 CMC, vol.60, no.1, pp.305-316, 2019

A New NTRU-Type Public-Key Cryptosystem over the Binary Field

Youyu Gu[1], Xiongwei Xie[2] and Chunsheng Gu[3,*]

Abstract: As the development of cloud computing and the convenience of wireless sensor netowrks, smart devices are widely used in daily life, but the security issues of the smart devices have not been well resolved. In this paper, we present a new NTRU-type public-key cryptosystem over the binary field. Specifically, the security of our scheme relies on the computational intractability of an unbalanced sparse polynomial ratio problem (DUSPR). Through theoretical analysis, we prove the correctness of our proposed cryptosystem. Furthermore, we implement our scheme using the NTL library, and conduct a group of experiments to evaluate the capabilities and consuming time of encryption and decryption. Our experiments result demonstrates that the NTRU-type public-key cryptosystem over the binary field is relatively practical and effective.

Keywords: Public key cryptosystem, NTRU, lattice attack, meet in the middle attack.

1 Introduction

In the past few years, cloud computing has attracted a lot of research efforts. At the same time, more and more companies start to move their data and operations to public or private clouds. For example, out of 572 business and technology executives that were surveyed in Berman et al. [Berman, Lynn, Marshall et al. (2012)],51% relied on cloud computing for business model innovation. These demands also become a driving force for the development of cloud security and wireless security, which ranges from very theoretical efforts such as homomorphic encryption to very engineering mechanisms defending against side channel attacks through memory and cache sharing [Xie and Wang (2013b); Xie, Wang and Qin (2015); Pan, Lei, Zhang et al. (2018)].

As the development of cloud computing and the convenience of wireless sensor netowrks, smart devices are widely used in daily life, such as smart phones, but the security issues of the smart devices have not been well resolved [Xie and Wang (2013a); Ren, Shen, Liu et al. (2016)]. One reason is that smart devices do not have enough computing resource, and they are not suitable for the use of traditional cryptographic schemes directly, such as RSA,

[1] School of Mathematics, Hefei University of Technology, Hefei, 230000, China.
[2] Amazon, 410 Terry Ave N Seattle, WA 98109, USA.
[3] College of Computer Engineering, Jiangsu University of Technology, Changzhou, 213000, China.
[*] Corresponding Author: Chunsheng Gu. Email: chunsheng_gu@163.com.

ECC. Therefore, in order to design a lightweight cryptographic scheme suitable for smart devices, this paper constructs a candidate public key encryption scheme based on NTRU over the binary field to partially solve the security problems in smart device applications.

The NTRU public-key cryptosystem was introduced by Hoffstein, Pipher and Silverman in 1996 [Hoffstein, Pipher and Silverman (1998)]. Unlike more classical public-key cryptosystems such as RSA, ECC or ElGamal, its security is based on the hardness of finding the shortest vector problem (SVP) and the closest vector problem (CVP) in a cyclic modular lattice, which are not known to be susceptible to quantum attack. As a consequence, it is considered as one of the most viable quantum-resistant public-key cryptosystems, whereas the classical cryptosystems based on the hardness of integer factorization, or the discrete logarithm over finite fields are no longer secure once the quantum computer becomes a reality [Shor (1997)].

The NTRU system is determined by a set of parameters $(n, q, p, \chi_f, \chi_g, \chi_r, \chi_e)$. First, the parameter n is set to be prime and used to define the polynomial ring $R = \mathbb{Z}[x]/\langle x^n - 1\rangle$. Second, p and q are relatively prime, q is much larger than p, and they are used to define the quotient polynomial rings $R_q = R/qR$ and $R_p = R/pR$ that form the ciphertext space and message space of NTRU, respectively. Finally, $(\chi_f, \chi_g, \chi_r, \chi_e)$ are probability distributions defined in certain subsets of R, and output random polynomials with most coefficients being 0 and the rest in the set $\{1, -1\}$.

