Dynamic keys generation for internet of things

Omar Sapti Guma'a[1], Qasim Mohammed Hussein[2], Ziyad Tariq Mustafa Al-Ta'i[3]
[1,3]College of Science, Diyala University, Iraq
[2]College of Petroleum & Minerals Engineering, Tikrit University, Iraq

Article Info

Article history:

Received Aug 6, 2019
Revised Nov 7, 2019
Accepted Nov 21, 2019

Keywords:

Attack
IoT
lattice-based
LLL algorithm
NTRU
Security

ABSTRACT

In several aspects, interest in IoT has become considerable by researchers and academics in recent years. Data security becomes one of the important challenges facing development of IoT environment. Many algorithms were proposed to secure the IoT applications. The traditional public key cryptographic are inappropriate because it requires high computational. Therefore, lattice-based public-key cryptosystem (LB-PKC) is a favorable technique for IoT security. NTRU is one of a LB-PKC that based on truncated polynomial ring, it has good features, which make it to be an effective alternative to the RSA and ECC algorithms. But, there is LLL algorithm can success to attack it under certain conditions. This paper proposes modifications to NTRU public key cryptosystem to be secure against the lattice-based attack by using LLL algorithm, as well as a method for generating a new keys sequence dynamically. The results from simulations show that the performance of these modifications gives more secure from NTRU.

Corresponding Author:

Omar Sapti Guma'a,
College of Science, Diyala University,
Tikrit University Road, Tikrit, Salah Adin.
Email: omar.sapti@gmail.com

1. INTRODUCTION

The Internet of Things (IoT) is a modern concept, that all things in our daily life will have an ability to connect to the Internet and send or receive data to perform several specific functions [1]. There will be a hug exchange of data, including personal data and many data that need to be confidential. So, data security is an important challenge that should be taking a priority work to secure these data by encrypted them [2]. Since, the devices used in IoT are constrained devices (processing, memory, and power) [3], therefore, it is preferable to use encryption systems with lower computational power, less amount of storage in addition to less energy consumption. Several cryptographic systems are proposed to be used for IoT applications, including RSA, ECC and NTRU [4]. NTRU is a lattice-based public key cryptosystem that is based on the shortest vector problem in a lattice. Indeed, its cryptographic elementary operations require only polynomial multiplications, which are highly efficient and suitable for constrained devices [5]. Although the good features of NTRUs, there are some methods able to achieve some results in attacking it under certain conditions, such as Lenstra–Lenstra–Lovász lattice basis reduction algorithm (LLL) [6]. LLL algorithm [7] can be used to attack NTRU to discover either the original or an alternative secret key of it. In this paper there have been some additions and amendments to NTRU public key cryptography algorithm to ensure that the LLL algorithm does not succeed in attacking it, as well as generating dynamic new key sequences. It exploits the property of convolution multiplication, swapping any two values in one of the polynomial values will generate a new key that is completely different from the previous and has no relation between them. It is worth noting that these modifications do not effect on the complexity of the original NTRU.

There are many researchers interested in the security of data exchanged between IoT devices, and many research solutions offered, lattice-based cryptography considers as a one of these solutions [5].

Xuanxia et al. [8] depends on advantages of ECC (Elliptic Curve Cyptography) and the primitive syntax of KP-ABE (Attribute-Based Encryption where KP is Key Policy) to propose a suitable method for IoT environment. The proposed scheme is a lightweight one, which does not only have low communication overhead but also have low computational overhead.

Mingyuan Xin [9] proposed hybrid cipher algorithm that mixed encryption algorithm combines AES and ECC for IoT information security. Hybrid cipher algorithm in asymmetric cryptography and symmetric cryptography in one, with high security and fast speed, small storage space, more suitable for Internet of things some limited environment in such. But the application of the Internet of things is still in the exploration stage.

Harsh Durga et al. [10] proposes a new hybrid DNA-encoded ECC scheme that provides multilevel security. This proposed is more resilient to timing and SPA attacks and it is successfully applying on IoT devices.

