electronics

MDPI

Article

Design and Evaluation of Device Authentication and Secure Communication System with PQC for AIoT Environments

Yu-Jen Chen [1], Chien-Lung Hsu [2,3,4,5,6], Tzu-Wei Lin [7,8] and Jung-San Lee [1,*]

[1] Department of Information Engineering and Computer Science, Feng Chia University, Taichung 407, Taiwan; p1200382@o365.fcu.edu.tw
[2] Graduate Institute of Business and Management, Chang Gung University, Taoyuan 333, Taiwan; clhsu@mail.cgu.edu.tw
[3] Department of Information Management, Chang Gung University, Taoyuan 333, Taiwan
[4] Healthy Aging Research Center, Chang Gung University, Taoyuan 333, Taiwan
[5] Department of Visual Communication Design, Ming-Chi University of Technology, New Taipei City 243, Taiwan
[6] Department of Nursing, Chang Gung Memorial Hospital, Taoyuan 333, Taiwan
[7] i. School, Feng Chia University, Taichung 407, Taiwan; tweilin@fcu.edu.tw
[8] Information Security Office, Office of Information Technology, Feng Chia University, Taichung 407, Taiwan
* Correspondence: leejs@fcu.edu.tw

Abstract: With the rapid development of Internet of Things (IoT) technology, the number of IoT users is growing year after year. IoT will become a part of our daily lives, so it is likely that the security of these devices will be an important issue in the future. Quantum computing is maturing, and the security threat associated with quantum computing will be faced in the transmissions of IoT devices, which mainly use wireless communication technologies. Therefore, to ensure the protection of transmitted data, a cryptographic algorithm that is efficient in defeating quantum computer attacks needs to be developed. In this paper, we propose a device authentication and secure communication system with post-quantum cryptography (PQC) for AIoT environments using the NTRU and Falcon signature mechanism, which can resist quantum computer attacks and be used in AIoT environments to effectively protect the confidentiality, integrity, and non-repudiation of transmitted data. We also used Raspberry Pi to simulate AIoT devices for implementation.

Keywords: AIoT; device authentication; post-quantum cryptography; NTRU; falcon signature

Citation: Chen, Y.-J.; Hsu, C.-L.; Lin, T.-W.; Lee, J.-S. Design and Evaluation of Device Authentication and Secure Communication System with PQC for AIoT Environments. *Electronics* **2024**, *13*, 1575. https://doi.org/10.3390/electronics13081575

Academic Editors: Chuan Zhang, Xintao Huan, Heng Wang, Yan Zong and Guyue Li

Received: 15 January 2024
Revised: 17 April 2024
Accepted: 18 April 2024
Published: 20 April 2024

1. Introduction

Artificial intelligence (AI) has the potential to have widespread impacts and applications, and its development has been rapidly growing, including the implementation of strategies, policies, and their adoption [1,2]. The energy consumption of 5G networks can be reduced while still allowing for communication among a large number of devices, such as hundreds or thousands of sensors in Internet of Things (IoT) networks [2,3]. Although the 5G-IoT environment introduces potential possibilities in developments and applications in many fields due to the convenience and quality of services, the security and privacy of transmitted data are of utmost importance because data in IoT networks are transmitted through wireless communication [4]. Because wireless networks are easily attacked compared with wire networks, serious information security events will be faced by wireless networks, such as privacy leakage, man-in-the-middle attacks, data tempering, etc. The rapid progress of AIoT has led to the discussion on security concerns due to the distinct networking structure and method of acquiring and storing data [5,6]. Continuously capturing and gathering data increases security risks because devices are always online, and adversaries have unlimited opportunities to attack AIoT systems.

Quantum computing technology has attracted researchers and famous manufacturers' attention, such as IBM, Google, Microsoft, Hon Hai Precision Industry, Alibaba, etc., and

has been developed rapidly; quantum computing technology will change people's daily life [7]. On the other hand, the risk of traditional cryptographic mechanisms being broken by quantum computing technology via Shor's algorithm [8] is increasing. Post-quantum cryptography (PQC) is a cryptographic solution that can prevent attacks from quantum computers using modern computers to prevent such risks before quantum computers are produced rapidly. The proposed scheme utilized one PQC mechanism called the number theory research unit (NTRU) [9] and fast-Fourier lattice-based compact signatures over the NTRU (the Falcon signature mechanism) [10] to design secure communication mechanisms.

Due to the reasons above, we designed and evaluated a device authentication and secure communication system with PQC for AIoT environments. We utilized the NTRU [9] and Falcon signature mechanisms [10] because both can resist attacks from quantum computers with better speeds of encryption, decryption, and sign mechanisms than those of traditional cryptographic mechanisms [11]. In summary, the proposed scheme can protect the security and privacy of transmitted data while resisting attacks from quantum computers. The remaining sections of the paper are outlined below. AIoT, post-quantum cryptography, the NTRU [9], and the Falcon signature [10] are introduced in Section 2. Section 3 introduces the proposed scheme, and security and performance analysis are detailed in Sections 4 and 5, respectively. Section 6 describes system implementation. We present a discussion on the research results and limitations of this research in Section 7. Finally, the conclusion is drawn in Section 8.

2. Related Works

We introduce and review AIoT, post-quantum cryptography, the NTRU [9], and the Falcon signature [10] in this section.

2.1. AIoT

In many areas, AIoT is expected to improve the quality of services of the industry because devices in the Internet of Things have the ability to simulate human intelligence and support decision making by continuously learning from large amounts of data [5,12]. AIoT opens up possibilities for applications in all kinds of fields because of its transparency, agility, and adaptability [6,12]. The adoption of AIoT ends up on top of the to-do list of many applications, such as smart homes, smart factories, smart cities, and so on [12–19]. However, AIoT also provides opportunities for adversaries. Once an AIoT system is established, all related devices will exist in the network until the system is shut down. Adversaries can attack any device in wireless networks through various means, such as eavesdropping, the tempering of transmitted data, impersonation, etc. The proposed scheme is designed to be implemented in AIoT environments for transmitted data protection and device authentication.

2.2. Post-Quantum Cryptosystem

The development of quantum computing is expected to have a huge impact on information technology in the future. One of the applications, called quantum cryptanalysis, will be a serious threat to information security because of its ability to attack public key cryptosystems, such as RSA, the Diffie–Hellman cryptosystem, elliptic curve cryptography (ECC), etc., through Shor's algorithm [8]. The post-quantum cryptosystem can be a solution for resisting attacks from quantum computers. The post-quantum cryptosystem has advantages, which are listed below. First, the costs of establishment and development are lower than that of a quantum cryptosystem. The post-quantum cryptosystem can be executed by modern computers and systems. Second, public key encryption and digital signature mechanisms can be applied by post-quantum cryptosystems. Post-quantum cryptosystems include lattice-based, hash-based, code-based, multivariate, and supersingular elliptic curve isogeny cryptography. As the matter of fact, post-quantum cryptosystems will replace traditional public key cryptosystems soon. The proposed scheme utilizes lattice-based cryptosystems to design a secure communication mechanism.

2.3. NTRU

The NTRU is a latticed-based public key cryptosystem including encryption and signature [9]. The security of the NTRU depends on the shortest vector problem (SVP) in the lattice. The SVP is defined as the discovery of a datum point that has the shortest distance from the base point [9]. The NTRU can resist attacks from Shor's algorithm [8] while maintaining attractive features, such as better encryption and decryption speed, smaller key size, and higher security compared with those of traditional cryptosystems [9]. The NTRU is proven as one of the quantum-resistant cryptographic algorithms by National Institute of Standards and Technology (NIST) [9,20]. Although the NTRU has failed to compete among the post-quantum cryptography standardization finalists, it is still widely used because of its advantages over other lattice based cryptography systems [9,20,21]. The NTRU is more efficient than traditional cryptographic mechanisms and can be implemented in devices with restricted resources, such as AIoT, embedded devices, etc. [22–24]. A property comparison of RSA, ECC, and NTRU [9] is presented in Table 1 [25]. Compared with RSA and ECC, all algorithms can achieve an encryption and signature mechanism, but RSA cannot achieve a key exchange mechanism. The NTRU has a faster encryption speed than RSA and ECC. RSA and the NTRU are easier to use in key distribution than ECC is. Among the three algorithms, only the NTRU is a quantum-resistant cryptographic algorithm.