Given these parameters of NTRU, Alice samples g from χ_g, and f from χ_f so that f is invertible in R_q and R_p. Alice publishes $h = g/f \in R_q$ as his public key, and keep f as his private key. To encrypt a message polynomial $m \in R_p$, Bob takes Alice's public key h, samples r from χ_r and e from χ_e, computes the ciphertext $c = p(hr + e) + m \in R_q$, and sends it to Alice. To decrypt the ciphertext c, Alice computes $a = fc \mod q$, and outputs the message polynomial $m = af_p^{-1} \mod p$.

Related work. Since NTRU is the most efficient lattice-based public-key cryptosystem, many variants of NTRU were presented by replacing the ring of integers $\mathbb{Z}$ with other rings. Gaborit, Ohler, and Solé introduced CTRU as an analogue to NTRU where the coefficients of polynomials are from $\mathbb{Z}_2^k$ instead of $\mathbb{Z}$. However, Kouzmenko [Kouzmenko (2006)] presented a polynomial time algorithm which breaks CTRU. This is because the CTRU system uses low-degree polynomials instead of "small norm" polynomials. As a consequence, the CTRU system is no longer secure. Several variants of NTRU are proposed by using the Dedekind domains, including GNTRU over the Gaussian integers $\mathbb{Z}[i]$ [Kouzmenko (2006)], ETRU over the Eisenstein integers $\mathbb{Z}[\zeta_3]$ [Nevins, Karimianpour and Miri (2010); Jarvis and Nevins (2015)], NTRUSIGN [Hoffstein, Howgrave-Graham, Pipher et al. (2003)] and NTRU Signature Scheme (NSS) [Hoffstein, Pipher and Silverman (2001)]. The security of these variants is equivalent to the security of NTRU in general. On the other hand, some non-commutative versions of NTRU are also described over the non-commutative ring, including MaTRU over integer matrices [Coglianese and Goi

(2005)], QTRU and BQTRU over quaternion algebras [Malekian, Zakerolhosseini and Mashatan (2009, 2011); Bagheri, Sadeghi and Panario (2017)].

Recently, Aggarwal et al. [Aggarwal, Joux, Prakash et al. (2017)] presented a new public-key cryptosystem via Mersenne numbers (AJPS) that is an integer version of the NTRU system. The security of the AJPS system relies on the conjectured hardness of the Mersenne low hamming ratio assumption. However, Beunardeau et al. [Beunardeau, Connolly, Géraud et al. (2017)] described a practical LLL-based algorithm that recovering the secret key from the public key is much faster than the security estimates in Aggarwal et al. [Aggarwal, Joux, Prakash et al. (2017)] . Furthermore, de Boer et al. [de Boer, Ducas, Jeffery et al. (2017)] further refined the attack analysis of Beunardeau et al. [Beunardeau, Connolly, Géraud et al. (2017)].

Although there are many research results related to variants of NTRU in the past few years, secure NTRU-type public key cryptosystem over the binary field has not attracted a lot of research afforts.

1.1 Our contribution

We propose a new NTRU-type public key cryptosystem over the binary field. As a warmup, Alice chooses two sparse polynomials $f, g \in R_2 = \mathbb{Z}_2[x]/\langle x^n + 1\rangle$, and sets f as the secret key and $h = g/f \in R_2$ as the public key. For encrypting a bit $b \in \{0, 1\}$, Bob chooses sparse polynomials r, e, generates a ciphertext $c = rh + e + bm$, where m is the polynomial of all coefficients 1, and sends c to Alice. For decryption, Alice computes $a = cf$ and outputs $b = 0$ if the number of the non-zero coefficients of a is less than a fixed value (e.g., $n/4$), otherwise $b = 1$. The advantage of this scheme is simple, but it can not be extended to multi-bit schemes easily. In this paper, we propose a multi-bit scheme by using unbalanced sparse polynomials. Namely, Alice chooses two sparse polynomials $f, g \in R_2$ so that the degree of f is at most β, and sets the public key $h = \frac{g}{(x^\theta+1)f+1}$, and the secret key f, where β, θ are positive integers and $\beta + \theta < n$. It is not difficult to construct a multi-bit scheme by using these unbalanced sparse polynomials. Concrete construction is described in Section 2. However, the use of unbalanced polynomials in the construction makes it more vulnerable to man-in-the-middle attacks. Therefore, we will take large enough parameters to resist this attack.