Kavitha et al. [11] proposed a hybrid cryptographic algorithm that secures the sensitive medicinal information in IoT health care system, that include the combination of AES_HC and DH_HC. The performance analysis shows that the proposed key generation algorithm reduced computation time which provides faster execution significantly.

There are many proposed cryptosystems suggests modifications on NTRU, Table 1. illustrates that:

Table 1. A Review of Some Proposed Modifications for NTRU

Publishing	Principles
NNRU, a noncommutative analogue of NTRU [12]	It focus involves extension to noncommutative groups instead of using group algebra over Z_n (that is, the ring Z_q [X] / (X_n − 1)).It is has secure against lattice attacks with significant speed improvement.
OTRU: A non-associative and high speed public key cryptosystem [13]	It introduced a non-associative cryptosystem and proved that a lattice-based public key cryptosystem based on non-associative algebra is not only feasible. The OTRU lattice is not completely convolutional and the open problems and doubts which exist with respect to the cyclic structure of the NTRU lattice are not there in the case, the dimension of the lattice increases to 16N.
ETRU cryptosystem [14]	In this method, they used the concept of Euclidean domain and Dedekind domain. It is fast alternative to NTRU, offering comparable or better security for smaller key sizes and higher speed.
ILTRU cryptosystem [16]	It is based on ideal lattice [15] which is special structured lattices. This scheme was motivated by ETRU [14], which is used to make this scheme provably secure.
GR-NTRU cryptosystem [17]	It presented a method that based on group ring, called GR-NTRU to minimize the probability of decryption failure in the proposed scheme similarly as the original NTRU and the multivariate NTRU is most secure.
BTRU, A Rational Polynomial Analogue of NTRU Cryptosystem [18]	The role played by Z in NTRU replaced by the ring Q[α] of polynomial in one variable α over the Rational Field. The same value of N, CTRU is faster than NTRU, but not always.
NETRU: A Non-commutative and Secure Variant of CTRU Cryptosystem [19]	Presented a non-commutative extension of CTRU, they focus involves extension to non-commutative groups instead of using group algebra over Z2. NETRU cryptosystem is more secure than CTRU, because of its lattice structure and robustness against linear algebra attack.
Q-NTRU cryptosystem for IoT applications [20]	Proposed modifications on the NTRU cryptosystem algorithm to ensure that the attack by using Lenstra– Lenstra–Lovász (LLL) algorithm can be thwarted by adding a new parameter with a variable value. The implementation results showed that this modification gives NTRU resistance against the attack of LLL algorithm.

2. LATTICE-BASED CRYPTOGRAPHY

Lattice is a set of points in n-dimensional space with a periodic structure. More formally, given n-linearly independent vectors v1, ... ,vn ∈ Rn, the lattice generated by them is the set of vectors.

$$\mathrm{L}(\mathrm{v}_1,\ldots,\mathrm{v}_n) = \{\sum_{i=1}^{n} a_i \ v_i \ \big| a_i \ \in Z \}$$

The vectors v1,..., vn are known as a basis of the lattice [21].

A Signicant application such as mathematics and cryptography uses lattices. The main computational problem associated with lattices is the shortest vector problem (SVP). NTRU is classified as a lattice-based cryptosystem since its security is based on intractability of solving SVP/CVP (Shortest/Closest Vector Problem) in a particular type of lattice called Convolutional Modular Lattice [22].

Lattice-based cryptography is a new variant of post quantum cryptography [4]. Lattice-based constructions are currently important candidates for post-quantum cryptography. Unlike more widely used and known public-key schemes such as the RSA, Diffie-Hellman or Elliptic-Curve cryptosystems, which are consider suitable for IoT environment, but easily attacked by a quantum computer, while the lattice-based constructions appear to be resistant to attack by both classical and quantum computers [23]. Furthermore, many lattice-based constructions are considered to be secure under the assumption that certain well-studied computational lattice problems cannot be solved efficiently. Therefore, lattice-based public-key cryptosystem (LB-PKC) is a favorable technique for IoT applications security.

