Computers and Electrical Engineering 63 (2017) 220–231

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Random key rotation: Side-channel countermeasure of NTRU cryptosystem for resource-limited devices☆

An Wang[a,b,*], Ce Wang[a], Xuexin Zheng[c], Weina Tian[d], Rixin Xu[a], Guoshuang Zhang[e]

[a] School of Computer Science, Beijing Institute of Technology, Beijing 100081, China
[b] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, 100878, China
[c] China Academy of Electronics and Information Technology, Beijing 100041, China
[d] College of Bioengineering, Beijing Polytechnic, Beijing 100176, China
[e] Science and Technology on Information Assurance Laboratory, Beijing 100072, China

ARTICLE INFO

Article history:
Received 29 November 2016
Revised 27 April 2017
Accepted 6 May 2017
Available online 10 May 2017

Keywords:
NTRU
Random key rotation
Power analysis attacks
Side-channel attacks
Resource-limited devices

ABSTRACT

The NTRU algorithm, as IEEE P1363.1 standard, has been widely applied in resource-limited cryptosystems such as wearable embedded systems, smart cards, and so on. In 2013, Zheng et al. presented collision attack on three NTRU countermeasures from Lee et al., which are the only three countermeasures of NTRU against side-channel analysis so far. In this paper, we give a new countermeasure named Random Key Rotation (RKR) against the existing statistical side-channel analysis. According to analysis and experiments on STC89C52 microprocessor, little information of the key is leaked by collision attack, second-order correlation power analysis, etc. Furthermore, implementation schemes of RKR countermeasure in software and hardware are suggested. Under typical parameters, barrel shifter implementation of RKR only costs 8-bit extra register and one extra clock cycle (as well as 16 extra multiplexers).

1. Introduction

Nowadays, low-power communication devices such as wearable devices, RFID cards, etc. have been widely applied in people's daily life, in which it is quite important for the security of sensitive data. So, schemes of encrypted communication and authentication should be designed, and some cryptographic algorithms [1] are employed for their implementations. However, in some specific applications, the designers should consider serious restriction on the amount of gate area, computational complexity, and memory storage that are available for tiny sensors [2]. The NTRU cryptosystem [3] is suggested for public-key encryption and authentication in wireless environment due to its high efficiency [4,5].

The NTRU cryptosystem was proposed in rump session of CRYPTO 1996 by Hoffstein et al. NTRU decryption and signature take much less time and area than ECC (Elliptic Curve Cryptosystems), RSA, and most other public key cryptosystems with the same security level [6,7]. NTRU implementations include mainly convolution product over the polynomial ring. However, the difficulty of cracking it is equivalent to a hard problem over lattice [8]. Because of the fast performance, theoretical resistance against quantum computation, and widely application, it was regarded as the standard of IEEE P1363.1 in 2008

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Debiao He.

* Corresponding author at: School of Computer Science, Beijing Institute of Technology, Beijing 100081, China.
E-mail addresses: wanganl@sina.com (A. Wang), zhxuexin@163.com (X. Zheng).

http://dx.doi.org/10.1016/j.compeleceng.2017.05.007
0045-7906/© 2017 Published by Elsevier Ltd.

[9]. In the past ten years, the security of NTRU is widely studied by cryptography researchers [10–12]. Thus, most small shortcomings of NTRU have been eliminated. Now, it is thought that NTRU cryptosystem can resist all existing theoretical attacks due to the exponential complexity of the breaking algorithm.

However, NTRU may encounter some attacks because of implementation drawbacks. During the NTRU execution, the power analysis attacks [13] may recover the secret keys from the power consumed by devices [14–16]. For example, Atici et al. gave practical power analysis attacks on NTRU hardware [17,18]. One year later, Lee et al. discussed simple power analysis (SPA) and correlation power analysis (CPA) on NTRU software [19], and accordingly, they gave three SPA and CPA countermeasures. So far, their countermeasures are the only NTRU ones against power analysis attacks, but they are not adequate for hiding the secret. In 2013, Zheng et al. proposed a collision attack, which broke their countermeasures efficiently [20].

In this paper, we present a new countermeasure called Random Key Rotation (RKR) which has the following features.

- ➢ The information of secret key is hidden directly, instead of only ciphertext or sequence of arithmetic. So, the new proposal can resist first-order power analysis based on statistical analysis and the existing second-order CPA.
- ➢ Its extra implementations only include rotation operations. We implement random key rotation by two schemes, barrel shifter and one-bit rotation loop. Our countermeasure occupies much less storage and time than the existing ones, especially for hardware.
- ➢ The new countermeasure is unencumbered because it can join the unprotected NTRU naturally, and no operations are needed in the end.

The remainder of this paper is organized as follows. In Section 2, we briefly introduce NTRU cryptosystems, existing power analysis attack, and countermeasures. Section 3 introduces our new countermeasures against the existing attacks. In Section 4, some security analyses are given. Section 5 shows the implementations and efficiency comparisons. Finally, Section 6 concludes this paper.

2. Preliminaries

2.1. Security service and threat of resource-limited devices

Most resource-limited devices such as wearable embedded systems transfer private information between sensors and servers over the wireless channel and Internet, which should be private and confidential [21] because the attacker may tamper the real data, submit shoddy data, or violate user's privacy. Accordingly, some authentication and encryption protocols should be designed for devices security. Due to the restricted resources, some light weight cryptographic schemes such as NTRU are employed.

Taking wearable embedded environment for example, power analysis can be mounted in the way described in Fig. 1. First, the adversary gets the wearable device, and acquires the power consumption with a differential probe and oscilloscope. Then, the NTRU private key is recovered. So, the encrypted data monitored from the normal information channel between server and wearable device can be decrypted by this NTRU private key. This procedure is shown from (1) to (7).

