PQCMC: Post-Quantum Cryptography McEliece-Chen Implicit Certificate Scheme

Abel C. H. Chen
Information & Communications Security Laboratory, Chunghwa Telecom Laboratories
Email: chchen.scholar@gmail.com
ORCID: 0000-0003-3628-3033

Abstract—In recent years, the elliptic curve Qu-Vanstone (ECQV) implicit certificate scheme has found application in security credential management systems (SCMS) and secure vehicle-to-everything (V2X) communication to issue pseudonymous certificates. However, the vulnerability of elliptic-curve cryptography (ECC) to polynomial-time attacks posed by quantum computing raises concerns. In order to enhance resistance against quantum computing threats, various post-quantum cryptography methods have been adopted as standard (e.g. Dilithium) or candidate standard methods (e.g. McEliece cryptography), but state of the art has proven to be challenging to implement implicit certificates using lattice-based cryptography methods. Therefore, this study proposes a post-quantum cryptography McEliece-Chen (PQCMC) based on an efficient random invertible matrix generation method to issue pseudonymous certificates with less computation time. The study provides mathematical models to validate the key expansion process for implicit certificates. Furthermore, comprehensive security evaluations and discussions are conducted to demonstrate that distinct implicit certificates can be linked to the same end entity. In experiments, a comparison is conducted between the certificate length and computation time to evaluate the performance of the proposed PQCMC. This study demonstrates the viability of the implicit certificate scheme based on PQC as a means of countering quantum computing threats.

Keywords—implicit certificate, post-quantum cryptography, McEliece cryptography, code-based cryptography, ECQV

I. INTRODUCTION

Due to the development of quantum computing and Shor’s algorithm [1], numerous mainstream asymmetric cryptography techniques (such as RSA and elliptic-curve cryptography (ECC)) are susceptible to polynomial-time attacks. Consequently, the US National Institute of Standards and Technology (NIST) has initiated a call for proposals in the domain of post-quantum cryptography (PQC) methods. Various categories of PQC methods have emerged, including lattice-based cryptography, code-based cryptography, hash-based cryptography, multivariate-based cryptography, and supersingular elliptic curve isogeny cryptography [2]. Furthermore, lattice-based cryptography has emerged as one of the leading PQC approaches, offering efficient operations for encryption, decryption, signature, and verification [2].

For generating implicit certificates, the elliptic curve Qu-Vanstone (ECQV) technique is a widely used approach to achieve pseudonymous certificates within security credential management systems (SCMS) and secure vehicle-to-everything (V2X) communication [3],[4]. In efforts to enhance the security of SCMS and V2X communication, Shim conducted a survey to consider potential standard post-quantum cryptography (PQC) methods (such as Dilithium and Rainbow) and compared the performance of PQC methods and ECC methods [5]. In the realm of issuing implicit certificates for SCMS and V2X communication, Bindel and McCarthy tried to design emulating ECQV methods utilizing Falcon and Dilithium, and they examined common failed patterns based on lattice-based cryptography methods [6]. It has proven to be challenging to implement implicit certificates using lattice-based cryptography methods.

Therefore, this study designs an ECQV-like code-based PQC implicit certificate scheme for the issuance of implicit certificates in SCMS and V2X communication. A post-quantum cryptography McEliece-Chen (PQCMC) is proposed, which offers implicit certificates based on the principles of McEliece cryptography. The contributions of this study are summarized and emphasized as follows.

- The proposed PQCMC implicit certificate scheme enables the provision of pseudonymous certificates without necessitating signatures.
- Mathematical models are given to prove the process of key expansion for implicit certificates.
- An efficient method for generating a random invertible matrix, with a time complexity of O(n), is proposed to support the proposed PQCMC scheme with less computation time.

The structure of this manuscript include five sections. Section II presents the background of ECC, ECQV, and McEliece cryptography. Section III illustrates the proposed PQCMC and provides the proof and applications. Section IV obtains the evaluation and comparison of the proposed PQCMC. Finally, Section V concludes the contributions of this study and discusses the future work.