2.1. The LLL Algorithm

The LLL algorithm is the main technique in lattice reduction. In 1982, Arjen Lenstra, Hendrik Lenstra and L'aszl'o Lov'asz introduced the lattice basis reduction algorithm Lenstra-Lenstra-Lov'asz (LLL) [7], algorithm 1. This algorithm is the most famous algorithm for reducing a lattice basis and was a real breakthrough. It runs in polynomial time and gives a reduced basis of a lattice. It returns an approximation of the short vector. With an exponential approximation factor of 2(n-1)/2. It has good advantages for low dimensions and for many lattice problems approximation of the shortest vector or a reduced basis suffices. The breaking time appears experimentally to depend on the structure of f and g [24].

From another point of view, the LLL algorithm can discover either the original encryption key or an alternative secret key with the same decoding capability when the value of N is too small, while this algorithm cannot break the proposed encryption when the value of N is large [6].

3. THE PROPOSED ALGORITHM

To avoid weakness points in NTRU, this scheme presents a proposed modifications on the original NTRU: the first modification or append is swapping process between the public key values, this succeeded to make proposed algorithm more secure against LLL algorithm, and the second modification is generating a sequence private key that based on the public key, this makes the scheme uses independent public key for each block and no need to agree a new private polynomials between the sender and receiver. Figure 1. illustrates the flowchart of the proposed algorithm.

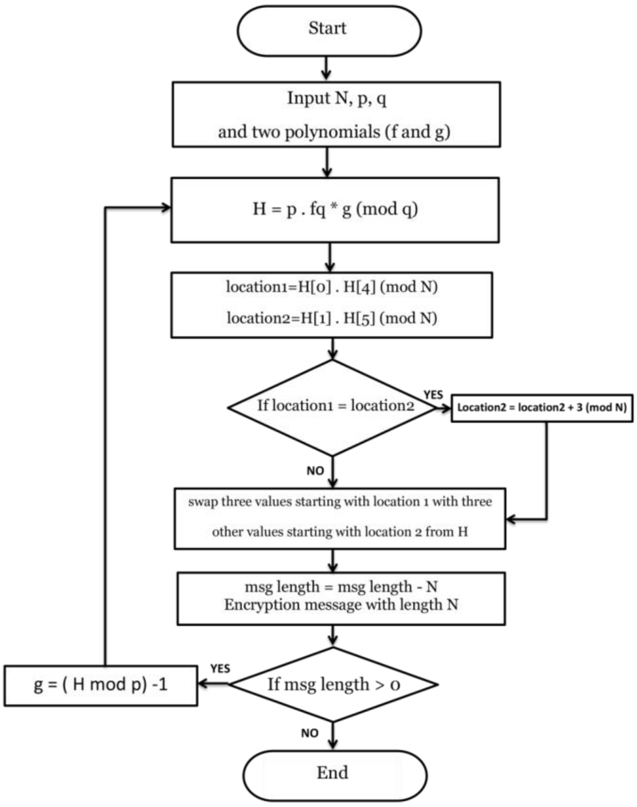
Start
Input N, p, q
and two polynomials (f and g)
H = p . fq * g (mod q)
location1=H[0] . H[4] (mod N)
location2=H[1] . H[5] (mod N)
If location1 = location2
YES
Location2 = location2 + 3 (mod N)
NO
swap three values starting with location 1 with three
other values starting with location 2 from H
msg length = msg length - N
Encryption message with length N
If msg length > 0
YES
g = (H mod p) -1
NO
End

Figure 1. Flowchart of the proposed algorithm

Where H[0], H[1], H[4], and H[5] are values in H (the locations 0, 1, 4, and 5 are agreement between sender and receiver).

A particular feature in convolution multiplication between two polynomials that changing any value in polynomial leads to a significant change in multiplication results, this feature of affairs has been exploited through swapping between the values of polynomial which is the public key, H, in order to obtain new polynomial that give completely different results than their predecessors. This feature does not require additional time or memory without effect on the security strength.

