Cryptanalysis and Countermeasure of “An Efficient NTRU-based Authentication Protocol in IoT Environment”

YoHan Park1, Woojin Seok2, Wonhyuk Lee3, Hong Taek Ju4
Department of Computer Engineering
Keimyung University, Korea, Republic1,4
KREONET Center, KISTI, Korea, Republic2,3

Abstract—A quantum computer is a paradigm of information processing that can show remarkable possibilities of exponentially improved information processing. However, this paradigm could disrupt the current cryptosystem, which is called quantum computing attacks, by calculating factoring problem and discrete logarithm problem. Recently, NTRU is applied to various security systems, because it provides security against to provide security against quantum computing attacks. Furthermore, NTRU provides similar security level and efficient computation time of encryption/decryption compared to traditional PKC. In 2018, Jeong et al. proposed an user authentication and key distribution scheme using NTRU. They claimed that their scheme provides various security properties and secure against quantum computing attacks. In this paper, we demonstrate that their scheme has security pitfalls and incorrectness in login and authentication phase. We also suggest countermeasures to fix the incorrectness and provide security against various attacks.

Keywords—Post-quantum; NTRU; biometrics; user authentication; key agreement

I. INTRODUCTION

Development of Internet of Things (IoT) technology help users connect service providers easily and fast and utilize various services such as Health care, SmartHome, SmartGrid, and so on. However, IoT environments have threats to security and privacy because of its wireless nature [1]. Such threats hinder users make use of beneficial applications and service providers may not continue to invest for profits. Security problems should get solved to make the IoT-based services widely spread and applied.

User authentication and key agreement are essential requirement among all other security concerns. Those security services provide integrity and confidentiality for IoT environments [2]–[4]. Malicious adversaries will freely access user’s critical and valuable information if secure authentication and key agreement methods do not provide.

Security of public key cryptosystem (PKC) are mostly based on the difficulty of factorization problem (FP) or discrete logarithm problem (DLP). RSA and elliptic curve cryptosystem (ECC) are major the examples of current cryptosystems based on FP and DLP. However, these algorithms are vulnerable to a quantum computing attack. In 1994, Peter W. Shor [5] proposed a quantum computing algorithm which can solve FP efficiently. And a quantum search algorithm proposed by Grover [6] can easily solve DLP. These algorithms based on quantum computing became major threats to all security protocols using RSA or DLP. Therefore, ETSI [7] and NIST [8] recommended that post-quantum cryptosystem (PQC) should be prepared with high priority.

There are several PQC which are secure against quantum computing attacks. These cryptosystems use Code, Lattice, Hash and Multivariate to provide security in quantum computing environments. Among many algorithms of PQC, NTRU, proposed by J. Hoffstein [9] in 1996, has been approved for standardization by Institute of Electrical and Electronics Engineers (IEEE) [10]. The security of NTRU is based on the difficulty on a finding shortest path in n-th degree Lattice. Comparing to traditional PKC, NTRU provides not only similar security level, but also efficient computation time of encryption/decryption. Recently, NTRU is applied to various security systems which provides user authentication and key agreement.

Recently, a number of authentication and key agreement schemes have been proposed in IoT environments [11], [12], [19]. In 2017, Li et al. [13] proposed a key distribution protocol using ECC in IoT. However, the scheme is vulnerable to quantum computing attacks, such as Shor [5] or Grover [6] algorithm. To overcome these security pitfalls, Jeong et al. proposed an efficient NTRU-based authentication protocol in IoT environments [14] in 2018. They proposed user authentication and key agreement protocols using NTRU and claimed their scheme is secure against quantum computing attacks and prevents impersonations attack and session key disclosure attack. However, we find out that their scheme does not provide a proper user authentication process and is weak to various attacks, such as privileged insider attacks, impersonation attacks, and session key disclosure attacks. In addition, we show their scheme fails to provide correctness in login phase and authentication phase.

In this paper, we show the security weaknesses of Jeong et al.’s scheme. In addition, we propose countermeasures for the weaknesses of Jeong et al.’s scheme.

A. Contributions

The contributions made in the paper are listed below:
1) We demonstrate that Jeong et al.’s scheme has an incorrectness in login phase and authentication phase.
2) We analyze security weaknesses of Jeong et al.’s scheme and show that their scheme is vulnerable to privilege insider attacks, impersonation attacks, and session key disclosure attacks.
3) We propose countermeasures to overcome the security weaknesses of Jeong et al.’s scheme. The countermeasures help to prevent various attacks such as password guessing attacks, user impersonation attacks and session key disclosure attacks from malicious adversaries.