2.2. NTRU cryptosystem and convolution product

NTRU is designed on polynomial ring $R = \mathbb{Z}[x]/(x^N - 1)$, on which an element $f \in R$ is written as a polynomial $f = \sum_{i=0}^{N-1} f_i x^i$ or $[f_0, f_1, \dots, f_{N-1}]$. In the ring R, the multiplication is denoted by convolution product, $f{*}g=h$ with

$$\begin{aligned} h_k &= \sum_{i=0}^{k} f_i g_{k-i} + \sum_{i=k+1}^{N-1} f_i g_{N+k-i} \\ &= \sum_{i+j \equiv k \bmod N} f_i g_j \end{aligned}$$

Here polynomial f modulo q means reducing the coefficients modulo q. And $f^{-1} mod\, q$ means $ff^{-1} \equiv 1\ mod\, q$ in R.

The integer N decides the ring $R = \mathbb{Z}[x]/(x^N - 1)$. p is a small modulus, and q is a large one. $f = 1 + pF$ is private key which is invertible modulo q. Here F is a binary polynomial including d_F nonzero coefficients. $h = f^{-1}g\ mod\, q$ is public key, while g is a small secret polynomial.

In encryption, m is the plaintext polynomial. A small polynomial r is randomly chosen as the blinding polynomial. Then the ciphertext can be computed by $c = p{*}r{*}h + m\ mod\, q$. Contrarily, the decryption can be computed by

$$a = f * c \ mod\ q = c + p * F * c \ mod\ q.$$

Then $m = a\ mod\, p$ is computed as the plaintext.

In decryption, the convolution product $F{*}c\ mod\, q$ is usually adopted for key recovery by power attacks. Here as a binary polynomial, F has fixed nonzero coefficients, while the coefficients of c are randomly distributed. Algorithm 1 describes the

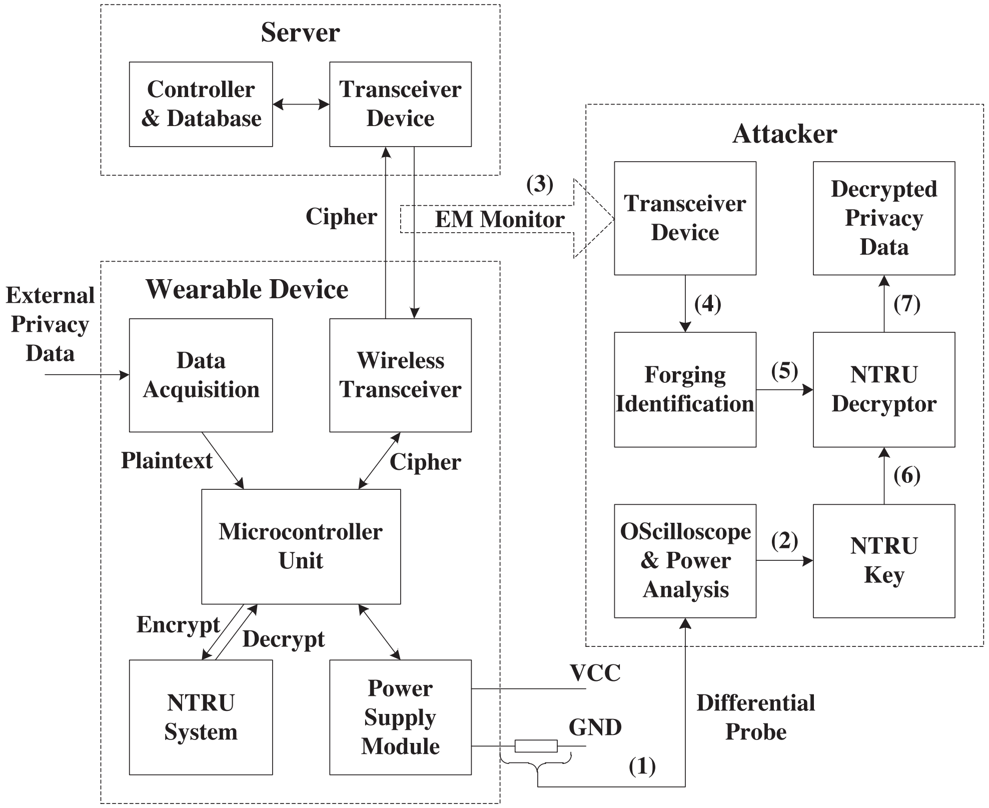

Fig. 1. Power attack process on NTRU-based wearable device.

Algorithm 1
Convolution product.

Input: b (an array of size d representing the locations of nonzero coefficients of the binary polynomial F), c (a general polynomial $c=[c_0,c_1,\ldots,c_{N-1}]$), d, q, N.
Output: $t=F^*c \bmod q$.

```
1:  for $0 \le j < 2N-1$ do
2:      $t_j \leftarrow 0$
3:  end for
4:  for $0 \le j < d$ do
5:      for $0 \le k < N$ do
6:      $t_{k+b[j]} \leftarrow t_{k+b[j]}+c_k$
7:      end for
8:  end for
9:  for $0 \le j < N-1$ do
10:     $t_j \leftarrow (t_j+t_{j+N}) \bmod q$
11: end for
12: return $\{t_j | j=0, 1, 2, \ldots, N-1\}$
```

algorithm of $t=F^*c \bmod q$. We regard F as array b which means the d locations of coefficient 1. For example, we simply represent

$$F(x) = x + x^4 + x^5 + x^7 = [0, 1, 0, 0, 1, 1, 0, 1]$$

as $b=[1, 4, 5, 7]$. Fig. 2 describes the flow of convolution product and its countermeasure of random initialization of t [19], which will be introduced in Section 2.4.