So, this paper uses a different private polynomials to generate the key sequence, public key, for each block of plaintext because repeating the same private polynomials to generate the keys will make the encryption process a weak, which is considered as a gift for the cryptoanalyzer, therefore thegenerated key, public key, must be changed dynamically.

3.1. Key Generation

The bellow algorithm illustrates the public key generation for proposed algorithm.

Algorithm 1. Public key Generation

```
Input: N, p, q, two polynomials (f and g)
Output: new public key (H)
1: Inverse Poly Fq(f , fq, N, q)
2: Inverse Poly Fp(f; fp, N, p)
3: polyMultiply(Fq, g, H, N, q)
4: for i = 0 to N – 1 do
5:     if H[i] < 0 then
6:         H[i] = H[i] + q // Make all coefficients in h positive.
7:     H[i] = H[i] . p mod q
8: end for
9: Location1 = ( H[0] . H[4] ) mod N
10: Location2 = ( H[1] . H[5] ) mod N
11: If Location1 = Location2 then
12:     Location2 = Location2 + 3
13: for i = 0 to 2 do
14:     Location1 = ( Location1 + i ) mod N
15:     Location2 = ( Location2 + i ) mod N
16:     temp = H[Location1]
17:     H[Location1] = H[Location2]
18:     H[Location2] = temp
19: end for
```

Inverse Poly is a function used to calculate the inverse of polynomial modulo q and p respectively, polyMultiply represent a function computes convolution multiplication of two polynomials modulo q. Location1 and location2 parameters represents the positions in the public key that will used to swap between values in public key, H. The loop from step 13 to step 19 used to swap three positions values of public key with other three positions values in it.

3.2. Encryption

An algorithm 3. illustrates encryption process for the proposal cryptosystem:

Algorithm 2. Encryption process

```
Input : N, q, Public Key H, message m, random polynomial r
Output : The encrypted message, e.
do
    for i = 0 to 2 do
        temp = H[Location1]
        H[Location1] = H[Location2]
        H[Location2] = temp
        Location1=( Location1-1) mod N
        Location2=( Location2-1) mod N
        if Location1 < 0 then
            Location1= Location1+N
        if Location2 < 0 then
            Location2= Location2+N
    end for
    PolyMultiplay ( r , H , e, N, q)
    for i = 0 to N - 1 do
        e[i] = e[i] + m[i] mod q
    for i = 0 to N – 1 do
        $g_i = (H_i \bmod p) - 1$
    Key generation (g) // Algorithm 2.
while (not encryption all data)
```

Firstly, the values that have been replaced should be returned to their original locations in public key, H, to get the original (H) that generated by convolution multiplication of the two private keys.

3.3. Decryption

The decryption process for this proposal is the same for the original NTRU without modifications.

Algorithm 3. Decryption process

```
Input : N, q, p, secret key f, inverse of polynomial Fp, and encrypted message e.
Output : The decrypted message, d.
polyMultiply(f, e, a, N, q)
for i = 0 to N – 1 do
    if a[i] < 0 then
        a[i] = a[i] + q // Make all coefficients positive
    if a[i] > q=2 then
        a[i] = a[i] – q //Shift coefficients into range (-q/2; q/2)
end for
polyMultiply(a, Fp, d, N, p)
```

Given an encrypted message e, compute a = f * e (mod q), and then b = a (mod p), and m = fp * b (mod p), where c is the original message m.

The use of a public key with a fixed length of N and then reuse of the same key will weaken the encryption process, to bypass this problem the proposed algorithm depends on the dynamic change for the value of H.

So, to generate a new public key (H1, H2… etc.) must choose a new private key, this paper proposed generating a new private key (polynomial g) from previous public key (Hi-1) by applying (1):

$$g = (H \; mod \; p) - 1 \tag{1}$$

The value of p is either 2 or 3, where p = 2 is used for binary representation, and p = 3 is used for ternary {-1, 0, 1}.