B. Paper Structure

The rest of the paper is organized as follows. In Section II, we introduce preliminaries used in this paper. In Section III, we review Jeong et al.’s scheme followed by the cryptanalysis of Jeong et al.’s scheme in Section IV. In Section V, we propose countermeasures for the weaknesses of Jeong et al.’s scheme. Finally, Section VI concludes the paper.

II. Preliminaries

A. NTRU

NTRU is a lattice-based public key cryptosystem proposed by Jeffry Hoffstein et al. [9]. This provides a similar security level, but high performance compared to RSA and ECC because of low computational complexity of polynomial convolution operation. NTRU requires \(O(n^2)\) operations to encrypt or decrypt a message of size \(n\), but RSA and ECC require \(O(n^3)\) operations. NTRU, furthermore, resists quantum computing attacks and has adopted standard as IEEE 1363.1 and X9.98. NTRU cryptosystem consists of three parts: key generation, encryption and decryption.

1) Key generation: Alice and Bob are required to generate private/public key in advance to exchange data securely in PKC. The detail steps of key generation are as follows:

**Step 1:** Alice chooses two polynomials \(f\) and \(g\) with degree \(N-1\) and coefficients in \([-1,0,1]\).

**Step 2:** The polynomial \(f \in L_f\) must have the inverse element for modulo \(p\) and \(q\). Alice computes \(f * f^{-1}_p \equiv 1 \pmod{p}\) and \(f * f^{-1}_q \equiv 1 \pmod{q}\).

**Step 3:** If \(f\) does not have an inverse element, Alice turns back to Step 1 and chooses another \(f\). Otherwise, Alice computes the public key \(h = pf^{-1}_g + g \pmod{q}\).

\(f\) and \(g\) are private keys and \(h\) is a public key of Alice.

2) Encryption: If Bob wants to send a message to Alice securely, Bob performs the encryption as follows:

**Step 1:** Bob who wants to send a plaintext polynomial \(m \in L_m\) chooses a random polynomial \(r \in L_r\), with \(N-1\) degree and small coefficients. Coefficients are not restricted to the set \([-1,0,1]\).

**Step 2:** Bob encrypts the message \(m\) into \(e\) using the public key \(h\) of Alice. \(e = r * h + m\)

**Step 3:** Bob sends the encrypted message \(e\) to Alice.

3) Decryption: After receiving \(e\) from Bob, Alice decrypts the message as follows:

**Step 1:** Alice calculates a convolution \(a = e * f \pmod{q}\), where \(f\) is a private key of Alice. The coefficient of \(a\) should satisfy \(A \leq a_i \leq A + q\).

**Step 2:** Alice retrieves \(m \equiv a \pmod{q}\).

B. Notations

Table I describes the notations used throughout the paper.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(U_A)</td>
<td>user A</td>
</tr>
<tr>
<td>(ID_A)</td>
<td>identity of (U_A)</td>
</tr>
<tr>
<td>(PW_A)</td>
<td>password of (U_A)</td>
</tr>
<tr>
<td>(RPW_A)</td>
<td>pseudo password of (U_A)</td>
</tr>
<tr>
<td>(B_A)</td>
<td>biometric template of (U_A)</td>
</tr>
<tr>
<td>(SC_A)</td>
<td>smart card of user (U_A)</td>
</tr>
<tr>
<td>(GWN)</td>
<td>gateway node</td>
</tr>
<tr>
<td>(\ast)</td>
<td>convolution computation</td>
</tr>
<tr>
<td>(f, g)</td>
<td>private key polynomial (f \in L_f, g \in L_g)</td>
</tr>
<tr>
<td>(f^{-1}_p, f^{-1}_q)</td>
<td>inverse polynomial of (f)</td>
</tr>
<tr>
<td>(h)</td>
<td>public key</td>
</tr>
<tr>
<td>(H)</td>
<td>hash function</td>
</tr>
<tr>
<td>(|)</td>
<td>concatenate operation</td>
</tr>
<tr>
<td>(\oplus)</td>
<td>XOR operation</td>
</tr>
</tbody>
</table>

III. REVIEW OF JEONG ET AL.’S SCHEME

In this section, we review Jeong et al.’s NTRU-based authentication scheme. The scheme is composed of three phases: user registration phase, things registration phase, and login-authentication-key distribution phase.

