(x)$. For a proof of successful message recovery in general, we used a sequence of sublattices of the target lattice and showed that there is an optimal sublattice which gives the desired short vector by the LLL algorithm if the degree of $\deg \beta(x) \leq \frac{\log q}{4}$ in the homomorphic NTRU. Moreover, it is known that the actual shortest output vector of the LLL algorithm could be much shorter than its theoretical bound. In fact, our experiments using MLLL(Modified LLL) in [4] give much shorter vector than the theoretical bound and this suggests that avoiding $\beta(x)$ to have small degree is not enough to guarantee the security of the homomorphic NTRU under message recovery attack. Therefore we conclude that setting the public key of the homomorphic NTRU as an invertible polynomial in R_q is desirable since the security against message recovery attack is a minimal requirement for encryption scheme. We note that some lattice attacks called by the subfield attacks on NTRU cryptosystem were proposed by Cheon et al. [5] and Albrecht et al. [1] and the goal of the subfield attack is to recover private key which can be understood as a short vector of the NTRU lattice. Their subfield attacks are based on the fact that there exist subfields that allow to reduce the dimension of the NTRU lattice and successful when the modulus q is exponential in n. Contrary to [1,5], the goal of our lattice attack is the message recovery when the public key is non-invertible.

The rest of the paper is organized as follows. In Sect. 2, we review some basics of this paper. In Sect. 3, we show that how to mount the message recovery attack to be successful if the public key is not invertible. In Sect. 4, we conclude our paper.

2 Preliminaries

2.1 The Basic Scheme of Homomorphic NTRU

The homomorphic NTRU is defined on the ring $R_q = \mathbb{Z}[x]/\langle q, x^n + 1\rangle$ for q is a prime number and n is a power of two. Any element $k(x) \in R_q$ is represented as $k(x) = \sum_{i=0}^{n-1} k_i x^i$, where $-\frac{q}{2} < k_i < \frac{q}{2}$. For the ring $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$, we denote $k(x) \leftarrow \chi_\epsilon$ for an appropriate distribution χ_ϵ and each coefficient $|k_i| \leq \epsilon$ of $k(x)$ if $k(x) \leftarrow \chi_\epsilon$. In the homomorphic version in [8], it is assumed that $q = 2^{n^\delta}$ with $0 < \delta < 1$ and the message space is $\{0, 1\}$ while it was considered that $q = poly(n)$ with the message space $\{0, 1\}^n$ in the proven IND-CPA secure version [10]. The basic scheme of the homomorphic NTRU consists of three polynomial time algorithms $\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$).

$\mathsf{KeyGen}(1^\kappa)$: Sample polynomials $\tilde{f}(x),\ g(x) \leftarrow \chi_\epsilon$, repeat sampling $\tilde{f}(x)$ until $f(x) := 2\tilde{f}(x) + 1$ is invertible in R_q and denote the inverse of $f(x)$ in R_q as $(f(x))^{-1}$. Output $pk = h(x) := 2g(x)(f(x))^{-1} \pmod{q, x^n + 1}$ and $sk = f(x)$.

$\mathsf{Enc}(pk, m \in \{0, 1\})$: Sample polynomials $s(x),\ e(x) \leftarrow \chi_\epsilon$, and output $c(x) := h(x)s(x) + 2e(x) + m \pmod{q, x^n + 1}$.

$\mathsf{Dec}(sk, c)$: Compute $\mu(x) = f(x)c(x) \pmod{q, x^n + 1}$, and output $m' = \mu(x) \pmod{2}$.

2.2 Lattices and LLL Algorithm

The lattice L is an additive subgroup of $\mathbb{R}^m$ that is $\mathbb{Z}$-generated by a set of n linearly independent vectors $\{\mathbf{b}_1, ..., \mathbf{b}_n\}$ in $\mathbb{R}^m$. We say n as the dimension of the lattice L which is denoted by $\dim(L)$. For a given lattice L, there is a geometric invariant called the minimum of the lattice and there are several computational problems related to the minimum.