The proposed system can generate approximately (Np-1) versions of key sequence with length N from the original public key, this means that we do not repeat the public key with each block, but each block is encrypted with a key that is completely different from the key that precedes it and there is no relationship between them.

4. IMPELMENTATION AND RESULTS

Computer programs in C++ language were implemented on different samples for NTRU and our proposal with different values of their parameters, the public key, H, of these examples attacked by the LLL algorithm, to compare the security between them. The results showed the LLL algorithm discover the private key that generated by the original NTRU while cannot do that for our proposal algorithm, as shown in the following examples.

Example 1. This example shows the public keys that is generating by our proposal, when the parameters are (N, p, q) = (16, 3, 64) and f = -1 1 -1 1 0 -1 1 0 1 -1 1 0 0 1 0 -1 , g = 0 -1 0 0 0 0 0 1 -1 1 0 0 0 1 0 -1

Then the public key that generated by the original NTRU as follows:

H_0 = 42 17 40 34 26 30 46 29 15 33 57 35 45 48 46 33

When implementing the suggested method on H_0 will produce the following new public keys:

H_1 = 55 59 23 47 47 43 56 27 35 60 14 33 52 33 29 15
H_2 = 11 15 2 42 39 25 11 38 0 53 15 50 26 29 42 59
H_3 = 7 47 33 22 51 7 39 27 52 13 17 14 18 3 39 4
. . .
H_{11} = 45 49 17 53 37 50 53 56 17 11 30 41 32 21 22 36
. . . etc. and so on, until encrypted all message.

The new H that generated has no relationship between its values and other H values.

5. EXPERMINTAL ANALYSIS

The security of proposed algorithm analyzed to address the strength and weakness of it. The complexity of the proposed system is close to original NTRU. This means that the modification has not effect on the complexity and at the same time be more efficient in terms of security against lattice-based attack by LLL algorithm.

5.1. Brute Force attack

To recover the secret key, an attacker must attempt all possible F ∈ L_f and checking if it has small entries by F . H (mod N), or by attempting all G ∈ L_g and checking if it has small entries by G .H-1 (mod N) [KT16]. Also, to recover a message, an attacker must attempt all possible r ∈ L_r and checking if it has small entries by e.f – r * g (mod N). Practically, $L_g$will be lesser than L_f, therefore key security is fixed by L_g, and individual message security is fixed by L_r. So, the security for a key (both |Lg| and |Lf |) and the message is |Lr | [25].

$$security\ of\ key = \frac{N!}{d_g!^2 N-2d_g!}$$

$$security\ of\ message = \frac{N!}{d_r!^2 N-2d_r!}$$

5.2. Lattice-Based Attack

When applying the LLL algorithm to the proposed algorithm for this sheet, the private keys cannot be detected while the N value is too small, whereas when applied to the original NTRU, they detect the private keys. Example 2. demonstrate the efficiency of the proposed algorithm in crossing lattice-based attack:

Example 2. The parameters are (N , p, q) = (11, 3, 32) , f(x) = -1 + x + x^2 - x^4 + x^6 + x^9 - x^{10} and g(x) = 1 + x^3 -x^4 - x^5 - x^6 - x^8 - x^{10}

After applying the proposed algorithm to generate a public key:

H' = 1 5 27 23 12 27 22 15 25 8 18

The LLL Algorithm Outputs as shown in Table 2. Performing the LLL algorithm leads to a reduced basis like the following (L'):