A. User Registration Phase

In this phase, a user registers his/her information to the gateway node, and acquires a personalized smart card \(SC_A\). The Jeong et al.’s user registration phase is illustrated in Fig. 1, and the detailed steps of this registration phase are as follows:

**Step 1:** A user \(U_A\) chooses \(ID_A\) and \(PW_A\), then generates a random number \(x_A\). Then, \(U_A\) selects polynomials \(f_A \in L_f\) and \(g_A \in L_g\), then calculates inverse elements \(f^{-1}_A\) and \(f^{-1}_B\) of \(f\). Next \(U_A\) calculates the public key \(h_A = pf^{-1}_A + g_A \pmod{q}\) and the pseudo password \(RPW_A = H(PW_A|x_A). \ U_A\ sends the registration request message \(\{ID_A, RPW_A, h_A\}\) to the gateway node via a secure channel.

**Step 2:** After receiving registration request message from the user, the gateway node \(GWN\) stores the pair \(\{ID_A, h_A\}\) in database. \(GWN\) also selects polynomials \(f_B \in L_f\) and \(g_B \in L_g\), then calculates inverse elements \(f^{-1}_B\) and \(f^{-1}_D\) of \(f\). Next \(GWN\) calculates the public key \(h_B = pf^{-1}_B + g_B \pmod{q}\). Next, \(GWN\) computes \(H(ID_A||RPW_A||h_A)\), and issues a smart card \(SC_A\) with \(H(ID_A||RPW_A||h_A)\) and sends \(\{SC_A, h_B\}\) to \(U_A\).

**Step 3:** After receiving \(\{SC_A, h_B\}\) from the \(GWN\), \(U_A\) computes \(V_A = H(ID_A||RPW_A||H(B_A))\), where \(B_A\)
**B. Things Registration Phase**

In this phase, a thing registers its information to the gateway node \( GWN \), and receives an ephemeral key \( z_S \) from \( GWN \). The Jeong et al.’s things registration phase is illustrated in Fig. 2, and the detailed steps of this registration phase are as follows:

**Step 1:** A thing chooses a random number \( n_S \), then sends it to \( GWN \) through the secure channel.

**Step 2:** After receiving \( n_S \) from the thing, \( GWN \) choose a random number \( n_Z \). Then \( GWN \) stores the pair \((n_S, n_Z)\) in the database. Finally, \( GWN \) sends \( z_S \) to the thing via secure channel.

**Step 3:** After receiving \( n_Z \) from \( GWN \), the thing stores \( n_Z \) in its database.

**C. Login-Authentication-Key Distribution Phase**

In this phase, a user uses his/her multi-factor keys to login and authenticate oneself with \( GWN \). Then a user shares session key \( SK \) with a thing. The Jeong et al.’s login-authentication-key distribution phase is illustrated in Fig. 3, and the detailed steps of this registration phase are as follows:

**Step 1:** A user inputs \( ID_A, PW_A \), and imprints the biometrics \( B_A \) into the smart card \( SC_A \). Then, \( SC_A \) computes \( RPW_A, V_A \), and verifies the validity of the user as follows:

\[
\begin{align*}
RPW_A &= H(PW_A| x_A) \\
V_A &= H(ID_A||RPW_A||H(B_A))
\end{align*}
\]

If it is wrong, \( SC_A \) quits the login process. Otherwise, \( SC_A \) chooses random numbers \( r_A, k_A \) and computes \( I_A, e_A \) as follows:

\[
\begin{align*}
I_A &= H(ID_A||RPW_A) \\
e_A &= pr_A * h_B + k_A
\end{align*}
\]

Then, the user sends \( \{I_A, e_A, h_A\} \) to \( GWN \).

**Step 2:** After receiving \( \{I_A, e_A, h_A\} \), \( GWN \) verifies \( I_A \) using the stored pair \( \{ID_A, h_A\} \). Then, \( GWN \) retrieves \( k_A \), and computes \( c_B \) as follows:

\[
\begin{align*}
I_A &= H(ID_A||RPW_A) \\
a_B &= f_B * e_A (mod q) \\
k_A &= f_B^{-1} * a_A (mod p) \\
c_B &= z_S + k_A
\end{align*}
\]

Then, \( GWN \) sends \( c_B \) to the thing.

**Step 3:** After receiving \( c_B \) from \( GWN \), the thing retrieves \( k_A = c_B \oplus z_S \). Then the thing chooses a random number \( k_S \) and computes a session key \( SK = H(k_A||k_S||n_S) \). The thing computes \( c_S = k_S \oplus z_S \) and sends it to \( GWN \).