Table 1. Property comparison of RSA, ECC, and NTRU [25].

Properties \ Algorithms	RSA	ECC	NTRU
Encryption	O	O	O
Signature	O	O	O
Key exchange	X	O	O
Encryption speed	Slow	Fast	Fastest
Key distribution	Easy	Difficult	Easy
Quantum-resistant	X	X	O

ECC and the NTRU [9] can provide the same security level as RSA can with a shorter key length, so ECC and the NTRU [9] need much less storage and a much shorter transmission time. These attractive features are advantages for devices with restricted resources. Compared with RSA, ECC, and the NTRU [9], ECC has the shortest key length at the same security level. The NTRU [9] has a shorter key length than RSA does under a security level of 192 bits and 256 bits. The comparison of key lengths of RSA, ECC, and the NTRU [9] at different security levels is shown in Table 2 [26] and Figure 1 [26].

Table 2. Key length of RSA, ECC, and NTRU with different security levels [26].

Security Level (bits) \ Algorithms	RSA	ECC	NTRU
80	1024	160	2008
112	2048	224	3033
128	3072	256	3501
192	7680	384	5193
256	15,360	521	7690

Although the NTRU [9] needs more time to generate a key, the NTRU [9] needs less time to encrypt and decrypt than ECC does at a security level of 80 bits. The NTRU [9] is faster than ECC is in key generation, encryption, and decryption at a security level of 112 bits and beyond. The execution times of ECC and the NTRU [9] at different security levels are shown in Table 3 [26]. The proposed scheme utilized the NTRU [9] for encryption and decryption mechanisms. For details of the NTRU, readers can refer to Hoffstein et al.'s work [9].

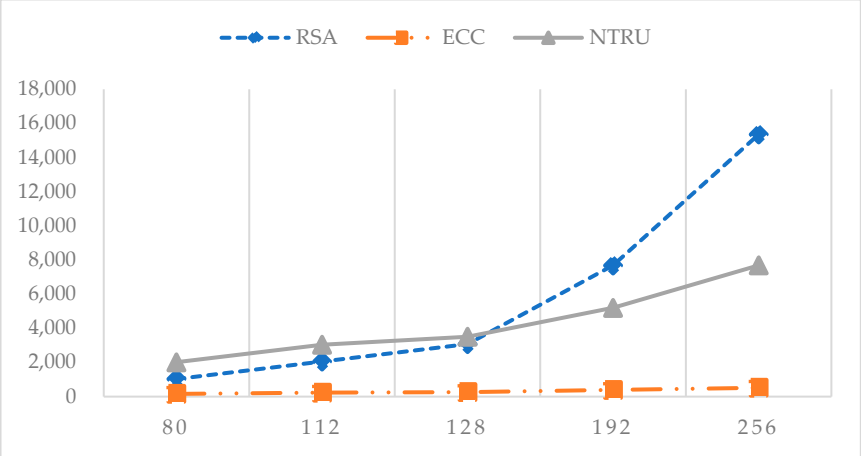

Figure 1. Key length of RSA, ECC, and NTRU with different security levels [26].

Table 3. Execution time of ECC and NTRU with different security levels [26].

Items / Algorithms	Security Level (bits)	Key Generation (ms)	Encryption (ms)	Decryption (ms)
NTRU-251	80	75.65	1.68	8.22
ECC-192	80	57.87	37.81	19.15
NTRU-347	112	144.16	3.11	15.70
ECC-224	112	234.11	52.52	26.35
NTRU-397	128	188.92	3.97	20.26
ECC-256	128	478.22	68.72	35.00
NTRU-587	192	412.10	8.42	44.42
ECC-384	192	947.43	182.35	90.61
NTRU-787	256	738.75	14.49	79.48
ECC-521	256	2055.04	423.25	211.35

2.4. Falcon Signature Mechanism

The Falcon signature mechanism adopts a trapdoor function named fast Fourier sampling [10]. The Falcon signature mechanism is also proven by NIST to be one of the quantum-resistant cryptographic algorithms [10,20]. The security of the Falcon signature mechanism is based on the short integer solution (SIS) [10]. Features of the Falcon signature mechanism [10] are listed as below. First, the Falcon signature mechanism [10] utilizes discrete Gaussian sampling over the integers, which is able to avoid the key exposure problem while generating multiple signatures. Second, signatures generated by the Falcon signature mechanism [10] are shorter than those generated by other lattice-based signature mechanisms with the same public key length. Third, the Falcon signature mechanism [10] can generate thousands of signatures in a few seconds with a verification speed about 5 to 10 times faster than that of other signature mechanisms. Forth, the Falcon signature mechanism [10] allows the use of long-term security parameters with the same time complexity $O(n\log n)$ under degree n. Although the Falcon signature mechanism [10] is intended to defend against quantum computer attacks, due to its high efficiency, its [10] use has become widespread. For details of the Falcon signature mechanism, readers can refer to Fouque et al.'s work [10]. The proposed scheme utilized the Falcon signature mechanism [10] for message verification.

3. Proposed Scheme

We design and evaluate a device authentication and secure communication system with PQC for AIoT environments. IoT devices in the proposed system capture and send data through wireless networks to the gateway. The gateway can be a mobile phone, an IoT gateway, a stand-alone laptop computer, etc. After receiving data from IoT devices, the gateway transmits data to the server.

3.1. System Structure

The system of the proposed scheme includes IoT devices, a gateway, a server, and a smart token. The smart token stores the private key and parameters of the gateway securely and output parameters after decryption and signing mechanisms come into effect. After that, data will be sent to server. The system structure of the proposed scheme is illustrated in Figure 2.

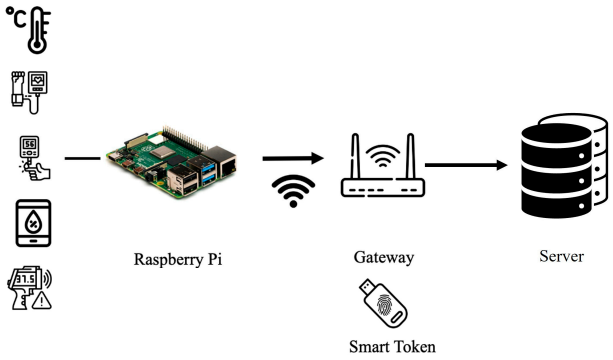

Figure 2. System structure of proposed scheme.

The proposed scheme includes four phases. In the preliminary phase, the system generates essential parameters and functions. IoT devices are registered on the server using a MAC address, and the server generates and distributes private keys, public keys, and initial values to IoT devices in the registration phase through a secure channel. In an IoT device's gateway communication phase, it captures and encrypts data with the public key of the gateway, and then generates a signature with the private key of the IoT device itself. Then, the IoT device sends encrypted data and the signature to the gateway. The gateway verifies the signature using the public key of the IoT device and decrypts data using the private key in the smart token. In the gateway's server communication phase, the gateway sends encrypted data and the signature to the server. After receiving encrypted data and verifying the signature, the server stores the encrypted data and signature. Notations of the proposed scheme are shown in Table 4.

Table 4. Notations of proposed scheme.

Notations	Definitions
h_D, h_G, h_S	Public key polynomial vectors for encryption and decryption of IoT devices, gateway, and server respectively.
f_D, f_{p_D}	Private key polynomial vectors for encryption and decryption of IoT devices.
f_G, f_{p_G}	Private key polynomial vectors for encryption and decryption of gateway.
f_S, f_{p_S}	Private key polynomial vectors for encryption and decryption of server.
r_D, r_G	Random polynomials.
m	Message, which is a polynomial.
e_{DG}, e_{GS}	Encrypted data, which are polynomials.
h_{s_D}, h_{s_G}, h_{s_S}	Public key polynomial vectors for signatures of IoT devices, gateway, and server respectively.
$(\hat{B}_D, T_D)$	Private key polynomial vectors for signatures of IoT devices.
$(\hat{B}_G, T_G)$	Private key polynomial vectors for signatures of gateway.
r_{s_D}, r_{s_G}	Random polynomials for signatures.
sig_{DG}, sig_{GS}	Signatures.
Q	Random integer for signature.
n	Degree of lattice polynomial.
β	Bound of vector.

3.2. Preliminary

A truncated polynomial, R, with degree $N-1$ is defined as $R = (a_0 + a_1x + a_2x^2 + \ldots + a_{N-1}x^{N-1}) \bmod (x^{N-1})$, where N is a positive integer and the highest degree of R. q and p are positive integers. q and p are coprime, and p is smaller than q.