Table 2. The LLL Algorithm Outputs

-1	-1	0	-2	1	1	0	2	-3	2	-2	-1	-4	2	0	1	0	1	-1	0	-1	-2
1	0	2	-1	-1	0	-2	3	-2	2	1	4	-2	0	-1	0	-1	1	0	1	2	1
1	0	2	-3	2	-2	-1	-1	0	-2	1	0	1	-1	0	-1	-2	-1	-4	2	0	1
1	0	1	-1	-1	0	-1	0	1	-4	3	0	1	3	-2	1	-1	0	-1	-2	6	4
0	0	-2	0	0	-1	-2	1	0	3	0	1	1	1	-4	3	2	-2	-2	2	6	1
-2	1	2	4	0	-1	-3	-2	1	0	0	-4	0	2	0	-4	2	-1	4	4	-3	0
-3	-3	4	-1	2	-2	-3	0	-1	1	2	-3	1	2	3	1	0	-2	2	-1	-4	5
2	1	0	1	-1	-2	-4	2	-1	2	0	-3	-4	0	2	-2	3	3	-1	3	3	-4
1	-2	-2	0	0	0	3	0	1	-2	0	-1	-1	-1	2	3	6	1	-2	-3	0	5
2	2	-1	0	0	-1	-2	1	-2	-1	-4	1	-1	0	-3	2	1	-4	-5	-2	1	0
-4	-2	0	-2	2	0	-1	-1	0	-3	-2	2	0	-1	-1	1	-3	-2	-2	-6	-1	2
-2	-1	-1	0	-2	1	1	0	2	-3	2	-2	-1	-4	2	0	1	0	1	-1	0	-1
-6	0	1	1	0	0	0	0	2	2	0	-1	2	-1	-2	2	0	1	1	1	1	-4
-1	-2	0	1	1	-3	3	-2	0	-1	0	-1	2	-1	0	6	1	-3	-4	0	1	3
-1	0	1	1	-1	-1	-2	-2	1	-5	1	2	-1	3	0	-5	1	1	-3	4	4	2
-1	-3	-2	-1	5	-1	-1	-2	0	3	3	2	4	-1	-3	0	-3	-1	1	1	0	0
0	1	-3	1	1	-1	1	3	-1	1	-3	0	-2	0	-2	3	4	-1	0	0	-1	-1
-2	1	1	0	2	-3	2	-2	-1	-1	0	0	1	0	1	-1	0	-1	-2	-1	-4	2
3	4	0	1	0	2	1	1	0	0	2	1	0	-4	2	0	4	0	-1	0	0	0
1	2	5	1	2	-1	-1	1	1	-1	1	0	1	0	0	-5	9	-1	0	2	-1	1
1	-1	0	2	0	-2	-1	-1	3	0	2	-3	-1	2	0	-3	0	2	-3	5	4	2
2	0	0	-1	-2	3	0	2	-3	1	1	0	1	-3	4	-3	-2	2	2	5	2	-3

The private keys (f, g) of the original NTRU cryptosystem and the proposed algorithm are sequence of the values {-1, 0, 1} only and the difference between number of (1) and (-1) is (1)(the number of "1" is exceed number of "-1" by 1) such as:

$$\sum_{k=1}^{N} row[k] = 1 \qquad (2)$$

5.3. Encryption Time and Decryption Time

The proposed system and the original NTRU were executed on the machine for the example 1 above. The specifications of this machine are (Operating system Windows 7 Ultimate 64-bit, processor Intel(R) Core i5 2.67GHz, and Installed memory (RAM) 4.00GB), this specifications gave the executing time (encryption and decryption) as illustrated in Figure 2:

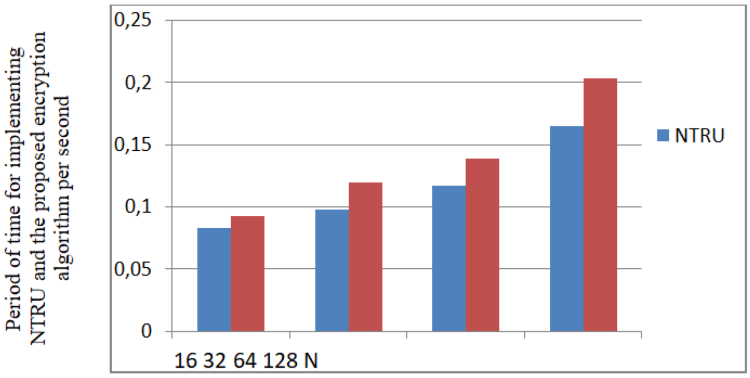

Figure 2. Performance timings of NTRU and the proposed algorithm

Figure 2. shows that there is no significant difference in time between the implementation of the original NTRU and the proposed algorithm. So, the complexity of encryption and decryption for both cryptosystem (original NTRU and modified) is O(N2).