Furthermore, we observe that the distribution of coefficients "1" in the product of two sparse polynomials is almost uniform. If the number of coefficients "1" in the product of two sparse polynomials is k, the probability that each coefficient is "1" is approximately equal to k/n. As a consequence, we assume that this distribution is uniform to improve the efficiency of our scheme.

1.2 Organization

The remainder of the paper is organized as follows. Firstly, we propose a NTRU-type public key cryptosystem and theoretically prove the correctness of it in Section 2. In Section 3, we analyze the security of our scheme and discuss the resistance to popular known attacks. In Section 4, we implement our NTRU-type scheme, and evaluate the capabilities and the consuming time for encryption and decryption. Finally, Section 5 concludes the paper.

2 NTRU-type public key cryptosystem

In this section, we present the details of our new NTRU-type public key cryptosystem over the binary field. Our construction is similar in form to the variant of NTRU [Stehlé and Steinfeld (2011)]. However, our scheme works over the binary field $\mathbb{Z}_2$, and their variant works over $\mathbb{Z}_q$ with $q \gg 2$. It is not trivial to generalize their construction from $\mathbb{Z}_q$ to $\mathbb{Z}_2$.

For simplicity, we concretely define the notations of our scheme as follows:

λ: the security parameter.

$\rho = \lambda/4$: the number of coefficients "1" of random polynomials.

$\alpha = 4\rho$: the length of message vectors.

$\delta = 2\rho$: the extended length of plaintext bits.

$\beta = 4\rho^2$: the degree of secret key polynomials.

$n \geq 20\rho^2 + 1$: the degree of modulo polynomial defined the ring.

$R = \mathbb{Z}_2[x]/\langle x^n + 1\rangle$: the working polynomial ring.

R^*: the set of all invertible polynomials in R.

$P = \mathbb{Z}_2[x]$: the ring of sampling random polynomials.

$P^{<\beta}$: the set of all polynomials of degree less than β in P.

$P_\rho^{<\beta}$: polynomials $r \in P^{<\beta}$ with $\|r\|_1 = \rho$.

$\mathbf{1} = (1, 1, \cdots, 1) \in \mathbb{Z}_2^\delta$: the vector with all entries of 1.

$\mathbf{m} \otimes \mathbf{1}$: the tensor product of two vectors $\mathbf{m}$ and $\mathbf{1}$.

2.1 Construction

Key generation: $(pk, sk) \leftarrow \text{KeyGen}(1^\lambda)$.

(1) Choose a prime $n \geq 20\rho^2 + 1$ so that

$$\text{GCD}(x^n + 1, x^{2^\beta} + 1) = x + 1 \mod 2,$$
$$x^n + 1 = (x + 1)k(x) \mod 2,$$

where $k(x)$ has at most two irreducible factors modulo 2.

(2) Choose at random $s \leftarrow P_\rho^\beta$, and set $f = s(x^{2^\beta} + 1) + 1$ such that $s, f \in R^*$.

(3) Choose at random $g \leftarrow P_\rho^n$ such that $g \in R^*$, and set $h = g/f \in R^*$.

(4) Output the public key $pk = \{\lambda, \rho, n, \beta, h\}$, and the secret key $sk = \{s\}$.

Encryption: $(c) \leftarrow \text{Enc}(pk, \mathbf{m})$.

(1) Given the public key pk, and a plaintext vector $\mathbf{m} \in \{0,1\}^{\alpha}$, compute $\mathbf{d} = \mathbf{m} \otimes \mathbf{1}$, and set $d = \sum_{i=0}^{2\beta-1} d_i x^i$.

(2) Sample $r \leftarrow P_{\rho}^{n}$, $e \leftarrow P_{\rho}^{n}$, and compute

$$c = rh + e + d \mod (x^n + 1) \mod 2$$

(3) Output the ciphertext c.

Decryption: $\mathbf{m} \leftarrow \text{Dec}(sk, c)$.