2.3. Simple and correlation power analysis on NTRU

Lee et al. proposed SPA on NTRU in 2010 [19]. Assuming that the power consumptions of the operation $x+y$ and $x+0$ are different (x and y are nonzero integers), the b array will be leaked from step 6 of Algorithm 1. First, the adversary chooses a polynomial c whose coefficients are nonzero. In line $j=1$ in Fig. 2, the first $N-(b[1]-b[0])$ operations are additions with the form of $c_{k+(b[1]-b[0])}+c_k$, and the two operands are nonzero. However, the remainder $b[1]-b[0]$ operations are the form

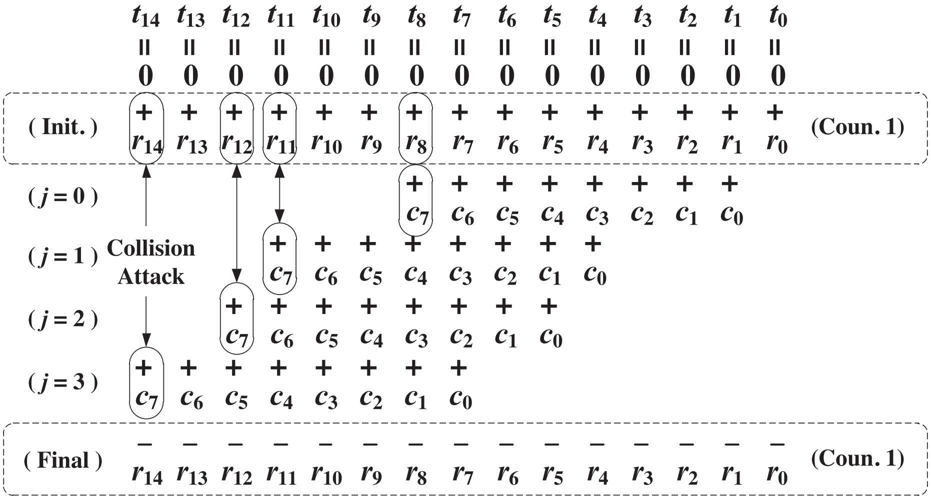
t_{14} t_{13} t_{12} t_{11} t_{10} t_9 t_8 t_7 t_6 t_5 t_4 t_3 t_2 t_1 t_0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
(Init.)
$+r_{14}$ $+r_{13}$ $+r_{12}$ $+r_{11}$ $+r_{10}$ $+r_9$ $+r_8$ $+r_7$ $+r_6$ $+r_5$ $+r_4$ $+r_3$ $+r_2$ $+r_1$ $+r_0$
(Coun. 1)
($j=0$) $+c_7$ $+c_6$ $+c_5$ $+c_4$ $+c_3$ $+c_2$ $+c_1$ $+c_0$
($j=1$) $+c_7$ $+c_6$ $+c_5$ $+c_4$ $+c_3$ $+c_2$ $+c_1$ $+c_0$
Collision Attack
($j=2$) $+c_7$ $+c_6$ $+c_5$ $+c_4$ $+c_3$ $+c_2$ $+c_1$ $+c_0$
($j=3$) $+c_7$ $+c_6$ $+c_5$ $+c_4$ $+c_3$ $+c_2$ $+c_1$ $+c_0$
(Final)
$-r_{14}$ $-r_{13}$ $-r_{12}$ $-r_{11}$ $-r_{10}$ $-r_9$ $-r_8$ $-r_7$ $-r_6$ $-r_5$ $-r_4$ $-r_3$ $-r_2$ $-r_1$ $-r_0$
(Coun. 1)

Fig. 2. Flow chart of Algorithm 1 for $N=8$, $d=4$, $b=[1, 4, 5, 7]$ and collision attack from Zheng et al.

of $0+c_k$. So, according to the differentiability between zero and nonzero, the adversary can recover $b[1]-b[0]$ by deciding the position in which the power consumption is changed. In the same way, the adversary can recover all the $b[i]-b[i-1]$. Subsequently, an exhaustive search can be applied for finding out $b[0]$, and the other $b[i]$.

In 2010, Lee et al. also showed CPA on NTRU. For line $j=1$ of Fig. 2, the adversary suppose that c_0 is added to the same register t_i as c_w does when $j=0$. According to the Hamming distance model, its power consumption will leak the Hamming distance from c_w to c_w+c_0. So, if the offset w is correctly guessed, the correlation between the power consumption and $HD(c_w, c_w+c_0)$ will be much bigger than the correlation corresponding the wrong w. So, the adversary will get the correct w, which is exactly the key $b[1]-b[0]$. Similarly, the keys $b[2]-b[1]$, ..., $b[d-1]-b[d-2]$ can also be obtained.

2.4. Existing countermeasures from Lee et al

The first countermeasure from Lee et al. [19] for preventing their SPA is named random initialization. Specifically, at the beginning of Algorithm 1, some random mask $r_i(i=0, \ldots, 2N-2)$ are chosen and added to all the register t_i. When the original Algorithm 1 is done, the r_i should be removed from the corresponding t_i. So, no operations in the form of $x+0$ are executed, and SPA can be avoided.

Blinding c with random values is the second NTRU countermeasure, which can resist CPA from Lee et al. Replacing the original input values c_i $(i=0, \ldots, N-1)$ by c_i+r (r is a random integer) can mask the intermediate values. In order to be decrypted correctly, step 10 of Algorithm 1 should be replaced by

$$t_j \leftarrow \left(t_j+t_{j+N}-dr\right) \bmod q.$$

The third NTRU countermeasure from Lee et al. is called randomization of b, in which shuffled order of $b[i]$ hinders the statistical analysis on the secret key $b[i]-b[i-1]$.

2.5. Collision attack on protected NTRU

For a protected NTRU, Zheng et al. proposed that recovering $b[0]$ is equivalent to finding which c_7 is added to r_i in line $j=0$ in Fig. 2 [20]. On the one hand, in the initialization step, the random mask r_i is saved to the register t_i. Let $T_{1,i}$ $(i=7, \ldots, 11)$ denote the power consumption of the save operation, which is related to r_i. On the other hand, in line $j=0$, before the last addition $(+c_7)$, r_8 should be moved to the addend register. Let T_2 denote its corresponding power consumption, which is related to r_8. The adversary can execute a collision detection between T_2 and each $T_{1,i}$ $(i=7, \ldots, 11)$ based on the Hamming weight model. So, the collision between $T_{1,8}$ and T_2 can be decided, and then $b[0]$ can be obtained.

3. New countermeasure - random key rotation

Although the third countermeasure disturbs the sequence of operations related to the secret key F, the secret itself is still fixed. Its invariability makes the statistical analysis feasible. In this section, we propose a new countermeasure named Randomly Rotating Key (RKR) against the statistical analysis. The idea is to randomize the secret key F during every decryption.