3.3. IoT Devices' Gateway Communication Phase

An IoT device captures and encrypts data with the public key of the gateway, and then generates a signature with the private key of the IoT device itself. Then, the IoT device sends the encrypted data and signature to the gateway. The gateway verifies the signature using the public key of the IoT device and decrypts data using the private key in the smart token. The gateway is a receiver in the phase. The gateway has an NTRU-based private and public key, obtaining this by randomly choosing polynomials f and g, which have to be secret. The gateway obtains the private keys (f_G and f_{p_G}) and public key, h_G. Detailed descriptions are given in the following and illustrated in Figure 3.

IoT Device	Gateway
1. Randomly choose r_D	
2. $e_{DG} = r_D * h_G + m \pmod q$	
3. Randomly Salt $r_{s_D} \leftarrow \{0, 1\}^{320}$	
4. $c_{DG} \leftarrow Hashtopoint(r_{s_D} \| \| m)$	
5. $t_{DG} \leftarrow (FFT(c_{DG}), FFT(0))\hat{B}_D^{-1}$	
6. $z_{DG} \leftarrow ffSampling_n(t_{DG}, T_D)$	
7. $s_{DG} = (t_{DG} - z_{DG})\hat{B}_D$	
8. While $\| \| s_{DG} \| \|^2 > \lfloor \beta^2 \rfloor$	
9. $(s_{1_{DG}}, s_{2_{DG}}) \leftarrow invFFT(s_{DG})$	
10. $s'_{DG} \leftarrow$ Compress$(s_{2_{DG}})$	
11. Return $sig_{DG} = (r_{s_D}, s'_{DG})$	
12. $(r_{s_D}, e_{DG}, sig_{DG})$ →	
	13. $a_{DG} = f_G * e_{DG} \pmod q$
	14. $m = f_{p_G} * a_{DG} \pmod q$
	15. $c_{DG} \leftarrow Hashtopoint(r_{s_D} \| \| m, q, n)$
	16. $s_{2_{DG}} \leftarrow$ Decompress(s'_{DG})
	17. $s_{1_{DG}} \leftarrow c_{DG} - s_{2_{DG}} h_{s_D} \bmod q$
	18. $\| \| s_{1_{DG}}, s_{2_{DG}} \| \|^2 \leq \lfloor \beta^2 \rfloor$

Figure 3. IoT devices' gateway communication phase.

Step 1: The IoT device randomly chooses polynomial r_D and computes encrypted data, e_{DG}.

$$e_{DG} = r_D * h_G + m(\text{mod } q) \tag{1}$$

Step 2: The IoT device randomly generates salt $r_{s_D} \leftarrow \{0, 1\}^{320}$ and utilizes r_{s_D} and m to generate $c_{DG} \leftarrow Hashtopoint(r_{s_D} \| m)$. Then, the IoT device computes (t_{DG}, z_{DG}, s_{DG}) via fast Fourier transforming (*FFT*) and fast-Fourier sampling (*ffSampling*) functions and checks if s_{DG} is in bound β via $\|s_{DG}\|^2 > \lfloor \beta^2 \rfloor$.

$$t_{DG} \leftarrow (FFT(c_{DG}), FFT(0)) * \hat{B}_D^{-1} \tag{2}$$

$$z_{DG} \leftarrow ffSampling_n(t_{DG}, T_D) \tag{3}$$

$$s_{DG} = (t_{DG} - z_{DG})\hat{B}_D \tag{4}$$

Step 3: The IoT device utilizes s_{DG} to generate ($s_{1_{DG}}$ and $s_{2_{DG}}$) via inverse fast Fourier transforming. $s_{2_{DG}}$ is compressed to string s'_{DG}. After that, the IoT device generates signature sig_{DG} and sends (r_{s_D}, e_{DG} and sig_{DG}) to the gateway.

$$sig_{DG} = (r_{s_D}, s'_{DG}) \tag{5}$$

Step 4: After receiving (r_{s_D}, e_{DG} and sig_{DG}), the gateway computes a_{DG}. The coefficient of a_{DG} will be between $-q/2$ and $q/2$. Then, the gateway utilizes f_{p_G} to recover m.

$$a_{DG} = f_G * e_{DG} \pmod q \tag{6}$$

$$m = f_{p_G} * a_{DG} \pmod p \tag{7}$$

Step 5: The gateway utilizes (m, r_{s_D}, q and n) to generate $c_{DG} \leftarrow HashToPoint(r_{s_D} \| m, q, n)$ and decompress s'_{DG} to $s_{2_{DG}}$. Then, the gateway utilizes (c_{DG}, $s_{2_{DG}}$, h_{s_D} and q) to compute $s_{1_{DG}}$ and checks if ($s_{1_{DG}}$ and $s_{2_{DG}}$) is in bound β via $\left\| (s_{1_{DG}}, s_{2_{DG}}) \right\|^2 \leq \lfloor \beta^2 \rfloor$ or rejects verification.

$$s_{1_{DG}} \leftarrow c_{DG} - s_{2_{DG}} h_{s_D} \bmod q \tag{8}$$

3.4. Gateways' Server Communication Phase

The gateway sends the encrypted data and signature to the server. Detailed descriptions are given in the following and illustrated in Figure 4.

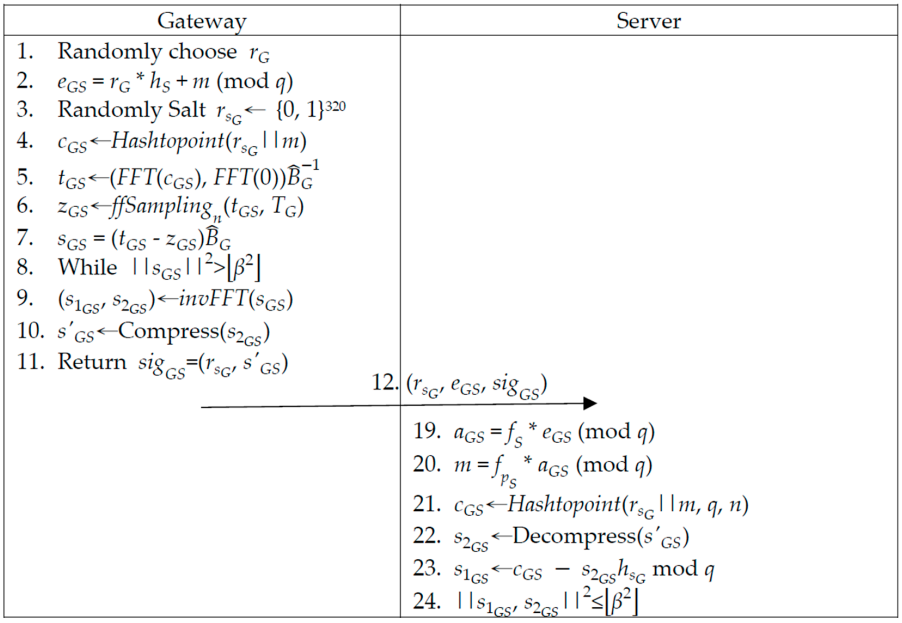

Figure 4. Gateways' server communication phase.