6. CONCLUSION

The lattice-based public-key cryptosystem (LB-PKC) is a suitable technique for IoT security. NTRU is a one of lattice-based cryptosystem. This paper suggests some modifications on the original NTRU by performing swap process between the public key values in the key generation step and then return these values into the original locations before the encryption step, these modifications to avoid the lattice-based attack on the public key by LLL algorithm to discover the private key (f and g) as well as, this paper suggests a method to generate a new private keys from the previous public key, H. These modifications successed to make the proposed cryptosystem more secure and it made generating long sequences of key using dynamic process, futhermore it is more secure from the original NTRU because it used independent public key for each block of the message.

REFERENCES

[1] Mohd Muntjir, Mohd Rahul, and Hesham A. Alhumyani, "An Analysis of Internet of Things (IoT): Novel Architectures, Modern Applications, Security Aspects and Future Scope with Latest Case Studies", *International Journal of Engineering Research & Technology (IJERT)*, Vol. 6 Issue 06, June-2017, pp. 422-447.

[2] Nada Qasim Mohammed, Qasim Mohammed Hussein, Ahmed M. Sana, Layth A. Khalil, "A Hybrid Approach to Design Key Generator of Cryptosystem", *Journal of Computational and Theoretical Nanoscience*, Volume 16, Number 3, March 2019, pp. 971-977(7).

[3] Christian Dancke Tuen, "Security in Internet of Things Systems", *Norwegian University of Science and Technology*, 2015.

[4] R. Chaudhary, G. S. Aujla, N. Kumar and S. Zeadally, "Lattice-Based Public Key Cryptosystem for Internet of Things Environment: Challenges and Solutions," in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4897-4909, June 2019.

[5] Joel J. P. C. Rodrigues, Amjad Gawanmeh, Kashif and Sazia Parvin, "Smart Devices, Applications, and Protocols for the IoT", *USA, IGI Global, Engineering science reference(an imprint of IGI Global)*, 2019, p. 97.

[6] Qasim Mohammed Hussein, "Recover the NTRU Private Keys from Known Public Information and Public Key", PhD thesis, Tikrit university, 2009, p. 60.

[7] Arjen Klaas Lenstra, Hendrik Willem Lenstra, and La'szlo' Lova'sz. "Factoring polynomials with rational coefficients". *Mathematische Annalen*, 261(4):515-534, 1982.

[8] Xuanxia Yaoa, Zhi Chena, and Ye Tian, "A lightweight attribute-based encryption scheme for the Internet of Things", *Future Generation Computer Systems* 49 (2015) 104-112.

[9] M. Xin, "*A Mixed Encryption Algorithm Used in Internet of Things Security Transmission System*," 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Xi'an, 2015, pp. 62-65.

[10] Harsh Durga Tiwari and Jae Hyung Kim, "Novel Method for DNA-Based Elliptic Curve Cryptography for IoT Devices", *ETRI journal*, 40, 3, 2018.

[11] Kavitha.S, P.J.A.Alphonse, "A Hybrid Cryptosystem to Enhance Security in IoT Health Care System", *International Journal of Wireless and Microwave Technologies(IJWMT)*, Vol.9, No.1, pp. 1-10, 2019.

[12] Nitin Vats, "NNRU, a noncommutative analogue of NTRU", *arXiv e-prints*, February 01, 2009.

[13] E. Malekian and A. Zakerolhosseini, "*OTRU: A non-associative and high speed public key cryptosystem*," 2010 15th CSI International Symposium on Computer Architecture and Digital Systems, Tehran, 2010, pp. 83-90.