Definition 1 (Minimum). *The (first) minimum of a lattice L is the norm of a shortest non-zero vector in L and denoted as $\lambda_1(L) = \min_{\mathbf{v} \in L \setminus \{\mathbf{0}\}} \|\mathbf{v}\|_2$ where $\| \cdot \|_2$ is the Euclidean norm of the vector.*

In [2], Ajtai proved that for a given lattice L, the problem of finding a vector of the minimum norm, which is called as the Shortest Vector Problem(SVP), is NP-hard. A relaxed SVP is a problem of finding a vector which is no longer than a factor of γ to the first minimum and these problems are often refer to as the approximate SVP$_\gamma$. Note that if γ increases, the problem gets easier. There is no known efficient algorithm solving the SVP$_\gamma$ for small γ in a lattice in arbitrary dimension even in quantum computer. The LLL algorithm is a polynomial time algorithm for SVP$_\gamma$ with $\gamma = 2^{\frac{n-1}{2}}$ [7]. Moreover, in theoretical view, the shortest vector $\mathbf{v}$ of the output vector of LLL algorithm for n dimensional lattice L satisfies that $||\mathbf{v}||_2 \leq 2^{\frac{n-1}{4}} \det(L)^{1/n}$. We note that the input of LLL algorithm should be a basis of the lattice. The MLLL is modified from LLL so that it works on any set of generating set of vectors of integer lattices [4].

3 Message Recovery of Homomorphic NTRU with Non-invertible Public Keys

The IND-CPA security of homomorphic NTRU was proven when the public key $h(x) = \frac{2g(x)}{f(x)} \in R_q$ is invertible in [10]. In this section, we consider the case that the public key $h(x)$ is not invertible in R_q. Because q is prime, we see that $\mathbb{Z}_q[x]$ is a unique factorization domain. If $h(x)$ is not invertible in R_q, then $\gcd(h(x), x^n + 1) = d(x) \neq 1$ in $\mathbb{Z}_q[x]$. Therefore, we see that $x^n + 1 = \beta(x)d(x)$ and $\gcd(\beta(x), h(x)) = 1$ in $\mathbb{Z}_q[x]$. Since $x^n + 1$ divides $\beta(x)h(x)$, we see that $\beta(x)h(x) = 0$ in R_q. For a given ciphertext $c(x) = h(x)s(x) + 2e(x) + m$, we see that $w(x) = \beta(x)c(x) \bmod (q, x^n + 1) = \beta(x)(2e(x) + m) \bmod (q, x^n + 1)$. In the homomorphic NTRU, the plaintext is chosen from $\{0, 1\}$, and therefore, its IND-CPA security is equivalent to the security in message recovery attack. Therefore, the IND-CPA adversarial goal is to recover $m \in \{0, 1\}$ from

$$w(x) = \beta(x)(2e(x) + m) \bmod (q, x^n + 1), \tag{1}$$

while m and $e(x)$ are unknown and $w(x)$ and $\beta(x)$ are known.

3.1 A Sufficient Condition for Message Recovery

For $\beta(x) = \frac{x^n+1}{\gcd(h(x), x^n+1)} = \sum_{i=0}^{\ell} \beta_i x^i \in \mathbb{Z}_q[x]$, we consider the following matrix $[B] \in \mathbb{Z}^{n \times n}$:

$$[B] = \begin{bmatrix} \beta_0 & \cdots & \cdots & \beta_\ell & \cdots & 0 \\ \vdots & \ddots & & & \ddots & \vdots \\ 0 & \cdots & \beta_0 & \cdots\cdots & & \beta_\ell \\ -\beta_\ell & \cdots & 0 & \beta_0 & \cdots & \beta_{\ell-1} \\ & \ddots & & & \ddots & \\ -\beta_1 & \cdots & -\beta_\ell & 0 & \cdots & \beta_0 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_{n-1} \\ \vdots \\ \mathbf{b}_\ell \\ \mathbf{b}_{\ell-1} \\ \vdots \\ \mathbf{b}_0 \end{bmatrix} \tag{2}$$

Note that the Eq. (1) can be represented by using matrices over $\mathbb{Z}_q$ for $e(x) = \sum_{i=0}^{n-1} e_i x^i$ and $w(x) = \sum_{i=0}^{n-1} w_i x^i$:

$$\mathbf{w} = [B] \cdot (2\mathbf{e} + \mathbf{m}) \bmod q \tag{3}$$

with $\mathbf{w} = [w_{n-1}, \ldots, w_0]^T$; $\quad 2\mathbf{e} + \mathbf{m} = [2e_{n-1}, \ldots, 2e_1, 2e_0 + m]^T$. Again, Eq. (3) of matrices can be written as

$$w_i = \langle \mathbf{b}_i, 2\mathbf{e} + \mathbf{m} \rangle \bmod q \text{ for all } i = 0, ..., n-1,$$

where $\langle \cdot, \cdot \rangle$ is the usual inner product of two vectors in $\mathbb{Z}^n$.