Step 1: The gateway randomly chooses polynomial r_G and computes the encrypted data, e_{GS}.

$$e_{GS} = r_G * h_S + m (\bmod\ q) \tag{9}$$

Step 2: The gateway randomly generates salt $r_{s_G} \leftarrow \{0, 1\}^{320}$ and utilizes r_{s_G} and m to generate $c_{GS} \leftarrow Hashtopoint(r_{s_G} \| m)$. Then, the gateway computes (t_{GS}, z_{GS} and s_{GS}) using *FFT* and *ffSampling*, and checks if s_{GS} is in bound β via $\|s_{GS}\|^2 > \lfloor \beta^2 \rfloor$.

$$t_{GS} \leftarrow (FFT(c_{GS}), FFT(0)) * \hat{B}_G^{-1} \tag{10}$$

$$z_{GS} \leftarrow ffSampling_n(t_{GS}, T_G) \tag{11}$$

$$s_{GS} = (t_{GS} - z_{GS})\hat{B}_G \tag{12}$$

Step 3: The gateway utilizes s_{GS} to generate ($s_{1_{GS}}$ and $s_{2_{GS}}$) via inverse fast Fourier transform. $s_{2_{GS}}$ is compressed to string s'_{GS}. After that, the gateway generates signature sig_{GS} and sends (r_{s_G}, e_{GS} and sig_{GS}) to the server.

$$sig_{GS} = (r_{s_G}, s'_{GS}) \tag{13}$$

Step 4: The after receiving (r_{s_G}, e_{GS} and sig_{GS}), the server computes a_{GS}. The coefficient of a_{GS} will be between $-q/2$ and $q/2$. Then, the server utilizes f_{p_G} to recover m.

$$a_{GS} = f_S * e_{GS} \pmod{q} \tag{14}$$

$$m = f_{p_S} * a_{GS} \pmod{p} \tag{15}$$

Step 5: The server utilizes (m, r_{s_G}, q and n) to generate $c_{GS} \leftarrow HashToPoint(r_{s_G} \| m, q, n)$ and decompress s'_{GS} to $s_{2_{GS}}$. Then, the server utilizes (c_{GS}, $s_{2_{GS}}$, h_{s_G} and q) to compute $s_{1_{GS}}$ and checks if ($s_{1_{GS}}$ and $s_{2_{GS}}$) is in bound β via $\left\| (s_{1_{GS}}, s_{2_{GS}}) \right\|^2 \leq \lfloor \beta^2 \rfloor$ or rejects verification.

$$s_{1_{GS}} \leftarrow c_{GS} - s_{2_{GS}} h_{s_G} \mathrm{mod} q \tag{16}$$

4. Security Analysis

We analyze the proposed scheme below.

4.1. Correctness

Receivers in the proposed scheme (the gateway and server) compute polynomials a_{DG} and a_{GS}, respectively, where coefficients in both polynomials are between $-q/2$ and $q/2$. The whole computation is represented in Equation (17). After that, receivers recover message m, resulting in Equation (18). While recovering message m, because $f_p * f = 1$, the recovery of message m will be successful.

$$\begin{aligned} a &= (f * e) \mathrm{mod}\ q \\ &\equiv (f * (r * h + m)) \mathrm{mod}\ q \\ &\equiv ((f * r * p\ *\ f_q * g) + (f * m)) \bmod q \\ &\equiv ((r * p * g) + (f * m)) \bmod q \end{aligned} \tag{17}$$

$$\begin{aligned} m &\equiv (f_p * a) \mathrm{mod}\ p \\ &\equiv f_p * ((r * p * g) + (f * m)) \bmod p \\ &\equiv (f_p * r * p * g) + (f_p * f * m) \pmod{p} \\ &\equiv m (\mathrm{mod}\ p) \end{aligned} \tag{18}$$

4.2. Confidentiality

If an adversary aims to recover a message through sniffing in the IoT device's gateway communication phase, the adversary must have knowledge of the private key of gateway f_{p_G} according to Equation (7). f_{p_G} is stored in the smart token securely. Similarly, we can prove that an adversary cannot recover message m in the gateway's server communication phase because they lack knowledge of the private key of server f_{p_S} according to Equation (15). As a result, the proposed scheme can achieve confidentiality.

4.3. Integrity

Assuming that an adversary aims to modify transmitted messages, they must forge the signature generated through the Falcon signature mechanism [10]. The adversary cannot generate a signature without knowing the private key of gateway f_{p_G} and server f_{p_S}. As a result, integrity can be achieved by verifying the signatures.

4.4. Non-Repudiation

Because of the utilization of the signature mechanism, the IoT device and gateway cannot deny sending the message. IoT devices use private key polynomial vectors for signatures ($\hat{B}_D$ and T_D) to generate s_{DG} as a part of signature sig_{DG}, and the gateway verifies sig_{DG} using public key polynomial vectors for the signature of IoT device h_{s_D}. Because sig_{DG} can only be verified using h_{s_D}, IoT devices cannot deny sending the message to the gateway. By the same token, the gateway cannot deny sending the message to the server. As a result, non-repudiation can be achieved.

5. Performance Analysis

We analyzed the performance of NTRUEncrypt [9,27] and the Falcon signature mechanisms [10,28] with different security levels utilized in the proposed scheme. We used a personal computer (PC) with an i7-7000 3.60 GHZ 8-core central processing unit (CPU, Intel Corporation, Santa Clara, CA, USA), 32 GB random-access memory (RAM, Kingston Technology Corporation, Fountain Valley, CA, USA), and Windows 10 Education as an operation system (OS, Microsoft Corporation, Washington, DC, USA). We also used a Raspberry Pi 3B module (Raspberry Pi Foundation, Cambridge, UK) with an ARM Cortex-A53 1.4 GHz 4-core CPU (Arm Holdings plc, Cambridge, UK), 1 GB RAM (Micron Technology, Inc., Boise, US), and Raspberry Pi OS (Raspberry Pi Foundation, Cambridge, UK). We executed NTRUEncrypt [9,27] and Falcon signature mechanisms [10,28] with different security levels and recorded the execution time. Results of the performance analysis of the NTRUEncrypt [9,27] and Falcon signature mechanisms [10,28] are shown in Tables 5 and 6, respectively. As a result, the time required to execute NTRUEncrypt [9,27] on a PC is less than 2 s, while in Raspberry Pi 3B, the time required is less than 20 s. Executing the Falcon signature mechanism [10,28] on a PC takes less than 3 s, but executing Falcon-512 in Raspberry Pi 3B takes about 5 times longer than executing Falcon-256 does.

Table 5. Performance analysis of NTRUEncrypt.

Hardware / Security Level	PC (s)	Raspberry Pi 3B (s)
Medium	0.2812	4.405
Standard (80 bits)	0.5312	8.688
High (128 bits)	0.8118	11.345
Highest (160 bits)	1.8266	17.993

Table 6. Performance analysis of Falcon signature mechanism.

Hardware / Security Level	PC (s)	Raspberry Pi 3B (s)
Falcon-64	1.277	20.004
Falcon-128	1.273	20.820
Falcon-256	1.334	39.242
Falcon-512	2.808	211.747

According to the results above, we can estimate the execution time of the proposed scheme with different security levels. We defined $NTRU_{IoT}$ as the time required for the IoT device to execute NTRUEncrypt [9,27], $NTRU_{GW}$ as the time the gateway requires to execute NTRUEncrypt [9,27], and $NTRU_S$ as the time the server requires to execute NTRUEncrypt [9,27]. We also defined Fal_{IoT} as the time the IoT device takes to execute the Falcon signature mechanism [10,28]. We defined Fal_{GW} as the time the gateway takes to execute the Falcon signature mechanism [10,28], and Fal_S as the time the server takes to execute the Falcon signature mechanism [10,28]. Results of the computational complexity and performance time of the proposed scheme are shown in Table 7. In an IoT device's

gateway communication phase, the IoT device will take 24.409 s, and the gateway will take 1.5582 s with a medium level of NTRUEncrypt [9,27] and Falcon-64 [10,28]. If the highest level of NTRUEncrypt [9,27] and Falcon-512 [10,28] is applied in the IoT device's gateway communication phase, the IoT device will take 229.74 s, and the gateway will take 4.6346 s. The execution of the IoT device's gateway communication phase will take 25.9627 to 234.3746 s. In the gateway's server communication phase, the gateway and server will take 1.5582 s separately with a medium level of NTRUEncrypt [9,27] and Falcon-64 [10,28]. If the highest level of NTRUEncrypt [9,27] and Falcon-512 [10,28] is applied in the gateway's server communication phase, the gateway and server will take 4.6346 s separately. The execution of the gateway's server communication phase will take 3.1164 to 9.2692 s. Compared with other research on similar system structures, the proposed scheme takes much more time. For examples, Shang et al.'s scheme takes at least 0.895 ms [29], and Zhang et al.'s scheme takes 0.75 s [30]. However, the proposed scheme achieves encryption, signature, and device authentication at once, so the number of execution rounds is smaller than that in Shang et al.'s [29] and Zhang et al.'s scheme [30] while maintaining security features. Moreover, the proposed scheme applies quantum-resistant cryptographic algorithms: the NTRU [9] and Falcon signature mechanism [10]. A performance analysis of the proposed scheme is shown in Table 7.

Table 7. Performance analysis of proposed scheme.