(1) Given the secret key sk, and a ciphertext c, compute over R

$$w = fc \mod (x^n + 1) \mod 2$$

$$v = w \mod (x^{2\beta} + 1) \mod 2$$

(2) For $i = 0, 1, \cdots, \alpha - 1$

(2.1) Compute $u_i = \sum_{j=0}^{\delta-1} v_{\delta i + j}$.

(2.2) If $u_i \geq \rho$, then $m_i = 1$, otherwise $m_i = 0$.

(3) Output the plaintext vector $\mathbf{m}$.

Remark 2.1 (1) To improve the efficiency of our construction, we can relax the condition of the factor number of $x^n + 1$ over the polynomial ring P. Namely, for a large enough prime n, the factor number of x is only required to be a small constant. In this case, in addition to factor $x + 1$ of $x^n + 1$, other factors need to be able to resist man-in-the-middle attacks.

(2) Our scheme uses unbalanced sparse polynomials to encrypt multi-bit plaintexts. If we construct a single-bit scheme, we only require to use sparse polynomials instead of unbalanced sparse polynomials.

2.2 Correctness

For the correctness of our scheme, it requires to prove that the algorithm Dec correctly recovers the plaintext from a ciphertext with high probability.

We first give the following Chernoff bound.

Lemma 2.2 Let $X_1, \cdots, X_{\delta}$ be independent identically distributed random variables such that $X_i \leftarrow \text{Ber}_{\tau}$, where Ber_{τ} denotes the Bernoulli distribution with the parameter $0 \leq \tau \leq 1$. If $X = \sum_{i=1}^{n} X_i$, then

$$\Pr[X \geq (\tau + \epsilon)\delta] \leq e^{-2\delta\epsilon^2}.$$

Lemma 2.3 Given sk and a ciphertext c, the algorithm Dec correctly decrypts the plaintext vector $\mathbf{m}$.

Proof. According to Dec, we have

$$\begin{aligned} w &= fc \mod (x^n+1) \mod 2 \\ &= f(rh+e+d) \mod (x^n+1) \mod 2 \\ &= rg+fe+fd \mod (x^n+1) \mod 2 \end{aligned}$$

By KeyGen, we have $\deg(f) = \deg(s(x^{2\beta}+1)+1) < 3\beta = 12\rho^2$.

Again through $\deg(d) \leq 2\beta - 1 < 8\rho^2$, we get $\deg(fd) < 20\rho^2 < n$.

So, the polynomial fd remains unchanged in modulo x^n+1. Namely, $fd \mod x^n+1 = fd$.

Without loss of generality, let $e = e_{(1)} + x^{2\beta}e_{(2)}$. Similarly, the polynomial $fe_{(1)}$ also remains unchanged in modulo x^n+1 since $\deg(e_{(1)}) \leq 2\beta - 1$.

So, $w = u + fe_{(1)} + fd \mod 2$, where $u = (rg + fx^{2\beta}e_{(2)}) \mod (x^n+1)$.

As a result, $v = w = (u \mod (x^{2\beta}+1)) + e_{(1)} + d \mod (x^{2\beta}+1) \mod 2$.

In the following analysis, we assume that the coefficients "1" of noise polynomials are uniformly distributed. Concretely speaking, the probability that any coefficient of a noise polynomial y with length k is "1" is equal to $\frac{\|y\|_1}{k}$.

Hence, we get $\|e_{(1)}\|_1 = \frac{2\beta}{n}\rho \approx \frac{2}{5}\rho$, and $\|e_{(2)}\|_1 \approx \frac{3}{5}\rho$. It follows that for u,

$$\begin{aligned} &\|u \mod (x^{2\beta}+1) + e_{(1)}\|_1 \\ &\leq \|r\|_1 \times \|g\|_1 + \|f\|_1 \times \|x^{2\beta}\|_1 \times \|e_{(2)}\|_1 + \|e_{(1)}\|_1 \\ &\leq \rho \times \rho + 2\rho \times 1 \times \frac{3}{5}\rho + \frac{2}{5}\rho \\ &\approx \frac{11}{5}\rho^2 + \frac{2}{5}\rho \end{aligned}$$

Since $z = u \mod (x^{2\beta}+1) + e_{(1)}$ is a noise polynomial in v, the probability that any coefficient of z is "1" is equal to $(\frac{11}{5}\rho^2 + \frac{2}{5}\rho)/2\beta \approx \frac{11}{40}$.