Theorem 1. *Suppose that $\mathbf{b}_i$'s are given as in Eq. (2) and a vector $\eta = (\eta_0, \ldots, \eta_{n-1}) \in \mathbb{Z}^n$ is known to satisfy the following condition:*

$$\mathsf{Condition(*)} \begin{cases} (i) \quad \eta = \sum_{i=0}^{n-1} \lambda_i \mathbf{b}_i \bmod q \text{ for } \lambda_i \in \mathbb{Z} \\ (ii) \quad |\eta_i| < \frac{q}{4n\epsilon+2} \text{ for all } i = 0, 1, \ldots, n-1 \\ (iii) \quad \eta_{n-1} = 1 \bmod 2 \end{cases}$$

For any given ciphertext $c(x)$, the plaintext $m \in \{0, 1\}$ can be recovered by $m = (\sum_{i=0}^{n-1} \lambda_i w_i \bmod q) \bmod 2$, where $w(x) = \beta(x)c(x) \bmod (q, x^n + 1) = \sum_{i=0}^{n-1} w_i x^i$.

Proof. For a given vector $\eta = (\eta_0, \ldots, \eta_{n-1}) = \sum_{i=0}^{n-1} \lambda_i \mathbf{b}_i \bmod q$ with the $\mathsf{Condition(*)}$ holds, we have

$$\sum_{i=0}^{n-1} \lambda_i w_i \bmod q = \langle \sum_{i=0}^{n-1} \lambda_i \mathbf{b}_i, 2\mathbf{e} + \mathbf{m} \rangle \bmod q = (\sum_{i=0}^{n-1} 2e_i \eta_i) + m\eta_{n-1} \bmod q.$$

From the assumptions $|\eta_i| < \frac{q}{4n\epsilon+2}$, $|e_i| \le \epsilon$ and $m \in \{0, 1\}$, we see that

$$|(\sum_{i=0}^{n-1} 2e_i \eta_i) + m\eta_{n-1}| < 2n\epsilon \frac{q}{4n\epsilon + 2} + \frac{q}{4n\epsilon + 2} = \frac{q(2n\epsilon + 1)}{4n\epsilon + 2} = q/2.$$

Therefore, we have $\sum_{i=0}^{n-1} \lambda_i w_i \bmod q = \left(\sum_{i=0}^{n-1} 2e_i \eta_i\right) + m\eta_{n-1}$, which implies that $(\sum_{i=0}^{n-1} \lambda_i w_i \bmod q) \bmod 2 = m$. □

Note that Theorem 1 works for any solution $(\lambda_i)_{0 \le i \le n-1}$ which is easy to compute from η by a simple linear algebra over $\mathbb{Z}_q$. Therefore, for a successful message recovery attack, it is enough to get a vector $\eta \in \mathbb{Z}^n$ that satisfies $\mathsf{Condition(*)}$.

3.2 A Lattice Attack for the Message Recovery

Now we present how to apply a lattice reduction algorithm, to find such a short vector η that is described in Theorem 1.

For the vectors $\mathbf{b}_i$'s as given in Eq. (2), we consider the lattice $L_B = \{\zeta \in \mathbb{Z}^n | \zeta = \sum_{i=0}^{n-1} x_i \mathbf{b}_i \bmod q$ for some $x_i \in \mathbb{Z}\}$. Now we describe the process of finding a short vector in L_B that satisfies $\mathsf{Condition(*)}$ in two ways. Firstly, we apply a lattice reduction algorithm MLLL [4] for the linearly dependent generating set of vectors

$$S = \{(q, 0, ..., 0), (0, q, 0, ..., 0), ..., (0, ..., 0, q), \mathbf{b}_{n-1}, ..., \mathbf{b}_0\} \subset \mathbb{Z}^n.$$

From our experiments, we see that the algorithm MLLL outputs a short vector with $\mathsf{Condition(*)}$ holds if the degree ℓ of $\beta(x)$ is small. However, the only thing we can prove on the size of the shortest vector of the output of MLLL is that it is at least smaller than $2^{\frac{n-1}{4}} (\det L_B)^{\frac{1}{n}} \le 2^{\frac{n-1}{4}} q^{\frac{\ell}{n}}$ from the LLL reducedness of the output. This does not give enough reason why a short vector from the output of MLLL satisfies the $\mathsf{Condition(*)}$.