Role \ Phase	IoT Devices-Gateways Communication Phase	Gateways-Server Communication Phase
IoT Device	$NTRU_{IoT} + Fal_{IoT}$ $= (4.405 + 20.004) \sim (17.993 + 211.747)$ s $= 24.409 \sim 229.74$ s	N/A
Gateway	$NTRU_{GW} + Fal_{GW}$ $= (0.2812 + 1.277) \sim (1.8266 + 2.808)$ s $= 1.5582 \sim 4.6346$ s	$NTRU_{GW} + Fal_{GW}$ $= (0.2812 + 1.277) \sim (1.8266 + 2.808)$ s $= 1.5582 \sim 4.6346$ s
Server	N/A	$NTRU_S + Fal_S$ $= (0.2812 + 1.277) \sim (1.8266 + 2.808)$ s $= 1.5582 \sim 4.6346$ s
Total	$NTRU_{IoT} + Fal_{IoT} + NTRU_{GW} + Fal_{GW}$ $= 25.9672 \sim 234.3746$ s	$NTRU_{GW} + Fal_{GW} + NTRU_S + Fal_S$ $= 3.1164 \sim 9.2692$ s

6. System Implementation

We utilized Raspberry Pi 3B with temperature and humidity sensor module DHT11 as an IoT device, a PC as a gateway, and a server. We used Python as the programing language for gathering data, NTRUEncrypt [9,27], and Falcon signature mechanisms [10,28]. The PC was connected to a smart token that stored private keys of the gateway. We installed VMWare ESXi in the server and executed a virtual machine with Ubuntu Linux as the OS, with 2 GB RAM and a 50 GB hardware capacity, and MySQL phpMyAdmin for storing the information on device registration, the cipher, and the signature. Table 8 presents specifications of the proposed system, and Figure 5 illustrates the system implementation structure of the proposed system.

Table 8. Specifications of proposed system.

Specification \ Devices	IoT Device	Gateway	Server
Module	Raspberry Pi 3B	PC	Server
CPU	ARM Cortex-A53 1.4 GHz 4-core	i7-7000 3.60 GHZ 8-core	E5-2620v3 6-core
RAM	1 GB	32 GB	32 GB
OS	Raspberry Pi	Windows 10 Education	VMWare ESXi-6.7.0

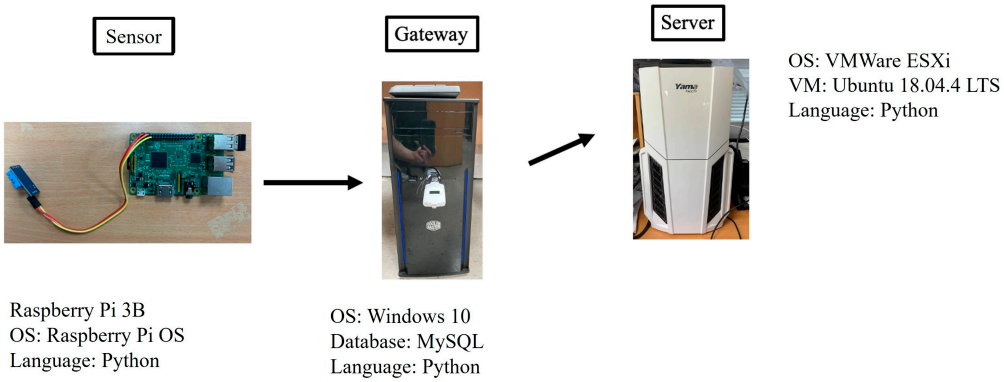

Figure 5. System implementation structure of proposed system.

Raspberry Pi was registered in the server first (Figure 6), and the server stored information on Raspberry Pi in the database utilizing the hostname and MAC address (Figure 7). We can see that Raspberry Pi's information (Figure 6) will appear in server (Figure 7) if Raspberry Pi registered successfully.

```
Please input your UsaLab NTRU Website Username:M0844009
Hostname:raspberrypi
Mac address:B827EB48E0B1
username:M0844009
```

Figure 6. Raspberry Pi's registration request.

id	date	username	name	mac
10	2021-07-08 15:37:01	nike913058	ntruserver-virtual-machine	C29DCE4B5
13	2021-07-09 14:17:42	M0844009	raspberrypi	B827EB48E0B1

Figure 7. Server storing information on Raspberry Pi in database.

After gathering data from DHT11, Raspberry Pi transformed data into binary (Figure 8), encrypted binary data (Figure 9), and signed binary data (Figure 10) using the scheme proposed above, and then Raspberry Pi transmitted data to the gateway through wireless communication.

```
Temp=17*C  humidity=41%
Temperature_Binary:10001 Humidity_binary:101001
NTRU Temperature Message= [1, 0, 0, 0, 1]
NTRU Humidity Message= [1, 0, 1, 0, 0, 1]
```

Figure 8. Raspberry Pi transformed data to binary.

```
Temperate Encrypted Message :  [117, 94, 91, 35, 78, 126, 37, 109, 59, 27, 121, 10, 95, 122, 110, 2
, 77, 60, 58, 39, 96, 73, 92, 113, 17, 52, 80, 86, 126, 117, 125, 7, 22, 94, 49, 111, 126, 123, 71,
117, 21, 47, 18, 49, 30, 14, 6, 55, 68, 50, 125, 120, 71, 36, 31, 35, 80, 44, 114, 98, 119, 87, 81,
22, 86, 91, 53, 102, 53, 5, 59, 43, 41, 98, 127, 26, 70, 114, 69, 28, 2, 111, 0, 89, 124, 113, 121,
92, 56, 54, 46, 109, 37, 7, 94, 6, 57, 7, 86, 11, 47, 117, 101, 127, 26, 17, 94, 86, 86, 127, 36, 4
2, 85, 120, 87, 4, 31, 91, 8, 68, 50, 18, 95, 15, 113, 78, 14, 19, 29, 78, 5, 29, 59, 123, 89, 122,
31, 122, 94, 98, 61, 124, 74, 119, 96, 41, 28, 113, 33, 68, 32, 86, 30, 37, 13, 71, 38, 110, 65, 31
, 46, 91, 23, 117, 117, 79, 98]
Humidity Encrypt Message :  [117, 94, 92, 35, 77, 127, 37, 109, 59, 27, 121, 10, 95, 122, 110, 2, 7
7, 60, 58, 39, 96, 73, 92, 113, 17, 52, 80, 86, 126, 117, 125, 7, 22, 94, 49, 111, 126, 123, 71, 11
7, 21, 47, 18, 49, 30, 14, 6, 55, 68, 50, 125, 120, 71, 36, 31, 35, 80, 44, 114, 98, 119, 87, 81, 2
2, 86, 91, 53, 102, 53, 5, 59, 43, 41, 98, 127, 26, 70, 114, 69, 28, 2, 111, 0, 89, 124, 113, 121,
92, 56, 54, 46, 109, 37, 7, 94, 6, 57, 7, 86, 11, 47, 117, 101, 127, 26, 17, 94, 86, 86, 127, 36, 4
2, 85, 120, 87, 4, 31, 91, 8, 68, 50, 18, 95, 15, 113, 78, 14, 19, 29, 78, 5, 29, 59, 123, 89, 122,
31, 122, 94, 98, 61, 124, 74, 119, 96, 41, 28, 113, 33, 68, 32, 86, 30, 37, 13, 71, 38, 110, 65, 31
, 46, 91, 23, 117, 117, 79, 98]
```