Therefore, the expected number of "1" in a polynomial of length 2ρ is $\frac{22}{40}\rho$.

Let $z_{(i)} = \sum_{j=0}^{\delta-1} z_{i\delta+j}x^{i\delta+j}$ for $i = 0, \cdots, \alpha-1$. So, $\text{Exp}(\sum_{j=0}^{\delta-1} z_{i\delta+j}) = \frac{22}{40}\rho$.

By Lemma 2.2, we have

$$\Pr\left[\sum_{j=0}^{\delta-1} z_{i\delta+j} \geq \rho\right] \leq e^{-2\delta(\frac{9}{40})^2} \approx e^{-\frac{\rho}{5}}.$$

So, the probability that m_i can be correctly recovered is about $1 - e^{-\frac{\rho}{5}}$.

3 Security

In this section, we will define decisional unbalanced sparse polynomial ratio problem (DUSPR) and the DUSPR assumption, and analyze some known attacks.

The security of the NTRU variant [Stehlé and Steinfeld (2011)] is reduced to worst-case problems over ideal lattices, but the security of NTRU is still based on the computational hardness assumption generated by NTRU. Similarly, the security of our NTRU-type scheme is also based on the new DUSPR hardness assumption.

3.1 Hardness assumption

Definition 3.1 Decisional unbalanced sparse polynomial ratio problem (DUSPR). Given the above parameters $\{\lambda, \rho, \beta, n\}$, a distinguisher *D* is said to $(\lambda, \rho, \beta, n, t, \epsilon)$-solve the $\text{DUSPR}_{\lambda,\rho,\beta,n}$ problem if

$$|\Pr[D(h) = 1] - \Pr[D(a) = 1]| \geq \epsilon$$

where $h = g/f \in R^*$, $f = s(x^{2\beta} + 1) + 1$, $g \leftarrow P_\rho^n$, $s \leftarrow P_\rho^\beta$ with $s, g \in R^*$, and $a \leftarrow R^*$, and *D* runs in time at most t.

Our public key cryptosystem is based on the following assumption.

Definition 3.2 DUSPR assumption. For any probabilistic distinguisher *D* that $(\lambda, \rho, \beta, n, t, \epsilon)$-solves the $\text{DUSPR}_{\lambda,\rho,\beta,n}$ problem for all large enough λ, where $\rho = \lambda/4$, $\beta = 4\rho^2$, $n = 20\rho^2 + 1$, and t is polynomial in λ, the advantage ϵ that *D* holds is negligible as a function of λ.

Lemma 3.3 Under the DUSPR assumption, the public key encryption scheme (Enc, Dec) described in Section 2 is secure against chosen plaintext attack.

Proof. Given two polynomials $d_0, d_1 \in P^{<2\beta}$ corresponding to plaintext vectors $\mathbf{m}_0, \mathbf{m}_1$, for $i = 0, 1$ let $c_i = r_i h + e_i + d_i \mod (x^n + 1) \mod 2$, be the ciphertexts of d_i, where $r_i \leftarrow P_\rho^n$, $e_i \leftarrow P_\rho^n$.

Note that for simplicity we assume that $c_1, c_2 \in R^*$. The reason is that if $GCD(c_i, x^n+1) \neq 1$, we can flip the 0-th coefficient of c_i.

By contradiction, assume that there exists a polynomial time algorithm *B*, so that

$$|\Pr[B(h, c_1) = 1] - \Pr[B(h, c_2) = 1]| \geq n^{-O(1)}. \tag{1}$$

Let $b \leftarrow R^*$. According to the DUSPR assumption, for any polynomial time algorithm *A* we have

$$|\Pr[A(h, c_i) = 1] - \Pr[A(h, b) = 1]| \leq \text{negl}_i(\lambda), \quad i = 0, 1. \tag{2}$$