II. BACKGROUND

In this section, the theories of ECC and elliptic curve digital signature algorithm (ECDSA) are presented in Subsection II.A and Subsection II.B. For illustration of implicit certificates, ECQV implicit certificate scheme and the implicit certificate schemes in security credential management system (SCMS) are described in Subsection II.C and Subsection II.D, respectively. Lastly, McEliece cryptography and McEliece-based digital signature scheme are shown in Subsection II.E and Subsection II.F.

A. Elliptic Curve Cryptography

An elliptic curve (EC) is defined as Eq. (1), and the coordinate of an EC point is denoted as (x, y). The values of a coefficient α, a constant β, and a prime modulus n in Eq. (1) are defined in the specification [7] for various security levels. For key generation, a private key a is randomly generated, and the public key A (Ax, Ay) can be calculated based on a base point G (Gx, Gy) by A = aG; the EC point addition and EC point doubling can be found in [7],[8].

Ⅰ
\[ y^2 = x^3 + ax^2 + \beta \pmod{n}. \] (1)

For compression, an EC point can be compressed to only include a tag and the x-coordinate, and the tag can represent whether the last bit of y-coordinate is one or zero. For decompression, the value of the y-coordinate can be calculated by Eq. (1) in accordance with the tag and the x-coordinate. For instance, the EC of NIST P-256 is selected as one of standard ECs for SCMS [3]. An uncompressed EC point has a length of 65 bytes, which includes a tag (one byte), a x-coordinate (32 bytes), and a y-coordinate (32 bytes). Moreover, a compressed EC point has a length of 33 bytes, including only a tag (one byte) and a x-coordinate (32 bytes). Consequently, utilizing a compressed EC point can save thirty-two bytes.

\[ s = (h + ax_b) / r \pmod{n}. \] (2)

For signature verification, the EC point Z can be calculated using Eqs. (3), (4), (5), and (6). Furthermore, if the EC point Z is equals to the key P, the signature verification passes. The mathematical proof for this assertion is represented by Eq. (7) \cite{9,10}.

\[ w = l / s = \frac{r}{h + ax_b} \pmod{n}. \] (3)

\[ u = hw = h / s = hr / (h + ax_b). \] (4)

\[ v = x_s w = x_s / s = x_s r / (h + ax_b). \] (5)

\[ Z = uG + vA = uG + vaG = (u + va)G. \] (6)

\[ u + va = hr / (h + ax_b) + [ax_s r / (h + ax_b)] = r. \] (7)

For the example of NIST P-256, The length of the signature (R, s) is 97 bytes with an uncompressed EC point P (65 bytes) and the integer number s (32 bytes). Furthermore, the length of the signature can be reduced to 65 bytes \cite{4} by utilizing a compressed EC point P (33 bytes) mentioned in Subsection II.A.

\[ hP = \frac{hr + c}{(mod \ n)} \]

\[ Q = hP + C \]

\[ qG = (ha + b)G \]

\[ = (ha + hr + c)G \]

\[ = h(aG + rG) + cG \] (8)

\[ = h(A + rG) + C \]

\[ = hP + C = Q. \]

D. ECQV Implicit Certificate Scheme in SCMS

The structure of certificate in SCMS, as defined in the IEEE 1609.2 standard \cite{3}, includes version, type, issuer, to-be-signed certificate, and signature, as depicted in Fig. 2. The type indicates whether the certificate is explicit or implicit. An explicit certificate has a verification key (an EC point) within the verify key indicator (VKI) and the signature signed by a CA. Furthermore, an implicit certificate has a reconstruction value (an EC point) within the VKI, but it does not contain a signature \cite{3}.

\[ hP \]

\[ Q \]

\[ C \]

\[ qG \]

\[ (ha + b)G \]

\[ = (ha + hr + c)G \]

\[ = h(aG + rG) + cG \]

\[ = h(A + rG) + C \]

\[ = hP + C = Q. \]
F. McEliece-based Digital Signature Scheme

This subsection presents the McEliece-based digital signature scheme, which involves a private key \{K_1, K_2, K_3\}, a public key L, a decoder matrix \(K_1\) based on the generator matrix \(K_2\).