Figure 9. Raspberry Pi encrypted data.

```
>>> sig=sk.sign(b"17*C")
>>> sig
b'7\x13\x84\xfb\x82?\xeb\x0e\xaf\xf5\xb2\xc1\xd6\x8a\xb6=\xd3t\xde\x92|0#\xbbG\x
1ax\xb0\xf3+\xf9\xfe+\x9f\n\xffp\x08\xfc\xa7e\xe0\xd54\x83\xc3\xf3\xc7*t\xb6\x88
\xea=\x9cT)&\x7f\x12E\xc1r|p\xed\xcc\x16F\xb2@w\xf5\x1cg=\xb6\xfd\xa6\xad\xa7!@\
xf5{\xa3\xf4\xa5c\x92\xe2i\xa9\xfe\xc3_\xa7z\xfd\\\x19\x0e}\xfc\xf5u\x0c-\xd2\xa
1\x1d\xb3\x10(\x0fJ\x9c\x89$\x93e9\x1e\x7fR\x95\x0f\x18}q\xa9\xfc7W \xd8\x9d\x8c
\x14U\xfcm0\xe2P\xcd$U\xe4\xcd\xe4\xd9r\xa6.\x86m\x19|\x0c\x1b\xd2\xcfhsjN1v\xc6
4D\xda\x8c\xd0\xd45o\xeeeB<6L)\xe2\xaf\xdb\xdc\xccj\x8c\xe7\xed\x10\x00\x00\x00\
x00\x00'
>>> sig=sk.sign(b"41%")
>>> sig
b'7je\xbe\x9cv\x9e\xea\xb7\xec\x9f>\x82"\xcdZ@y\xc1\x89K\xc43!\xbc-\xa4=\xe6\x13
\x8a\x96\x89\xbf\xd5f\x16y\x85\x98\xf8\x1e\x8c*\xeeK\x7f\x7f`\xcdd\xc2B\xd9\xf8\
xed\xd0\xd4_5\rV:\xca\xec\x0e\r)GrM\x0bQ@\x8e\xa8\xf4\xa2\x1d*\xb0\xd9\x98\xf2\x
be\xde\nS\x8b\xeb\x84\xb9K\x05bEk\xc1 e\x9a\xc1\xc8W\xab\xbd\xb7\x9a\x85\x02\x85
\x0c\x8b<\xd8\x1f\xf8\xa5\x12P\x86_/f)\x19\xef\'lA\x08\xf2%I\x96\xa6\x8f5S\x178\
xdao\xd4\x94\xc2D\xcf\x15\xf1Xu\xfc\x87S\xbfQ<*U\xa0\xa4G\xe1\xdfl\x9b\xc8fj\xb4
\xdd*\xc0\x8f\xa5;\xa6Z\xf4\xe7!\xceJ1\xabB\xb91R\x18\x9a\x91\x86a\x00\x00\x00\x
00\x00'
```

Figure 10. Raspberry Pi signed data.

After receiving data, as shown in Figures 11 and 12, the gateway verified the signature and decrypted data using the private key in the smart token, as shown in Figure 13. As a result, the gateway verified the signature successfully and showed that the signature was correct in Figure 14; the gateway also decrypted data successfully, as shown in Figure 15, which shows the same results as those in Figure 8.

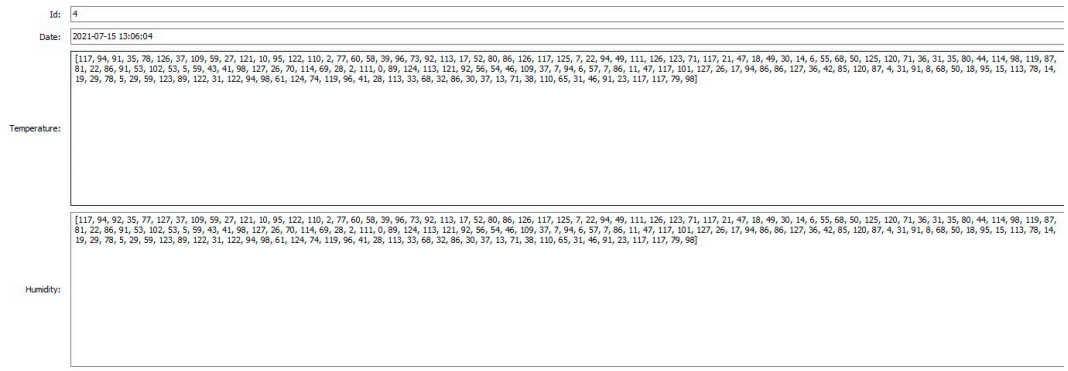

Figure 11. Gateway receiving encrypted data.

Figure 12. Gateway receiving signed data.

```
Data container: 0xc900
Label: NTRU
Appinfo: Private
Oid: Key
  NTRU   ?  0    Private  Key?0
Data type: Private
Value: 30 31 30 31 30 30 31 31 31 31 30 31
```

Figure 13. Private key in smart token.

```
>>> pk.verify(b"17*C", sig)
True
>>> pk.verify(b"41%", sig)
True
```

Figure 14. Gateway verifying signature.

```
b'30 31 30 31 30 30 31 31 31 31 30 31'
repalce::: 303130313030313131313031
[1, 1, 0, -1, -1, 1]

--------------------------------------------------

Temperate Decrypted Message :  [1, 0, 0, 0, 1]
Humidity Decrypted Message :  [1, 0, 1, 0, 0, 1]
```

Figure 15. Gateway decrypting data.

The gateway encrypted and signed the data using the private key in the smart token, as shown in Figures 16 and 17, and the gateway sent the data to the server, as shown in Figures 18 and 19.

```
Temperature Original Message:  [1, 0, 0, 0, 1]
Humidity Original Message:  [1, 0, 1, 0, 0, 1]
Temperate Encrypted Message :  [91, 0, 1, 57, 44, 127, 56, 40, 76, 64, 33, 27, 73, 76, 48, 19, 10
3, 76, 43, 31, 1, 15, 111, 6, 107, 18, 84, 46, 73, 50, 62, 101, 104, 63, 50, 93, 101, 36, 44, 100
, 72, 35, 73, 55, 80, 84, 9, 53, 56, 85, 21, 79, 9, 41, 102, 93, 6, 0, 26, 99, 70, 78, 47, 96, 81
, 102, 91, 39, 48, 64, 92, 32, 41, 67, 121, 116, 88, 59, 111, 100, 59, 23, 1, 47, 63, 30, 123, 11
4, 60, 30, 49, 2, 102, 45, 104, 79, 66, 109, 29, 12, 84, 116, 72, 123, 65, 39, 40, 76, 33, 109, 9
6, 101, 93, 79, 9, 89, 110, 13, 6, 56, 106, 59, 118, 38, 71, 0, 97, 22, 11, 15, 104, 88, 44, 64,
113, 43, 105, 28, 64, 99, 119, 124, 11, 15, 1, 15, 15, 118, 11, 18, 100, 14, 25, 82, 14, 53, 40,
31, 82, 125, 21, 52, 124, 68, 8, 19, 121]
Humidity Encrypt Message :  [91, 0, 2, 57, 43, 0, 56, 40, 76, 64, 33, 27, 73, 76, 48, 19, 103, 76
, 43, 31, 1, 15, 111, 6, 107, 18, 84, 46, 73, 50, 62, 101, 104, 63, 50, 93, 101, 36, 44, 100, 72,
35, 73, 55, 80, 84, 9, 53, 56, 85, 21, 79, 9, 41, 102, 93, 6, 0, 26, 99, 70, 78, 47, 96, 81, 102,
91, 39, 48, 64, 92, 32, 41, 67, 121, 116, 88, 59, 111, 100, 59, 23, 1, 47, 63, 30, 123, 114, 60,
30, 49, 2, 102, 45, 104, 79, 66, 109, 29, 12, 84, 116, 72, 123, 65, 39, 40, 76, 33, 109, 96, 101,
93, 79, 9, 89, 110, 13, 6, 56, 106, 59, 118, 38, 71, 0, 97, 22, 11, 15, 104, 88, 44, 64, 113, 43,
105, 28, 64, 99, 119, 124, 11, 15, 1, 15, 15, 118, 11, 18, 100, 14, 25, 82, 14, 53, 40, 31, 82, 1
25, 21, 52, 124, 68, 8, 19, 121]
```