**Step 4:** After receiving \( c_S, GWN \) chooses a random number \( r_B \in L_r \) and computes \( k_S \) and \( e_B \) as follows:

\[
\begin{align*}
k_S &= c_S \oplus z_S \\
e_B &= pr_B * h_A + (k_S||n_S) (mod q)
\end{align*}
\]
chooses a random number \( n_S \)

\[
\{n_S\} \quad \xrightarrow{\text{(secure channel)}} \quad \text{chooses a random number } z_S
\]

stores a pair \( \{z_S, n_S\} \) in the database

store \( z_S \) in the database

Fig. 2. Thing registration phase of Jeong et al.'s scheme

<table>
<thead>
<tr>
<th>User ((U_A))</th>
<th>Gateway Node ((GWN))</th>
<th>Thing</th>
</tr>
</thead>
<tbody>
<tr>
<td>inputs (ID_A, PW_A), imprints (B_A), computes (RPW_A = H(PW_A</td>
<td></td>
<td>x_A)), verifies (V_A = H(ID_A</td>
</tr>
<tr>
<td>( I_A )</td>
<td>( e_B )</td>
<td>( {e_B} \xrightarrow{\text{(cA)}} ) computes ( a_A = f_A * e_S \Mod{q} ), ( {k_S</td>
</tr>
</tbody>
</table>

Fig. 3. Login-Authentication-Key Distribution Phase of Jeong et al.’s Scheme

Then, \( GWN \) sends \( e_B \) to the user.

**Step 5:** After receiving \( e_B \), the user computes \( k_S || n_S \) and the session key \( SK \) as follows:

\[
\begin{align*}
    a_A &= f_A * e_B \\
    k_S || n_S &= f_A^{-1} * a_A \\
    SK &= H(k_A || k_S || n_S)
\end{align*}
\]
IV. CRYPTANALYSIS OF JEONG ET AL.’S SCHEME

In this section, we demonstrate the security flaws of Jeong et al.’s scheme. Their scheme does not provide correctness at the login phase and the key distribution phase. Thus, the user who tries to connect an IoT device cannot login to GWN and share a session key. Jeong et al. claimed that their scheme is secure against privileged insider attacks. However, we cryptanalyze and show that their scheme is vulnerable to the attacks. A malicious inside adversary can access to the database and compute user’s information, then guess identities of users. Using the information, the adversary can act as a legal user as follows:

1) An insider adversary $A$ can get the values $\{ID_A, h_A\}$ stored in the database and $\{H(ID_A||RPW_A||h_A), V_A, x_A\}$ from the smart card $SC_A$. If it is correct, $GWN$ authenticates the user and proceeds the key distribution phase. $GWN$ can find $ID_A$ using the pair $\{ID_A, h_A\}$ stored in the database. However, $GWN$ cannot get $RPW_A$ in the database and other transmitted values. Therefore, $GWN$ cannot check the legitimacy of a user who wants to access things.

2) Incorrectness at the key distribution phase: To establish a session key between a user and a thing, each party should know the information $\{k_A, k_S, n_S\}$. $k_A$ is a random number generated by a user, and encrypted with the public key $h_B$ of $GWN$. To retrieve $k_A$ from the encrypted message $e_A$, Jeong et al.’s present the mathematical equation as follows:

\[
\begin{align*}
  a_B &= f_B * e_A \pmod{q} \\
  k_A &= f_B^{-1} * a_A \pmod{p}
\end{align*}
\]

Unfortunately, it is incorrect and cannot find $k_A$. The equation should be presented as follows:

\[
\begin{align*}
  a_B &= e_A * f_B \pmod{q} \\
  &= (pr_A * h_B + k_A) * f_B \pmod{q} \\
  &= (pr_A * h_B * f_B) + (k_A * f_B) \pmod{q} \\
  &= k_A * f_B \pmod{q} \\
  a_B * f_B^{-1} \pmod{p} &= k_A \pmod{q}
\end{align*}
\]

In addition, a user cannot retrieve $\{k_S||n_S\}$ from $a_A$, because $GWN$ sends $e_B$ but a user decrypt $e_S$. $e_S$ should be replaced with $e_B$.

B. Privileged Insider Attack

Jeong et al.’s analyzed their scheme and insisted that the scheme is secure against privileged insider attacks. However, we cryptanalyze and show that their scheme is vulnerable to the attacks. A malicious inside adversary can access to the database and compute user’s information, then guess identities of users. Using the information, the adversary can act as a legal user as follows:

1) $A$ performs an offline password guessing attack. $A$ guesses a password $PW_A'$ and computes $RPW_A' = H(PW_A'||x_A)$.
2) $A$ compares the computed value $H(ID_A||RPW_A'||h_A)$ with $H(ID_A||RPW_A||h_A)$ which is stored in $SC_A$. If it matches, $A$ successfully guesses the password of the user.