Algorithm 2. Convolution product with random key rotation.

```
Input: $b$ (an array of size $d$ representing the locations of nonzero coefficients of the binary polynomial $F$), $c$ (a general polynomial $c=[c_0,c_1,\ldots,c_{N-1}]$), $d$, $q$, $N$.
Output: $t=F^*c \bmod q$.
$i \leftarrow$ Random([0, $N-1$])
$b^{(i)} \leftarrow F^*x^i$
$c^{(i)} \leftarrow c^*x^{N-i}$
$t \leftarrow$ ConvolutionProduct($b^{(i)}$,$c^{(i)}$,$d$, $q$, $N$)
return $t$
```

Table 1
Rotated key of a toy example.

$\boldsymbol{i}$	0	1	2	3	4	5	6	7
$\boldsymbol{b^{(i)}}$	1,4,5,7	0,2,5,6	1,3,6,7	0,2,4,7	0,1,3,5	1,2,4,6	2,3,5,7	0,3,4,6

3.1. Basic idea: key randomization

Before the start of every decryption, we employ random cyclic shift for the randomization. Specifically, we randomly choose an integer $i \in \{0, \ldots, N-1\}$, and use F^*x^i as the secret key. At the same time, we replace the ciphertext c with c^*x^{N-i}. We still represent F as $b=[b[0], \ldots, b[d-1]]$ and c as $[c_0,\ldots, c_{N-1}]$, and denote the rotated key F^*x^i as an array

$$\begin{aligned} b^{(i)} &= \left[b^{(i)}[0], \ \ldots, \ \ b^{(i)}[d-1]\right] \\ &= Order[(b[0]+i) \bmod N, \ldots, (b[d-1]+i) \bmod N] \end{aligned}$$

where $Order[i, j, k, \ldots]$ means an array with sorted numbers from small to large. Meanwhile, we denote the rotated ciphertext

$$c^{(i)} = \left[c^{(i)}_{N-1}, \ldots, c^{(i)}_0\right] = [c_{i-1} \ , c_{i-2}, \ldots, c_0, c_{N-1}, c_{N-2} \ , \ldots, c_i].$$

3.2. Protected decryption process

The decryption process is described in Algorithm 2. Here the **Random**() function means randomly choose an integer, and the **ConvolutionProduct**() function means Algorithm 1.

Because the secret key is randomly shifted in each decryption, the existing attacks based on statistical analysis of many traces using the same key F do not work anymore.

3.3. Correctness proof and example

We can prove the correctness of Algorithm 2 simply. Since this convolution product is computed in the ring $R = \mathbb{Z}[x]/(x^N-1)$, we have

$$(F^*x^i)^*(c^*x^{N-i})=F^*c^*x^N=F^*c.$$

So the decryption with random key rotation is correct.

The following toy example shows how our countermeasure hides the secret key F. We take the parameter $N=8$, $d=4$, $b=[1, 4, 5, 7]$. As we can see from Table 1, $\{b^{(i)}[k]\}$ go through all the possible values from 0 to 7 (i.e. $N-1$). We also give an illustration of RKR in Fig. 3.

3.4. Combine RKR with other countermeasures

For a high level of security, we strongly recommend that the combination of random key rotation and random initialization of t from Lee et al. is used as a practical countermeasure. We argue that this new combination countermeasure is secure against the first-order power analysis. That is because, random initialization of t protects NTRU from some potential SPA, and randomly rotating key resists all the attacks using statistical analysis method.

Similarly, the combination of RKR and randomization of b from Lee et al. is also an alternative in the case of low resource, but here d should be large enough for resisting SPA.

4. Security evaluation of RKR

We still employ the toy example above for intuitively showing three kinds of attacks on RKR (with random initialization of t) in Fig. 4. Just to be clear, we replace c_i with its subscript i, and omit the random initialization of t in this figure.

It is clear that the attacker cannot recover the value of i directly because it will be changed in each encryption. From the point of view of information theory, the amount of information of RKR is log$N$. For example, If $N=251$ and i is randomly

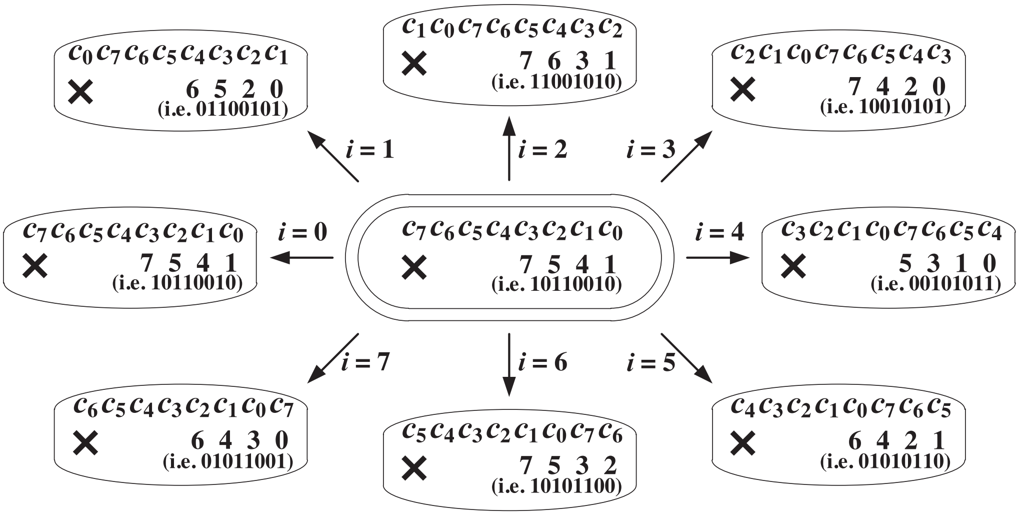

Fig. 3. Illustration of RKR for $N=8$, $d=4$, $b=[1, 4, 5, 7]$.