Now we present a method of finding a short vector in L_B that with $\mathsf{Condition(*)}$ holds provably if the degree $\ell \le \frac{\log_2 q}{4}$. We consider a sequence of sublattices $L_{\ell+1} \subset L_{\ell+2} \subset \cdots \subset L_n \subset L_B$), where $L_i (\ell + 1 \le i \le n)$ is generated by the row vectors of $B_i \in \mathbb{Z}^{i \times n}$ which are defined as follows:

$$B_i = \begin{bmatrix} 0 \cdots 0 & q & \cdots & 0 & 0 & \cdots & 0 \\ \vdots \quad \vdots & & \ddots & & \vdots & & \vdots \\ 0 \cdots 0 & 0 & \cdots & q & 0 & \cdots & 0 \\ 0 \cdots 0 & \beta_0 & \cdots & \beta_{\ell-1} & 1 & \cdots & 0 \\ \vdots \quad \vdots & & \ddots & \ddots & \ddots & \ddots & \\ 0 \cdots 0 & 0 & \cdots & \beta_0 & \cdots & \beta_{\ell-1} & 1 \end{bmatrix} = [0_{i \times (n-i)} | B'_{i,\mathsf{red}}], \quad B'_{i,\mathsf{red}} \in \mathbb{Z}^{i \times i}$$

Let $L_{B'_{i,\mathsf{red}}} \subset \mathbb{Z}^i$ be the lattice generated by the row vectors of $B'_{i,\mathsf{red}}$ for $i = \ell+1, ..., n$. If $\eta_{\mathsf{red}} = (\eta'_{n-i}, ..., \eta'_{n-1}) \in L_{B'_{i,\mathsf{red}}}$ is a short vector that satisfies $\mathsf{Condition(*)}$ then $\eta = (\eta_j)_{0\leq j\leq (n-1)}$ is a short vector in L_B that satisfies $\mathsf{Condition(*)}$, where $\eta_j = 0$ if $0 \leq j \leq (n-i-1)$ and $\eta_j = \eta'_j$ if $n-i \leq j \leq n-1$. From [4], we see that the shortest vector $\mathbf{v}'_i \in L_{B'_{i,\mathsf{red}}}$ of the output of the LLL algorithm for the lattice generated by the row vectors of $B'_{i,\mathsf{red}}$ satisfies that

$$||\mathbf{v}'_i|| \leq ||\mathbf{v}'_i||_2 \leq 2^{\frac{i-1}{4}} \det(B'_{i,\mathsf{red}})^{1/i} = 2^{\frac{i-1}{4}} q^{\frac{\ell}{i}}.$$

By setting $\log_2 q = \tau$, we have a sequence of vectors $\mathbf{v}_i \in L_B$ with $||\mathbf{v}_i|| \leq 2^{\frac{i-1}{4}+\frac{\ell\tau}{i}}$ for $i = \ell+1, ..., n$. From a simple calculation over real numbers using the derivatives, we see that the function $f(i) = 2^{\frac{i-1}{4}+\frac{\ell\tau}{i}}$ has its minimum $2^{-1/4+\sqrt{\ell\tau}}$ at $i = 2\sqrt{\ell\tau}$. For simplicity, we assume that $\kappa = 2\sqrt{\ell\tau}$ is an integer. Therefore, the LLL algorithm applied on $L_{B'_{\kappa,\mathsf{red}}}$ on the basis consists of the row vectors of $B'_{\kappa,\mathsf{red}}$ gives a vector $\mathbf{v} \in L_B$ with $||\mathbf{v}|| \leq 2^{-1/4+\sqrt{\ell\tau}}$.