To generate the signature \(s\) (with dimensions \(\zeta^* \times 1\)) of the message \(m\) (with dimensions \(\zeta^* \times 1\)), the signature function \(S(m, (K_1, K_2, K_3))\) is performed using the private key \(K_1, K_2, K_3\) as defined in Eq. (11). Furthermore, the verification function is executed with the utilization of the public key \(L\) as defined in Eq. (12), to verify the signature \(s\) [16],[17].

\[
S(m, (K_1, K_2, K_3)) = K_3^{-1}K_4K_1^{-1}m = s. \quad (11)
\]

\[
V(s, L) = Ls = K_1K_2K_3^{-1}K_4K_1^{-1}m = K_1K_2K_4K_1^{-1}m = K_1K_4^{-1}m. \quad (12)
\]

III. THE PROPOSED METHODS

This section proposes a PQCMC implicit certificate scheme in Subsection III.A. The proof and security evaluation of the proposed PQCMC are discussed in Subsection III.B and Subsection III.C. Furthermore, an efficient method for generating a random invertible matrix is proposed in Subsection III.D.

A. The Process of the Proposed PQCMC

In the illustration of PQCMC, EE 1 possesses its McEliece-based private key \(\{K_{1,\text{EE}}, K_{2,\text{EE}}, K_{3,\text{EE}}\}\), public key \(L_{\text{EE}}\), and information \(E\) as detailed in this subsection. Furthermore, a certificate authority (CA) holds its McEliece-based private key \(\{K_{1,\text{CA}}, K_{2,\text{CA}}, K_{3,\text{CA}}\}\) and public key \(L_{\text{CA}}\) which are utilized to generate the implicit certificate of EE 1 \((C_E)\) based on the encoding of the McEliece-based reconstruction value \((B)\) and the information of EE 1 \((E)\). Fig. 3 illustrates the PQCMC process for generating the implicit certificate \(C_E\) by the CA. The PQCMC process is presented as follows.

Firstly, EE 1 generates its private key \(\{K_{1,\text{EE}}, K_{2,\text{EE}}, K_{3,\text{EE}}\}\). The corresponding public key \(L_{\text{EE}}\) (i.e. \(K_{1,\text{EE}}K_{2,\text{EE}}K_{3,\text{EE}}\)) can be determined and then sent to the CA. Subsequently, the CA generates a random number \(r\) and a random invertible matrix.
\( M_i \) (with dimensions \( \zeta_i \times \zeta_i \)) based on \( r \). The random invertible matrix \( M_i \) can be signed by the CA using the signature function \( \mathcal{S}(M_i, \{ K_{1,CA}, K_{2,CA}, K_{3,CA} \}) \) (i.e. \( K_{3,CA}^{-1}K_{1,CA}^{-1}M_i^{-1}m \)) to produce \( T \). Furthermore, the matrix \( T \) can be encrypted by EE 1’s public key and the encryption function \( e(T, L(E)) \) (i.e. \( TL(E) \)) to create \( B \) which serves as a reconstruction value. The implicit certificate of EE 1 (\( C_E \)) includes the reconstruction value \( B \), EE 1’s information \( E \), and other necessary elements. Importantly, no signature is required for the implicit certificate. Finally, the CA transmits the random number \( r \) along with EE 1’s certificate \( C_E \) to EE 1.

To generate the updated private key, EE 1 can decrypt the reconstruction value \( B \) using its private key (i.e. \( \{ K_{1,EE}, K_{2,EE}, K_{3,EE} \} \)) with the decryption function \( d(B; \{ K_{1,EE}, K_{2,EE}, K_{3,EE} \}) \) to obtain the matrix \( T \). The hash value of the concatenation of \( H(C_E) \) and \( H(C_{CA}) \) is calculated as \( h \) based on EE 1’s certificate \( C_E \) and the CA’s certificate \( C_{CA} \) for generating a random invertible matrix \( M_h \) (with dimensions \( \zeta_i \times \zeta_i \)). Finally, the updated private key of EE 1 can be represented as \( q \) (i.e. \( \{ M_hL_{CA}TK_{1,EE}, M_hL_{CA}TK_{2,EE}, K_{3,EE} \} \)).