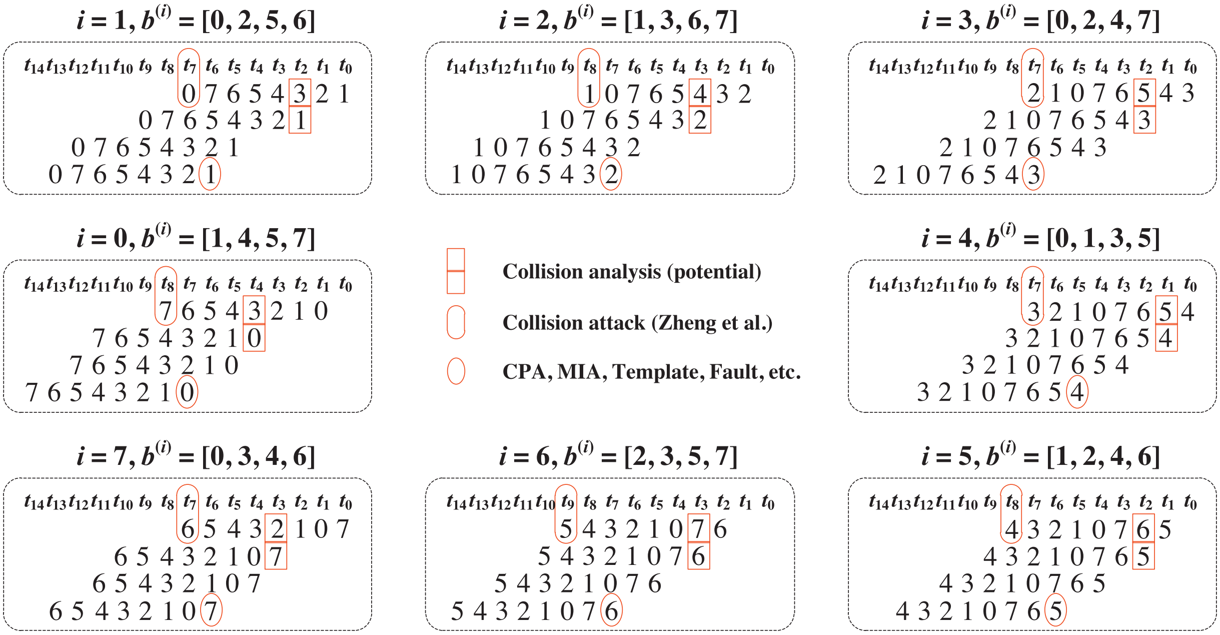

Fig. 4. Three kinds of attacks on the toy RKR.

chosen from [0250], the amount of information is slightly less than 8. Therefore, RKR is a lightweight and low entropy mask scheme.

In the following, some potential advanced attacks are discussed.

4.1. Evaluation by potential collision analysis

We first focus on a potential collision analysis which is described by double-box in Fig. 4. Because in RKR, each c_0 could not be located exactly, attacker may compute the correlation coefficients between each of 251 trace sections corresponding to the 251 c_k in line $j=0$ and the trace section corresponding to the least significant $c_0^{(i)}$ (i.e. c_i) in line $j=1$, respectively. Denote the computed correlation coefficient as ρ_k for $k \in [0250]$. The only information that ρ_k revealed was the average occurrence frequency of the difference value $\Delta = b^{(i)}[1] - b^{(i)}[0]$ for 251 kinds of different rotations.

Table 2
The frequency of Δ.

Δ	1	2	3	4	5	6	7	8	9	10	>10
Freq.	21	52	61	38	27	21	18	0	0	13	0

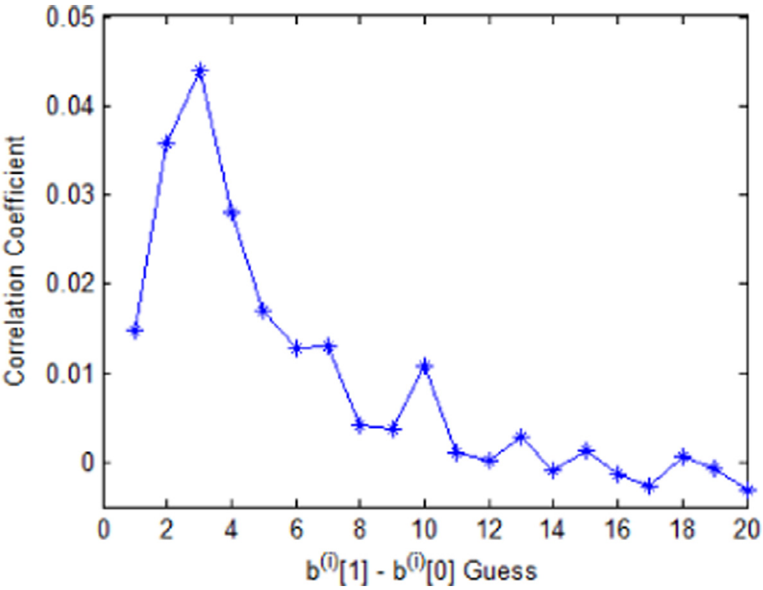

Fig. 5. Potential collision analysis on RKR.

Table 3
The frequency of $b^{(i)}[0]$.

$b^{(i)}[0]$	0	1	2	3	4	5	6	7	8	9	>9
Freq.	72	63	48	28	17	11	6	2	2	2	0

To verify this leakage, we implemented NTRU cryptosystem with RKR and random initialization of t on STC89C52 processor of MathMagic side-channel analyzer using the parameter $(N, d, q) = (251, 72, 128)$. We chose

$$\begin{aligned} b = [&0,\ 3,\ 7,\ 9,\ 13,\ 15,\ 17,\ 20,\ 23,\ 24,\\ &25, 29, 31, 33, 38, 41, 43, 47, 49, 52,\\ &59, 62, 63, 65, 66, 69, 71, 72, 75, 78,\\ &81, 83, 85, 88, 92, 95, 96, 97, 100, 102,\\ &105, 108, 114, 118, 121, 122, 127, 131, 135, 139,\\ &145, 147, 150, 153, 157, 162, 167, 171, 173, 179,\\ &189, 194, 199, 205, 212, 218, 225, 235, 238, 245,\\ &247, 250]. \end{aligned}$$

For this b, the expected occurrence frequency of Δ is shown in Table 2, whose total is 251.