Table 1: The concrete parameter settings of our NTRU-type scheme

Security level (bits)	λ	ρ	β	n (prime)	x^n+1 (the number of factors)	the size of pk (bits)	the size of sk (bits)
80	120	30	3600	18013	2	18013	3600
112	144	36	5184	25931	2	25931	5184
128	160	40	6400	32003	2	32001	6400
160	200	50	10000	50021	2	50021	10000

Since B is a polynomial time algorithm, we get

$$\begin{aligned}
&|\Pr[B(h,c_1)=1]-\Pr[B(h,c_2)=1]| \\
&\le |\Pr[B(h,c_1)=1]-\Pr[A(h,b)+ \\
&\qquad \Pr[A(h,b)-\Pr[B(h,c_2)=1]| \\
&\le |\Pr[B(h,c_1)=1]-\Pr[A(h,b)|+ \\
&\qquad |\Pr[A(h,b)-\Pr[B(h,c_2)=1]| \\
&\le \mathrm{negl}_0(\lambda)+\mathrm{negl}_1(\lambda) \\
&= \mathrm{negl}(\lambda),
\end{aligned} \tag{3}$$

where $\mathrm{negl}_0(\lambda), \mathrm{negl}_1(\lambda)$, and $\mathrm{negl}(\lambda)$ are negligible functions in λ.

This generates a contradiction for the expression (1) and (3).

3.2 Known attacks

In the following subsection, we theoretically analyze how our proposed scheme prevents known attacks, including NTRU-type lattice attack, meet in the middle attack, and attack of factoring modulo x^n+1. Our analysis result demonstrates that our scheme can resist all these known attacks.

NTRU-type lattice attack. For the NTRU system, given the public key $h=g/f$ over the ring $\mathbb{Z}_q[x]/(x^n-1)$, it is easy to construct the NTRU public lattice [Coppersmith and Shamir (1997); Hoffstein, Pipher and Silverman (1998)] as follows:

$$L_1=\begin{pmatrix} qI & 0 \\ H & I \end{pmatrix} \tag{4}$$

where H is a circulant matrix generated from h.

According to the parameter settings of NTRU, the vector (g,f) in L_1 has size $(d_f+d_g)^{1/2}$, where d_f, d_g are the number of the non-zero coefficients of f, g, respectively. Since $\det(L_1)=q^n$, the Gaussian heuristic suggests that (g,f) is in general the shortest vector in L_1. However, the current lattice reduction algorithm that find (g,f) requires exponential in the security parameter n.

Similarly, for our NTRU-type system, given the public key $h=g/f$ over $\mathbb{Z}_2[x]/(x^n+1)$, we can also construct a lattice from h. Owing to using the unbalanced private key f, we only need to use the 2β rows of the circulant matrix H generated by h. The reaseon is that

Table 2: The performance of our NTRU-type scheme

Security level (bits)	Length per plaintext (bits)	Length per ciphertext (bits)	Expansion rate	Time per encryption (ms)	Time per decryption (ms)	Testing frequency	Successful rate (%)
80	120	18013	150	3.382	3.198	2000	100
112	144	25931	180	5.744	5.547	2000	100
128	160	32003	200	7.693	8.209	2000	100
160	200	50021	250	11.613	15.735	2000	100

$fh = (s+1)h + s(x^{2\beta}h) = f_1h + f_2h$. As a reasult, we write a matrix form as follows:

$$L_2 = \begin{pmatrix} 2I_{n\times n} & 0 & 0 \\ H[\,0:\ \beta-1] & I_{\beta\times\beta} & 0 \\ H[2\beta:3\beta-1] & 0 & I_{\beta\times\beta} \end{pmatrix} \tag{5}$$

where H is a circulant matrix generated from h, $H[i:j]$ represents the sub-matrix of the i-th row to the j-th row of H.

By our parameter settings, the vector (g, f_1, f_2) in L_2 has size $(3\rho+1)^{1/2}$ or $(3\rho-1)^{1/2}$. Since $\det(L_2) = 2^n$, the Gaussian heuristic suggests that (g, f_1, f_2) is usually the shortest vector in L_2. When n is large enough, the lattice reduction algorithm that computes (g, f_1, f_2) requires time complexity at about $2^{O(n)}$.