Now we want to show that this vector satisfies $\mathsf{Condition(*)}$ as long as the last component is an odd number. For this, it is enough to show that $2^{-1/4+\sqrt{\ell\tau}} \leq \frac{q}{4n\epsilon+2}$. From the equality $2^{-1/4+\sqrt{\ell\tau}} = 2^{-1/4}q^{\sqrt{\frac{\ell}{\tau}}}$, it is enough to show that $2^{-1/4}(4n\epsilon+2) \leq q^{1-\sqrt{\frac{\ell}{\tau}}}$. In particular, if q is subexponential in n as in the homomorphic NTRU, one can assume that $2^{-1/4}(4n\epsilon+2) \leq q^{1/2}$. Moreover, if $\ell \leq \frac{\log_2 q}{4} = \frac{\tau}{4}$, we clearly have $q^{1/2} \leq q^{1-\sqrt{\frac{\ell}{\tau}}}$ and thus $2^{-1/4}(4n\epsilon+2) \leq q^{1-\sqrt{\frac{\ell}{\tau}}}$. Therefore, we conclude that $2^{-1/4+\sqrt{\ell\tau}} \leq \frac{q}{4n\epsilon+2}$ if $\ell \leq \frac{\log_2 q}{4}$.

Note that the condition $\ell \leq \frac{\log_2 q}{4}$ to guarantee the desired shortness of the vector $\mathbf{v}$ is deduced from the theoretical bound of the shortest vector of the output of the LLL algorithm. It is known that the actual shortest vector of the LLL algorithm is shorter than the theoretical bound in general. Moreover, as in the example of the following section, the method using MLLL gives a shorter vector than the method using the sublattice. This suggests that the message recovery attack can be successful for much larger ℓ's. Therefore, setting $h(x)$ as an invertible polynomial in R_q is more appropriate than avoiding $\beta(x)$ with successful lattice reduction attack using sublattice as described above.

4 Conclusion

The IND-CPA security of the homomorphic NTRU is proven when the public key is invertible in R_q [10]. However, no result on the security of the homomorphic NTRU is known when the public key is not invertible. In this paper, we show that if the public key is not invertible in the homomorphic NTRU, then one can use a lattice reduction algorithm effectively to recover the plaintext of any ciphertext. Therefore, we conclude that the public key of homomorphic NTRU should be invertible in the ring R_q to guarantee the IND-CPA security of homomorphic variants of NTRU [3,8,9].

Acknowledgement. Hyang-Sook Lee and Seongan Lim were supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (Grant Number: 2015R1A2A1A15054564). Seongan Lim was also supported by Basic Science Research Program through the NRF funded by the Ministry of Science, ICT and Future Planning (Grant Number: 2016R1D1A1B01008562). Ikkwon Yie was supported by Basic Science Research Program through the NRF funded by the Ministry of Science, ICT and Future Planning (Grant Number: 2017R1D1A1B03034721).

References

1. Albrecht, M., Bai, S., Ducas, L.: A subfield lattice attack on overstretched NTRU assumptions. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 153–178. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_6
2. Ajtai, M.: The shortest vector problem in L_2 is NP-hard for randomized reductions. In: STOC 1998, pp. 10–19 (1998)
3. Bos, J.W., Lauter, K., Loftus, J., Naehrig, M.: Improved security for a ring-based fully homomorphic encryption scheme. In: Proceedings of IMA International Conference 2013, pp. 45–64 (2013)
4. Bremner, M.R.: Lattice Basis Reduction-An Introduction to the LLL Algorithm and its Applications. CRC Press, Boca Raton (2012)
5. Cheon, J.H., Jeong, J., Lee, C.: An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without an encoding of zero. Cryptology ePrint Archive, Report 2016/139 (2016)
6. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054868
7. Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring polynomials with rational coefficients. Mahtematische Ann. **261**, 513–534 (1982)
8. Lopez-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multyparty computation on the cloud via multikey fully homomorphic encryption. In: STOC 2012, pp. 1219–1234 (2012)
9. Rohloff, K., Cousins, D.B.: A scalable implementation of fully homomorphic encryption built on NTRU. In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) FC 2014. LNCS, vol. 8438, pp. 221–234. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44774-1_18
10. Stehlé, D., Steinfeld, R.: Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_4
11. Security Inovation: NTRU PKCS Tutorial. https://www.securityinnovation.com