Figure 16. Gateway encrypting data.

```
>>> sig = sk.sign(b"17*C")
>>> sig
b"7']u\xf1\x18\xdb\xb9\xbaZ\x94\xd3\xc8\xd3\x8f\x98\x11\xff\x183E\x06\x9d\xa2\x96\xfco\x
8c\x83\xacy\xc9\xb9\xfb\xbdy\xfb\x98\x94\x0e\xc6Z\xe20G\xf6\xe5\xcfw:\xdd[:\xa3\xce\xbc\
xe4T\xc9\xda\xb8F\x99\xf8\xc5\xeb?7\x93\x18\x1d\x94j\x1f%:\x1a.\xbc\xb2\x1b&9\x90m4z\xa6
P2\xc8]\xdfv\xbfa\xa5\x04v-\x9a\x92WV\xb8=\x17O\x88\x82\xff\xe9\xbe\xd9{\x11\x82~\xf4\x1
1\\\x08\xf9\xbd\xce\xd1\x9c\x8e\xd5(\xdfy\xd4\x03D\xacM\xe9\x13J\xc7\xe2\xd5d\x96\xa91,{
\xbaR\x9e\x169R\x99\xe0\xde\xbe\xd4%\x10\xc9hO\xa6a\xdc[\xf2\xdb$\x08\xac_\xf9\x9az\x8a\
xb3\xc0\xcc\xdaa\xf8\xb9U\xe3\x7fJ\xff@\x00\x00\x00\x00\x00\x00\x00"
>>> sig=sk.sign(b"41%")
>>> sig
b'7\x1a\x1fn\rrX*yC\x8aO/\xa7\x0f\xdd\x89\x04\xea\x07\xc5\'\xd7[\xed\x1d\x91\xe4\xd2\x0b
\x174B\x00\x1b~H\x9f\xa1\x1f\xe8\x85}m\xb8L\xc7\xbeNc\x1d\xfd1\x9aHa)Q:\xa7\xee\xce~!\x8
6\x18\xacZ\x04\xebo\xf6\xfb\xb1;\xb5+\xac\x1e\xd0\x80\xb7o\xe3Ds"\xbe\xf5\xd7\xcbz\xe9E-
Pc\xbb~W\x10\xa9CP\x8e9\xaag\x8d\x86bY\x9bu1\x83\xc5\xa3i\xd4\x01\xa7\xe17\x84\xf2p\xcb\
xf1wy\xbb.\x91\xfa\xff%<\xce\xe1\x8f(\x12Uz\xa7\xeaQ\xe4\x8bU\x82\x0fY\x83$m$\xc2 \xa4$\
x89F\xa9U\x93\x92\x127C\xaa\xed,m\x96e<\xcb\x1c\xd5"\xc7e\xd1\x88\xe2\x87z\x8eH\xbdJ\xbc
J\x00\x00\x00\x00\x00'
```

Figure 17. Gateway signing data.

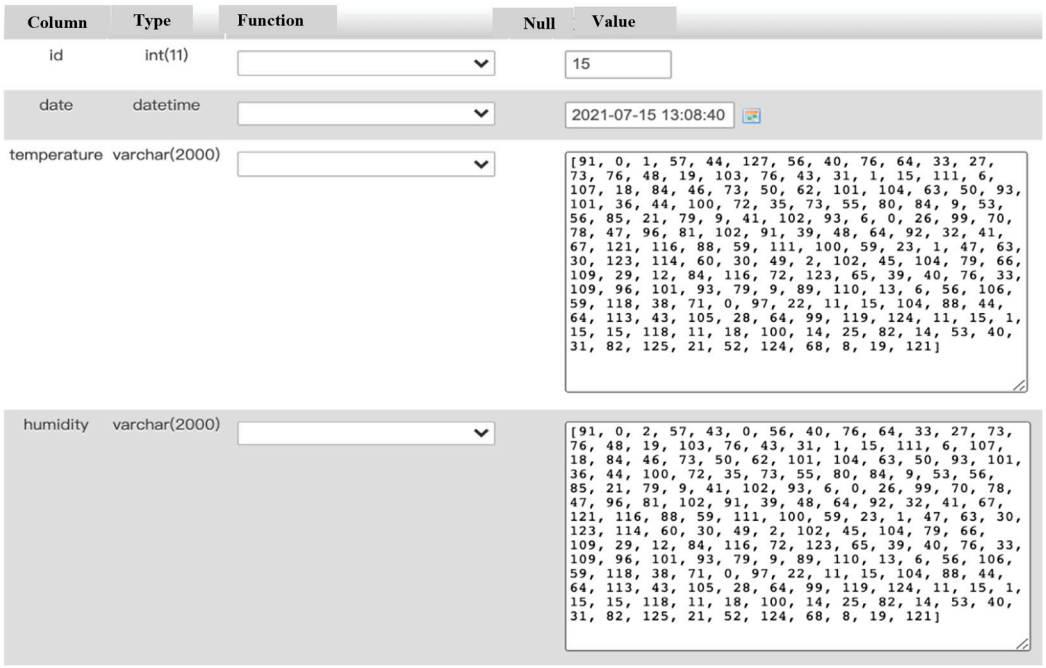

Figure 18. Encrypted data being stored in server.

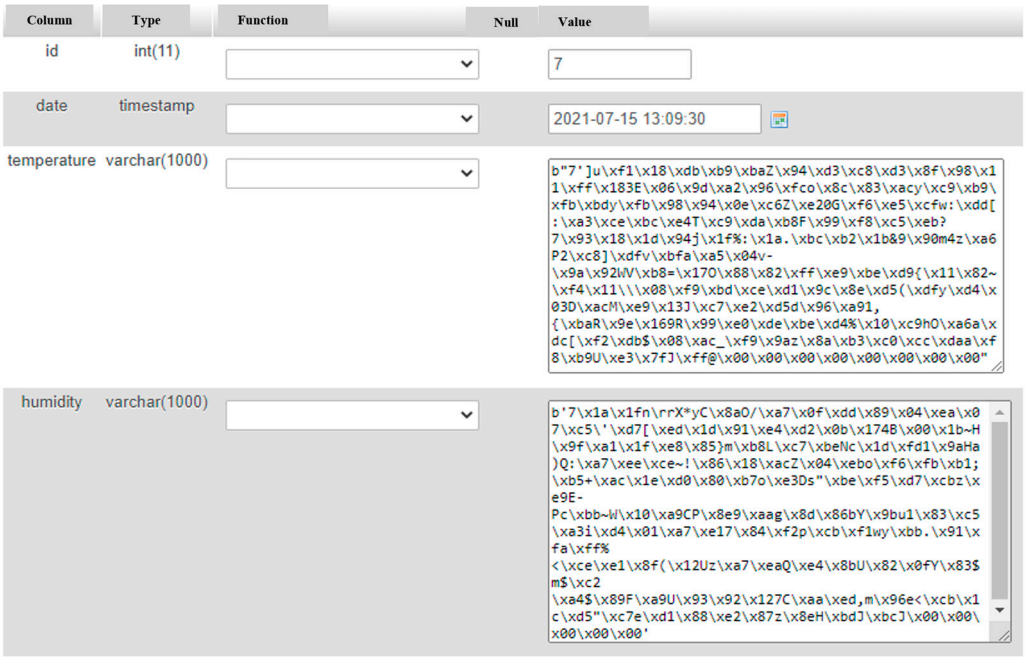

Figure 19. Signed data being stored in server.

7. Results and Discussion

We present a discussion on the research results and limitations of this research.

We proposed a device authentication and secure communication system for AIoT environments using NTRU [9] and Falcon signature mechanisms [10]. NTRU and Falcon signature mechanisms have been proven to be quantum-resistant public key cryptosystems [9,10,20]. Although the NTRU cannot be used in post-quantum cryptography standardization, it is still applied widely because of its advantages over other lattice-based schemes [9,20,21]. Moreover, the Falcon signature mechanism [10] is a lattice-based compact signature with advantages over the NTRU, so the NTRU [9] and Falcon signature mechanisms [10] can be integrated. Because of reasons above, we could design a scheme

that allows the execution of encryption, authentication, and signature mechanisms at once. However, since the NTRU [9] fell short among the candidates of post-quantum cryptography standardization, other lattice-based schemes can be discussed in the future, such as CRYTSTAL-Kyber [31] or CRYSTALS-Dilithium digital signature algorithms [32].

This research has limitations. Although the proposed scheme was able to achieve security features, such as quantum resistance, it did not seem ideal in terms of execution time. The results will limit the implementation possibilities of AIoT environments. We take smart medical or telemedicine systems with AIoT as examples. Biodata are measured by wearable devices and transmitted through gateways to a server, and the server can analyze and predict each patient's health condition or possible disease using AI algorithms. Biodata may not be transmitted every second, so the proposed scheme can be applied in this scenario. If data have to be transmitted every second, the proposed scheme is not suitable unless a redeployment of end devices better than that provided by the Raspberry Pi 3B module is carried out in the system implementation of the proposed scheme. With the rapid development of PQC, hardware circuit and system design has been discussed. For example, Xie et al. proposed a tutorial for PQC and introduced related techniques [33].