Based on Hamming weight model, we performed an experiment with 100,000 traces, and the data in Fig. 5 verified our analysis. We tried to compute the correlation coefficients between 251 trace sections corresponding to the 251 c_k in line $j=0$ and that corresponding to $c_0^{(i)}$ in line $j=1$. The correlation coefficient in experiment could reflect the frequencies of Δ with low accuracy. However, even if the occurrence frequency could be recovered exactly from power analysis, the system is still secure due to the extremely high computation complexity from $\Delta = b^{(i)}[1] - b^{(i)}[0]$ to the key.

4.2. Evaluation by collision attack

According to Section 2.5, collision attack from Zheng et al. can recover $b[0]$ of unprotected NTRU. When it is applied to RKR, some $b^{(i)}[0]$ may be gotten, each of which means the offset of line $j=0$ in the current i. For the b above, we can infer the expected occurrence frequency of $b^{(i)}[0]$ listed in Table 3. Obviously, it follows the normal distribution, which means no useful information is leaked.

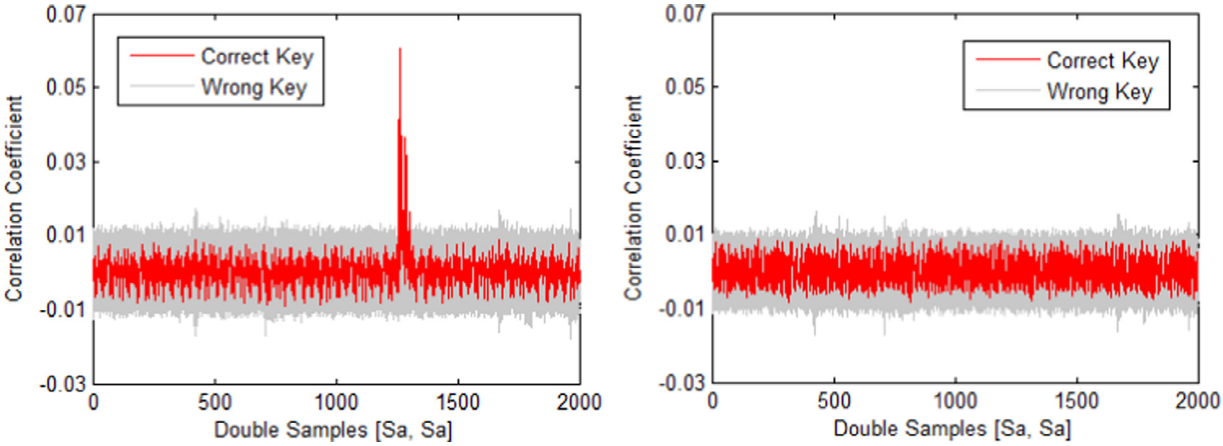

Fig. 6. Second-order CPA attack on Lee countermeasures (left) and RKR (right).

Table 4
The resistance of four implementations.

Methods	Timing / SPA	CPA	MIA / TA / FA	2O–CPA	CA
Unprotected	No	No	No	No	No
Lee et al.	Yes	Yes	–	Limited	No
Single RKR	No	Yes	Yes	Yes	Yes
Combine RKR with rand. t	Yes	Yes	Yes	Yes	Yes

4.3. Evaluation by second-order CPA

We assessed the security of random key rotation against the second-order CPA proposed by Lee et al. A practical attack using 50,000 power traces could not retrieve any information of the secret key, as shown in Fig. 6 (right). The red curve stands for the correlation coefficient corresponding to the right $b[1]-b[0]$ guess, while the grey curve represents that of the wrong guess. For reference, the second-order CPA can distinguish the correct key of Lee countermeasures, as shown in Fig. 6 (left).

4.4. Comparison of security

The RKR can protect NTRU from first-order statistical analysis such as CPA, template attack (TA), mutual information analysis (MIA), etc. This is because, for a fixed location (computational moment), its intermediate value c_k is uniformly distributed in the set $[c_0,\dots, c_{N-1}]$. Similarly, the fault attacker cannot accurately inject a fault to a certain intermediate value. Furthermore, with the help of random initialization of t, the combinational RKR countermeasure can resist timing attack and SPA natively.

We show the security of unprotected NTRU, three countermeasures from Lee et al., RKR, and combinational countermeasure of RKR and random initialization of t described in Section 3.4. For timing attack, SPA, CPA, TA, MIA, fault attack (FA), second-order CPA, and collision attack (CA), Table 4 shows whether the countermeasures can resist the corresponding attacks.

5. Implementation and efficiency

5.1. Software implementation

We implemented random key rotation based on STC89C52 with assembly language. Our RKR only costs two simple loops to rotate the key and ciphertext.

The efficiency comparison among our RKR, unprotected NTRU, and the combination of random initialization of t and randomization of b from Lee et al. is described in Table 5. The RAM occupied by RKR can be reused in the following convolution product computation, so our countermeasure does not need extra RAM.

5.2. Hardware implementation

In hardware, RKR can be implemented by two schemes efficiently. For fast implementation, barrel shifter can be employed for an arbitrary bits rotation. Before convolution product is computed, the key b and ciphertext c are rotated left and

Table 5
Efficiency comparisons in software.

Parameters		$N=8$, $d=4$	$N=251$, $d=72$	$N=251$, $d=48$
Time (clocks)	Unprotected	620	189,116	128,420
	Lee et al.	680	204,566	138,912
	RKR	694	191,056	130,216
ROM (bytes)	Unprotected	82	82	82
	Lee et al.	110	110	110
	RKR	109	109	109
Extra RAM (bytes)	Unprotected	0	0	0
	Lee et al.	8	251	251
	RKR	0	0	0