Meet in the middle attack. The idea of the meet-in-the-middle attack on NTRU [Howgrave-Graham (2007)] is that if $f_1 + f_2 = f$, then $(f_1 + f_2)h = g \mod q$. In other words, the entries of $y_1 = f_1h$ and $y_2 = -f_2h$ differ only by 0 or 1 mod q. According to this property, the meet-in-the-middle attack performs sampling f_1 with $d_f/2$ "1" coefficients, and storing them in boxes dependent on the y_1. If two binary elements f_1, f_2 are satisfied $f = f_1+f_2$, then we hope that this can be detected by a collision in a box. For any collisions, we can retrieve the f_1, f_2 from the stored box, and determine whether $(f_1 + f_2)h$ is binary or not. Once we find a very small vector in the NTRU public lattice, it is very likely one of the rotation of (g, f). According to the analysis, the classical (resp. quantum) meet-in-the-middle attack requires the time complexity and space complexity at least $\binom{n}{d_f}^{1/2} \approx 2^{\frac{1}{4}d_f \log_2 n}$ [Howgrave-Graham (2007)] (resp. $\binom{n}{d_f}^{1/3} \approx 2^{\frac{1}{6}d_f \log_2 n}$ [de Boer, Ducas, Jeffery and Wolf (2017)]).

Similarly, for our NTRU-type system, it is not difficult to verify that the classical (resp. quantum) meet-in-the-middle attack requires the time complexity and space complexity at least $2^{\frac{1}{4}\rho \log_2 \beta}$ (resp. $2^{\frac{1}{6}\rho \log_2 \beta}$).

Attack of factoring $x^n + 1$ modulo 2. According to our parameter settings, the $x^n + 1$ has at most three factors modulo 2. In other words, $x^n + 1 = (x + 1)k(x) \mod 2$ such that $k(x)$ is irreducible or $k(x) = k_1(x)k_2(x)$ modulo 2. As far as we know, when n is large enough, no effective algorithm can use the factors of $x^n + 1$ to attack our system.

4 Implementation

To evaluate the encryption and decryption capabilities of the proposed approach, and access its consuming time on different security level, we conduct one group of experiments. The experiment environment setup is as follows. We implemented our NTRU-type public key cryptosystem over the NTL library. All programs were run on the physical machine, which has a 3.20 GHz Intel Core i5-3470 processor, and 8 GB of RAM.

Tab. 1 is our concrete parameter settings. We define different security level with different parameter values. Tab. 2 is the performance result of our NTRU-type scheme. Note that the estimate of the security level mainly relies upon the time complexity of the classical meet-in-the-middle attack on our NTRU-type scheme.

When security level is 80 (λ=120, ρ=30, β=3600, n=18013), we have 100% successful rate for testing frequency=2000, and average excryption/decryption time is about 3 ms with 150 expansion rate. When security level is 160 (λ=200, ρ=50, β=10000, n=50021), we have 100% successful rate for testing frequency=2000, and average excryption/decryption time is about 15ms with 250 expansion rate. From our experiments result, we can notice that if we directly encrypt plaintexts by applying our public key scheme, its performance is relatively weak, especially for the ciphertext expansion rate. However, if we use our public key scheme for key encapsulation mechanism, our scheme will be relatively practical and effective.

It should be noted that we did not optimize our implementation and only illustrate the relative practicality of our construction.

5 Conclusions

In this paper, we propose a new NTRU-type public-key cryptosystem over the binary field, whose security relies on the computational intractability on the DUSPR problem.We present the details of our new NTRU-type plublic key cryptosystem with the theoretical analysis, and prove our decryption algorithm correctly recovers the plaintext from a ciphertext with high probability. We also theoretically analyze and prove that our proposed cryptosystem could avoid known attacks, including NTRU-type lattice attack, meet in the middle attack, and and attack of factoring modulo $x^n + 1$. Furthermore, we implement our scheme using the NTL library, and conduct a group of experiments in different security level. Our result demonstrates that our proposed NTRU-type public-key cryptosystem over $\mathbb{Z}_2$ is relatively practical.