Therefore, Jeong et al.’s scheme does not provide security against privileged insider attacks.

C. Impersonation Attack

Jeong et al. claimed that their scheme is secure against impersonation attacks. However, once the inside adversary $A$ correctly guesses $PW_A$ and finds secret values, such as $ID_A$ and $RPW_A$, $A$ can generate the login message $\{I_A, e_A, h_A\}$. Then, $A$ can impersonate the user.

D. Session Key Disclosure Attack

Jeong et al.’s insisted that the scheme provides session key disclosure attacks. But, we show that their scheme is weak to this attack. An inside adversary can access to the database and obtain secret information pair of Things $\{n_S, z_S\}$. The adversary can compute a session key using the information as follows:

1) The insider adversary $A$ who knows $\{n_S, z_S\}$ can acquire $c_B$ and $c_S$ which are transferred via a insecure channel.
2) $A$ can compute $k_A = c_B \oplus z_S, k_S = c_S \oplus z_S$, because $A$ knows $z_S$ from the database and $c_B$ and $c_S$ from an insecure channel.
3) $A$ who successfully computes $k_A$ and $k_S$ can finally derive a session key $SK = H(k_A||k_S||n_S)$, because $n_S$ is also disclosed.

Therefore, Jeong et al.’s scheme is vulnerable to session key disclosure attacks.

V. COUNTERMEASURES

In this section, we present the fixes for the incorrectness and the countermeasures to improve the security weakness of Jeong et al.’s scheme.
A. Fixes for the Incorrectness

Jeong et al.’s scheme cannot provide the user authentication, because $GWN$ cannot compute $IA$ using the data in the database. $GWN$ should know the pair $\{ID_A, RPW_A\}$ to compute $IA$ and verify the validity of user $UA$. Therefore, $GWN$ should store the three sets $\{ID_A, RPW_A, h_A\}$ in the database at the user registration phase.

Unfortunately, if $GWN$ store that tuple in the database, privileged insider can easily obtain $ID_A$ and $RPW_A$, and compute $IA$ without password guessing process. To solve this problem, I recommend not to allow storing the identity and password of a user in the database. Instead, a user generates a pseudo-identity and sends it to $GWN$ for the verification. There are many authentication schemes which do not allow to store the identity of a user but provide an authentication of a user [16–[18].

The fixes for the incorrectness at the key distribution phase are introduced at Section IV.

B. Countermeasure of Privileged Insider Attacks

The privileged insider adversary $A$ can use the data $ID_A$ stored in the database. $A$ uses this identity and the data $H(ID_A||RPW_A||h_A)$ stored in the smart card. Unfortunately, $H(ID_A||RPW_A||h_A)$ is not utilized along Jeong et al.’s scheme, i.e. it is useless data. Therefore, $GWN$ does not need to store the data when it generates a smart card. If that data is not in the smart card, $A$ cannot correctly guess $PW_A$ and $RPW_A$. Then the scheme provide security against privileged insider attacks.

C. Countermeasure of Impersonation Attacks

The adversary $A$ can impersonate the user, because $A$ can computes $IA$. However, we mentioned, just before, that the scheme could provide security against privileged insider attacks and $A$ cannot guess $PW_A$ and $RPW_A$. Therefore, $A$ cannot computes $IA$ as well.

D. Countermeasure of Session Key Disclosure Attacks

The session key is easily disclosed, because the random numbers $k_A$ and $k_S$ are encrypted with same key $z_S$. To prevent this attack, the random number should be encrypted with other data [19], [20] or another method to conceal data [21].

VI. CONCLUSIONS

User authentication and key agreement are important security requirements for IoT environments. And several multi-factor authentication schemes have been proposed in recent years. However, these schemes are vulnerable to quantum computing attacks and the security threats should be resolved. Recently, Jeong et al.’s proposed a NTRU-based user authentication scheme in IoT environments. They insisted that their scheme provides various security properties, even security against the quantum computing attacks. Unfortunately, we found out that their scheme has some incorrectness in authentication phase and security weakness against the privileged insider adversary. We presented the fixes for the incorrectness and the countermeasure for the security weakness. The scheme with the countermeasures provides a proper user authentication and security properties against various attacks.

For further works, we are designing completely a security-enhanced NTRU-based user authentication scheme in IoT environments.

ACKNOWLEDGMENT

This research was sponsored by ‘QKD based next generation KREONET project’ of KISTI.

REFERENCES