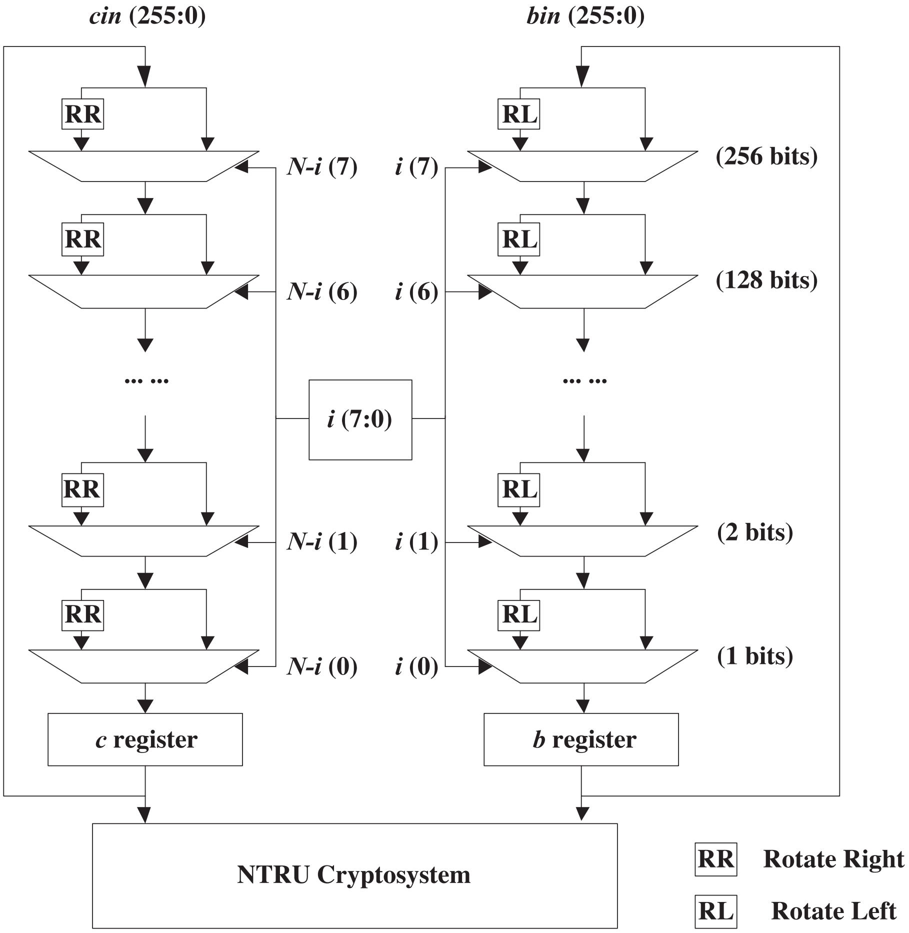

Fig. 7. Hardware implementation of RKR with barrel shifter.

right for i and $N-i$ bits respectively in a clock cycle. Fig. 7 shows this circuit. Here the rotation elements only include wires, which cost little resource.

For less area, a control module can be used for one-bit rotation loop. However, this implementation may cause an SPA due to its time dependence. Dummy rotations can resist this attack, i.e. N one-bit rotations are always carried out. If the regular i rotations of b are finished, the next $N-i$ rotations of b will be invalided by a multiplexer. Fig. 8 describes this circuit.

We compare our two hardware implementations with that of Lee et al, which is described in Table 6. Taking $N=251$, $d=72$ for example, our barrel shifter implementation only costs 8-bit extra register and only one extra clock cycle, while the implementation of Lee et al. cost 258-bit extra register and 74 extra clock cycles. The additional cost of our implementation is only 16 extra multiplexers.

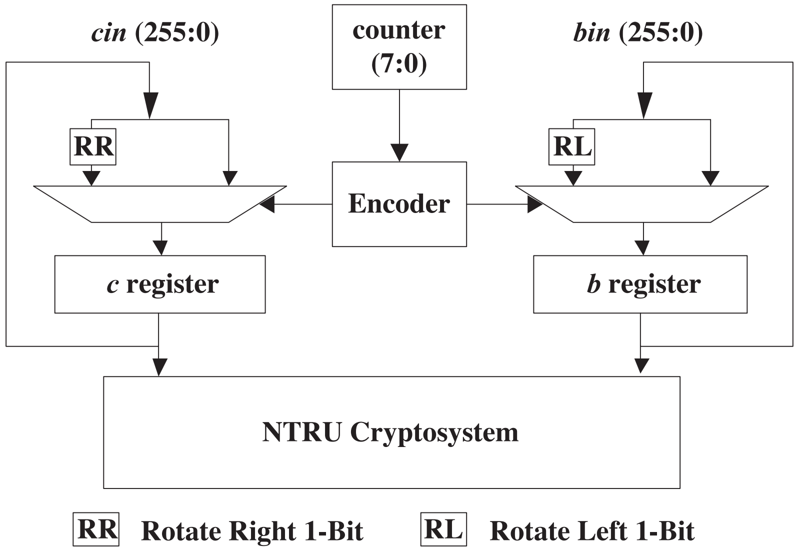

Fig. 8. Hardware implementation of RKR with repeating one-bit rotation.

Table 6
Efficiency comparisons in hardware.

Methods	Lee et al.	RKR (speed first)	RKR (area first)
Extra reg. (bits)	$N+log_2 d$	$log_2 N$	$log_2 N$
Extra comb. circuit	1 Adder	$2log_2 N$ MUX	2 MUX & 1 Controller
Extra time (clocks)	$2+d$	1	N

5.3. Application discussions

In the near future, NTRU will play more important roles in lightweight and post-quantum cryptography. NISTIR 8114 [22] describes the scope of the NIST standardization project for lightweight cryptography, and NISTIR 8105 [23] considers solutions such as NTRU to replace ECC and RSA, in order to achieve security even in the presence of quantum computers. Both projects are currently working on the standardization of new algorithms, for which the RKR protection on NTRU seems very well-suited. Furthermore, the combination or reconfigure of several public-key cryptosystems such as NTRU, RSA and ECC will implemented on the resource-constraint processors [24,25].

6. Conclusion

The existing NTRU countermeasures are not secure because invariability of secret key makes the statistical analysis feasible. From the idea of random change, we refer to the key randomization feature of the exponential mask of RSA algorithm and scalar multiplication mask of ECC algorithm. In this paper, we give an efficient NTRU countermeasure named random key rotation against all the existing power attacks. It shows very high efficiency, which is suitable for both hardware and software. There would be some other attacks against our RKR such as fault attacks. So, we will study the combinational attacks and countermeasures for higher security of NTRU implementation.

Acknowledgments

This work was supported by National Natural Science Foundation of China (Nos. 61402252, 61402536), Beijing Natural Science Foundation (No. 4162053), Foundation of Science and Technology on Information Assurance Laboratory (No. KJ-15-002), and Beijing Institute of Technology Research Fund Program for Young Scholars.