[14] Nevins M, Karimianpour C, Miri A. "NTRU over rings beyond Z. design". *Codes and Cryptography*, 2010; 56:65-78.

[15] Stehle D, Steinfeld R. "*Making NTRU as secure as worst-case problems over ideal lattices*:. In Proceeding of EUROCRYPT. LNCS, Vol. 6632. Springer, 2011; 27-47.

[16] Karabsi AH, Atani RE. "ILTRU: An NTRU-like public key cryptosystem over ideal lattices". *Cryptplogy ePrint* Archive: Report 2015/549. https://eprint.iacr.org/2015/549 .

[17] Yashoda T, Dahan X, Sakurai K. "Characterizing NTRU-variants using group ring evaluating their lattice security", 2015. *Cryptplogy ePrint* Archive: Report 2015/1170.https://eprint.iacr.org/2015/1170.

[18] Thakur, Khushboo. (2016). "BTRU, A Rational Polynomial Analogue of NTRU Cryptosystem". *International Journal of Computer Applications*. 145. 22-24. 10.5120/ijca2016910769.

[19] Ebrahimi Atani, Reza & Ebrahimi Atani, Shahabaddin & Hassani Karbasi, Amir. (2018). "NETRU: A Non-commutative and Secure Variant of CTRU Cryptosystem. International Journal of Information Security.

[20] Omar Sapti Guma'aa, Qasim Mohammed Husseinb, Ziyad Tariq Mustafa, "Q-NTRU Cryptosystem for IoT Applications", *Journal of Southwest Jiaotong University*, 54 (4), pp. 1-12, 2019.

[21] P. Q. Nguyen and J. Stern. "*The two faces of lattices in cryptology*". In Proc. Of CALC '01, volume 2146 of *LNCS*. Springer-Verlag, 2001.

[22] D. Coppersmith and A. Shamir, "Lattice attacks on NTRU," in *EUROCRYPT*, 1997, pp. 52-61.

[23] Cheng, Chi & Lu, Rongxing & Petzoldt, Albrecht & Takagi, Tsuyoshi. (2017). "Securing the Internet of Things in a Quantum World". *IEEE Communications Magazine*. 55. 116-120. 10.1109/MCOM.2017.1600522CM.

[24] Phong Q. Nguyen and Brigitte Valle. "The LLL Algorithm: Survey and Applications". *Springer Publishing Company*, Incorporated, 1st edition, 2009.

[25] Sonika Singh and Sahadeo Padhye, "Generalisations of NTRU cryptosystem", *Security And Communication Network Journal*, 9:6315-6334, 2016.

BIOGRAPHIES OF AUTHORS

Prof. Qasim Mohammed Hussein has PhD degree in computer sciences from Technology University, and currently Professor at Tikrit University. He published 26 journal articles in the fields of cryptography and computer security, He participated in writing (5) books in computer science, registering in Iraqi national library-the house of books and documentations. He reviewed dozens of scientific research to scientific upgrade in Iraq universities or for publication in the scientific journals, and a member of the preparation and scientific committees for scientific conferences. He was a member of the editorial board of Journal of Pure Science at the University of Tikrit, and expert in many committees, and contributed to a number of scientific and preparation committees of scientific conferences. He attended many conferences and scientific symposia, inside and outside of Iraq, and has a number of lectures and field studies in the field of computer.

Prof. Ziyad Tariq Mustafa Al-Ta'i received a B.Sc. degree in Electrical Engineering from Baghdad University (1987). In (1995) he obtained an M.S. degree in Computer Science from the University of Technology / Iraq. In (2002) got a Ph.D. in Computer Science from the University of Technology / Iraq. He had been a Professor in (2014) at the University of Diyala / Iraq. His research interests include network security and AI techniques.

Omar Sapti Guma'a is currently a M.S student at the department of Computer Science, college of science, Diyala University, Iraq/Diyala. He had received a B.Sc. degree in computer science from Tikrit University, College of Computer sciences and Mathematics, Iraq, Salah Adin (2010). His research interest includes data security, and network security.