8. Conclusions

The ability to collect data in real time through sensors and analyze data using machine learning algorithms is enabling AIoT to expand its possibilities and applications into many sectors. AIoT creates value for sustainable industries by combining artificial intelligence, the Internet of Things, and big data analysis. Nevertheless, AIoT requires security measures suitable for the environment in which it operates because of its reliance on networks that are exposed to adversaries from any internet location. Due to the limited resources of devices in AIoT systems, which can lead to a loss of attractive features, traditional security measures may not be an appropriate solution. We designed and evaluated a device authentication and secure communication system with PQC for AIoT environments that utilized PQC to provide a lightweight security scheme while resisting attacks from quantum computers. We chose NTRU and Falcon signature mechanisms to design and implement secure algorithms. We analyzed the security and performance of the proposed scheme and proved that it can achieve confidentiality, integrity, and non-repudiation meanwhile also achieving efficiency.

Author Contributions: Conceptualization, Y.-J.C., C.-L.H. and T.-W.L.; methodology, Y.-J.C. and C.-L.H.; software, Y.-J.C.; validation, Y.-J.C. and T.-W.L.; formal analysis, Y.-J.C.; writing—original draft preparation, Y.-J.C.; writing—review and editing, T.-W.L., C.-L.H. and J.-S.L.; visualization, Y.-J.C. and T.-W.L.; supervision, C.-L.H. and J.-S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Murphy, K.; Di Ruggiero, E.; Upshur, R.; Willison, D.J.; Malhotra, N.; Cai, J.C.; Malhotra, N.; Lui, V.; Gibson, J. Artificial intelligence for good health: A scoping review of the ethics literature. *BMC Med. Ethics* **2021**, *22*, 14. [CrossRef] [PubMed]
2. Lin, T.-W.; Hsu, C.-L. Privacy-Preserved Hierarchical Authentication and Key Agreement for AI-Enabled Telemedicine Systems. In Proceedings of the 2021 International Conference on Security and Information Technologies with AI, Internet Computing and Big-Data Applications, Taichung City, Taiwan, 18–20 November 2021; pp. 134–142.
3. Li, S.; Xu, L.D.; Zhao, S. 5G Internet of Things: A survey. *J. Ind. Inf. Integr.* **2018**, *10*, 1–9. [CrossRef]
4. Wong, A.M.; Hsu, C.-L.; Le, T.-V.; Hsieh, M.-C.; Lin, T.-W. Three-Factor Fast Authentication Scheme with Time Bound and User Anonymity for Multi-Server E-Health Systems in 5G-Based Wireless Sensor Networks. *Sensors* **2020**, *20*, 2511. [CrossRef] [PubMed]
5. Cheng, S.M.; Hong, B.K.; Hung, C.F. Attack Detection and Mitigation in MEC-Enabled 5G Networks for AIoT. *IEEE Internet Things Mag.* **2022**, *5*, 76–81. [CrossRef]
6. Nozari, H.; Szmelter-Jarosz, A.; Ghahremani-Nahr, J. Analysis of the Challenges of Artificial Intelligence of Things (AIoT) for the Smart Supply Chain (Case Study: FMCG Industries). *Sensors* **2022**, *22*, 2931. [CrossRef] [PubMed]

7. Dyakonov, M. When will useful quantum computers be constructed? Not in the foreseeable future, this physicist argues. Here's why: The case against: Quantum computing. *IEEE Spectr.* **2019**, *56*, 24–29. [CrossRef]
8. Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509. [CrossRef]
9. Hoffstein, J.; Pipher, J.; Silverman, J.H. NTRU: A ring-based public key cryptosystem. In Proceedings of the Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, OR, USA, 21–25 June 1998; pp. 267–288.
10. Fouque, P.-A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Prest, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU. Available online: https://falcon-sign.info/falcon.pdf (accessed on 22 December 2023).
11. Lei, X.; Liao, X. NTRU-KE: A Lattice-based Public Key Exchange Protocol. *IACR Cryptol. ePrint Arch.* **2013**, *2013*, 718.
12. Rong, G.; Xu, Y.; Tong, X.; Fan, H. An edge-cloud collaborative computing platform for building AIoT applications efficiently. *J. Cloud Comput.* **2021**, *10*, 36. [CrossRef]
13. Ricquebourg, V.; Menga, D.; Durand, D.; Marhic, B.; Delahoche, L.; Loge, C. The Smart Home Concept: Our immediate future. In Proceedings of the 2006 1ST IEEE International Conference on e-Learning in Industrial Electronics, Hammamet, Tunisia, 18–20 December 2006; pp. 23–28.
14. Lucke, D.; Constantinescu, C.; Westkämper, E. Smart Factory—A Step towards the Next Generation of Manufacturing. In Proceedings of the Manufacturing Systems and Technologies for the New Frontier: The 41st CIRP Conference on Manufacturing Systems, Tokyo, Japan, 26–28 May 2008; pp. 115–118.
15. Schaffers, H.; Komninos, N.; Pallot, M.; Trousse, B.; Nilsson, M.; Oliveira, A. Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation. In *The Future Internet: Future Internet Assembly 2011: Achievements and Technological Promises*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 431–446.
16. Fernández-Caramés, T.M.; Fraga-Lamas, P. A Review on the Use of Blockchain for the Internet of Things. *IEEE Access* **2018**, *6*, 32979–33001. [CrossRef]
17. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575. [CrossRef]
18. Dai, H.N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A Survey. *IEEE Internet Things J.* **2019**, *6*, 8076–8094. [CrossRef]
19. Ray, P.P.; Dash, D.; De, D. Edge computing for Internet of Things: A survey, e-healthcare case study and future direction. *J. Netw. Comput. Appl.* **2019**, *140*, 1–22. [CrossRef]
20. Post-Quantum Cryptography. Available online: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography (accessed on 18 March 2024).
21. Kim, J.; Park, J.H. NTRU+: Compact Construction of NTRU Using Simple Encoding Method. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 4760–4774. [CrossRef]
22. Perlner, R.A.; Cooper, D.A. Quantum resistant public key cryptography: A survey. In Proceedings of the 8th Symposium on Identity and Trust on the Internet, Gaithersburg, MD, USA, 14–16 April 2009; pp. 85–93.
23. Mailloux, L.O.; Lewis, C.D., II; Riggs, C.; Grimaila, M.R. Post-Quantum Cryptography: What Advancements in Quantum Computing Mean for IT Professionals. *IT Prof.* **2016**, *18*, 42–47. [CrossRef]
24. Bi, J.; Han, L. Lattice Attacks on NTRU Revisited. *IEEE Access* **2021**, *9*, 66218–66222. [CrossRef]
25. Ahmed Othman, K.; Shaimaa Khudhair, S.; Hind Jumaa, S.; Zainab Khyioon, A. Subject Review: Comparison between RSA, ECC & NTRU Algorithms. *Int. J. Eng. Res. Adv. Technol.* **2019**, *5*, 11–15. [CrossRef]
26. Loriya, H.T.; Kulshreshta, A.; Keraliya, D.R. Security analysis of various public key cryptosystems for authentication and key agreement in wireless communication network. *Int. J. Adv. Res. Comput. Commun. Eng.* **2017**, *6*, 267–274.
27. Singh, G. NTRU-Python3. Available online: https://github.com/topShotZexN/NTRU-Python3 (accessed on 22 December 2023).
28. Prest, T. falcon.py. Available online: https://github.com/tprest/falcon.py (accessed on 22 December 2023).
29. Shang, Z.; Ma, M.; Li, X. A Secure Group-Oriented Device-to-Device Authentication Protocol for 5G Wireless Networks. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 7021–7032. [CrossRef]
30. Zhang, Y.; Li, B.; Wu, J.; Liu, B.; Chen, R.; Chang, J. Efficient and Privacy-Preserving Blockchain-Based Multifactor Device Authentication Protocol for Cross-Domain IIoT. *IEEE Internet Things J.* **2022**, *9*, 22501–22515. [CrossRef]
31. Schwabe, P.; Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Seiler, G.; Stehle, D. CRYSTALS-Kyber Algorithm Specifications and Supporting Documentation (Version 3.02). 2021. Available online: https://www.pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf (accessed on 1 February 2024).
32. Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2018**, *2018*, 238–268. [CrossRef]
33. Xie, J.; Zhao, W.; Lee, H.; Roy, D.B.; Zhang, X. Hardware Circuits and Systems Design for Post-Quantum Cryptography—A Tutorial Brief. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 1670–1676. [CrossRef]