References

[1] Menezes AJ, Oorschot PC, Vanstone SA. Handbook of applied cryptography. CRC Press; 1997.
[2] Perrig A, Szewczyk R, Tygar JD, Wen V, Culler DE. Spins: security protocols for sensor networks. Wireless Netw 2002;8(5):521–34.
[3] Hoffstein J, Pipher J, Silverman J. NTRU: a new high speed public key cryptosystem. Rump session of Crypto; 1996. 1996.
[4] Hu F, Hao Q, Lukowiak M, Sun Q, Wilhelm K, Radziszowski S,WuY. Trustworthy data collection from implantable medical devices via high-speed security implementation based on IEEE 1363. IEEE Trans Inf Technol Biomed 2010;14(6):1397–404.
[5] Hu F, Wilhelm K, Schab M, Lukowiak M, Radziszowski S, Xiao Y. NTRU-based sensor network security: a low-power hardware implementation perspective. Security Comm Netw 2009;2:71–81.

[6] Hoffstein J, Pipher J, Silverman J. NTRU: a ring-based public key cryptosystem. Algorithmic number theory (ANTS III). LNCS 1423. Heidelberg: Springer; 1998. p. 267–88.
[7] Rourke C, Sunar B. Achieving NTRU with montgomery multiplication. IEEE Trans Comput 2003;52(4):440–8.
[8] Coppersmith D, Shamir A. Lattice attacks on NTRU, EUROCRYPT 1997. LNCS 1233. Heidelberg: Springer; 1997. p. 52–61.
[9] IEEE standard specification for public key cryptographic techniques based on hard problems over lattices. IEEE Std P1363.1-2008;2009.
[10] Gama N, Howgrave-Graham N, Nguyen P. Symplectic lattice reduction and NTRU, EUROCRYPT 2006. LNCS 4004. Heidelberg: Springer; 2006. p. 233–53.
[11] Howgrave-Graham N, Nguyen P, Pointcheval D, Proos J, Silverman J, Singer A, et al. The impact of decryption failures on the security of NTRU encryption, CRYPTO 2003. LNCS 2729. Heidelberg: Springer; 2003. p. 226–46.
[12] Jaulmes E, Joux A. A chosen-ciphertext attack against NTRU, CRYPTO 2000. LNCS 1880. Heidelberg: Springer; 2000. p. 20–35.
[13] Kocher P, Jaffe J, Jun B. Differential power analysis, CRYPTO 1999. LNCS 1666. Heidelberg: Springer; 1999. p. 388–97.
[14] Chari S, Rao J, Rohatgi P. Template attacks, CHES 2002. LNCS 2523. Heidelberg: Springer; 2002. p. 51–62.
[15] Schramm K, Wollinger T, Paar C. A new class of collision attacks and its application to DES, Fast Software Encryption 2003. LNCS 2887. Heidelberg: Springer; 2003. p. 206–22.
[16] Brier E, Clavier C, Olivier F. Correlation power analysis with a leakage model, CHES 2004. LNCS 3156. Heidelberg: Springer; 2004. p. 135–52.
[17] Atici A., Batina L., Gierlichs B., Verbauwhede I. Power analysis on NTRU implementations for RFIDs: first results. RFIDSec 2008:128-39, 2008.
[18] Atici A. Low cost NTRU implementations. Institute of Science and Technology, Istanbul Technical University; 2009.
[19] Lee MK, Song JE, Choi D, Han DG. Countermeasures against the power analysis attack for the NTRU public key cryptosystem. IEICE Trans Fund Electr Commun Comput Sci, E93-A 2010:153–63.
[20] Zheng X, Wang A, Wei W. First-order collision attack on protected NTRU cryptosystem. Microprocess Microsyst 2013;37(6-7):601–9.
[21] Bhargava A, Zoltowski M. Sensors and wireless communication for medical care. In: 14th international workshop on database and expert systems applications; 2003. p. 956–60.
[22] McKay KA, Bassham L, Turan MS, Mouha N. Report on lightweight cryptography; 2016. DRAFT NISTIR 8114 http://csrc.nist.gov/publications/drafts/nistir-8114/nistir_8114_draft.pdf NIST.
[23] Chen L, Jordan S, Liu Y, Moody D, Peralta R, Perlner R, et al. Report on post-quantum cryptography; 2016. NISTIR 8105 http://dx.doi.org/10.6028/NIST.IR.8105 NIST.
[24] Liu Z, Groszschaedl J, Hu Z, Jarvinen K, Wang H, Verbauwhede I. Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the Internet of Things. IEEE Trans Comput 2016;66(5):773–85.
[25] Liu Z, Huang X, Hu Z, Khan M, Seo H, Zhou L. On emerging family of elliptic curves to secure Internet of Things: ECC comes of age. IEEE Trans Depend Secure Comput 2016 In Press.

An Wang was born in 1983. He received his PH.D. degree in Shangdong University in 2011. From 2011 to 2015, he worked as a post doctor in Tsinghua University. He currently works in Beijing Institute of Technology. His main research interests include side-channel analysis, embedded system, and cryptographic implementation.

Ce Wang is currently studying in Beijing Institute of Technology. His research interests include side-channel analysis and countermeasures, as well as design and implementation of cryptographic hardware in general.

Xuexin Zheng was born in 1985. She received her B.S. degree in information security from Shangdong University in 2008. She currently works in China Academy of Electronics and Information Technology. She is mostly interested in lattice-based cryptography and side-channel attack.

Weina Tian was born in 1981. She received her PH.D degree in China Agriculture University in 2012. From 2012 to 2014, she worked as a post doctor in Peking University. She is currently working in Beijing Polytechnic. Her research interests include electronical and mathematical problems in engineering.

Rixin Xu is a PH.D student in Beijing Institute of Technology. His research includes side-channel analysis, cryptographic hardware implementation and security of the embedded system.

Guoshuang Zhang was born in 1982. He received his M.S. degree in the Zhengzhou Information Science and Technology Institute 2009. He is currently a Research Assistant in the Science and Technology on Information Assurance Laboratory. His main research interests include lattice-based cryptography and cryptanalysis.