Immediate extensions to our approach consist of the following aspects. First, we plan to experiment our approach with cell phone so that we can evaluate its improvements comparing to traditional cryptosystem. Second, we plan to study the feasibility and security of digital signature and authentication through conducting NTRU-type public key cryptosystem over the binary field. Finally, we plan to reduce the security of our scheme to the learning parity with noise (LPN) [Pietrzak (2012)] problem theoretically, so that we

could get rid of the assumption of DUSPR.

Acknowledgement: This work was supported by the National Natural Science Foundation of China (Nos. 61672270, 61702236 and 61602216) and Changzhou Sci&Tech Program (Grant No. CJ20179027). We thank anonymous reviewers for their helpful suggestions which greatly improved the presentation of this paper.

Conflict of Interest

We declare that the funding in the Acknowledgment section did not lead to any conflict of interests regarding the publication of this manuscript. Also, there is no any other conflict of interests in the manuscript.

References

Bagheri, K.; Sadeghi, M. R.; Panario, D. (2017): A non-commutative cryptosystem based on quaternion algebras. *Designs, Codes and Cryptography*, pp. 1-33.

Coglianese, M.; Goi, B. (2005): Matru: a new ntru-based cryptosystem. *6th International Conference on Cryptology in India*, pp. 232-243.

Coppersmith, D.; Shamir, A. (1997): Lattice attacks on ntru. *EUROCRYPT'97 Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques*, pp. 52-61.

de Boer, K.; Ducas, L.; Jeffery, S.; Wolf, R. (2017): Attacks on the ajps mersenne-based cryptosystem. *International Conference on Post-Quantum Cryptography*, pp. 101-120.

Hoffstein, J.; Pipher, J.; Silverman, J. H. (1998): Ntru: a ring-based public key cryptosystem. *International Algorithmic Number Theory Symposium ANTS 1998: Algorithmic Number Theory*, pp. 267-288.

Hoffstein, J.; Pipher, J.; Silverman, J. H. (2001): Nss: an ntru lattice-based signature scheme. *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 211-228.

Howgrave-Graham, N. (2007): A hybrid lattice-reduction and meet-in-the-middle attack against ntru. *Annual International Cryptology Conference CRYPTO 2007: Advances in Cryptology*, pp. 150-169.

Jarvis, K.; Nevins, M. (2015): Etru: Ntru over the eisenstein integers. *Designs Codes and Cryptography*, vol. 74, no. 1, pp. 219-242.

Kouzmenko, R. (2006): Generalizations of the ntru cryptosystem. *Diploma Project, école Polytechnique Federale de Lausanne*, pp. 1-20.

Malekian, E.; Zakerolhosseini, A.; Mashatan, A. (2009): Qtru: a lattice attack resistant version of ntru pkcs based on quaternion algebra. *Cryptology ePrint Archive, Report 2009/386*, pp. 1-25.

Malekian, E.; Zakerolhosseini, A.; Mashatan, A. (2011): Qtru: quaternionic version of the ntru public-key cryptosystems. *The ISC International Journal of Information Security*, vol. 3, no. 1, pp. 29-42.

Nevins, M.; Karimianpour, C.; Miri, A. (2010): Ntru over rings beyond $\mathbb{Z}$. *Designs Codes and Cryptography*, vol. 56, no. 1, pp. 65-78.

Pietrzak, K. (2012): Cryptography from learning parity with noise. *International Conference on Current Trends in Theory and Practice of Computer Science*, pp. 99-114.

Shor, P. W. (1997): Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484-1509.

Stehlé, D.; Steinfeld, R. (2011): Making ntru as secure as worst-case problems over ideal lattices. *EUROCRYPT'11 Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques*, pp. 27-47.

Xie, X.; Wang, W. (2013): Detecting primary user emulation attacks in cognitive radio networks via physical layer network coding. *Procedia Computer Science*, vol. 21, pp. 430-435.

Xie, X.; Wang, W. (2013): Rootkit detection on virtual machines through deep information extraction at hypervisor-level. *IEEE Conference on Communications and Network Security*, pp. 498-503.

Xie, X.; Wang, W.; Qin, T. (2015): Detection of service level agreement (sla) violation in memory management in virtual machines. *24th International Conference on Computer Communication and Networks*, pp. 1-8.