To generate the updated public key, each device can receive EE 1’s certificate \( C_E \) and extract the reconstruction value \( B \) from \( C_E \). The hash value \( h \) and the random invertible matrix \( M_h \) can be calculated following the procedure described in the previous paragraph. Consequently, the updated public key of EE 1 can be expressed as \( Q \) (i.e. \( M_hL_{CA}B \)) utilizing the hash value \( h \) and the CA’s public key \( L_{CA} \).

**B. Proof**

This subsection proves the proposed PQCMC based on mathematical models. The plaintext \( m \) can be signed by the CA’s private key, resulting in the signature \( K_{3,CA}^{-1}K_{4,CA}^{-1}M_i^{-1}m \) as described in Eq. (11). This signature can then be verified using the CA’s public key, allowing the retrieval of \( m \) (as shown in Eq. (13)). Furthermore, the plaintext \( m \) can be encrypted by EE 1’s public key, leading to \( mL(E) \), as specified in Eq. (9). Subsequently, decryption using EE 1’s private key, as outlined in Eq. (10), restores the original plaintext \( m \) (as demonstrated in Eq. (14)).

\[
L_{CA}K_{3,CA}^{-1}K_{4,CA}^{-1}M_i^{-1}m = K_{1,CA}^{-1}K_{2,CA}K_{3,CA}^{-1}K_{4,CA}^{-1}M_i^{-1}m = m. \tag{13}
\]

\[
l_{CE}(K_{3,EE})^{-1}K_{4,EE}(K_{1,EE})^{-1} = m. \tag{14}
\]

The updated EE 1’s public key \( Q \) (i.e. \( M_hL_{CA}B \)) can be derived from the updated EE 1’s private key \( q \) (i.e. \( \{ M_hL_{CA}TK_{1,EE}, K_{3,EE} \} \)). Eq. (15) proves that the multiplication of the private key \( \{ M_hL_{CA}TK_{1,EE}, K_{3,EE} \} \) (i.e. \( M_hL_{CA}TK_{1,EE}, K_{3,EE} \)) is equivalent to the public key \( M_hL_{CA}B \). Furthermore, the inverse matrix of the first matrix in the private key can be computed using Eq. (16), enabling encryption and signing operations based on the private key.

\[
M_hL_{CA}TK_{1,EE} = M_hL_{CA}TL(E) = M_hL_{CA}B = Q. \tag{15}
\]

\[
\begin{align*}
(M_hL_{CA})^{-1}K_{1,CA}^{-1} & = (M_hL_{CA}K_{3,CA}^{-1}K_{4,CA}^{-1}M_i^{-1})^{-1} \\
& = (M_hL_{CA}K_{3,CA}^{-1}K_{4,CA})^{-1}M_iK_{1,CA}^{-1} = (M_hL_{CA})^{-1}(M_hM_iK_{1,CA}^{-1})^{-1} \\
& = K_{1,CA}^{-1}M_i^{-1}M_h^{-1}. \tag{16}
\end{align*}
\]

**C. Security Evaluation and Discussions**

The advantages of the proposed PQCMC are summarized and discussed as follows.

- The PQCMC implicit certificate includes solely a reconstruction value (akin to the public key of an end entity) and the end entity’s information. No signature is necessitated within the implicit certificate.
- A random number \( r \) is generated for each implicit certificate. Therefore, different implicit certificates can be attributed to the same end entity.
- The updated public key can be derived through the CA’s public key, the reconstruction value in the end entity’s implicit certificate, and the hash of both the CA’s certificate and the end entity’s certificate. This facilitates verification of the issuer of the implicit certificate without necessitating a signature.
- The proposed PQCMC supports the issuance of pseudonymous certificates. The updated public key is obtained by the reconstruction value \( B \) within which a random invertible matrix \( M_i \) is incorporated. As a result, the original public key of the end entity remains undisclosed.

**D. The Proposed Random Invertible Matrix**

An efficient method is proposed for generating a random invertible matrix with a time complexity of \( O(\zeta_i) \), as outlined in Algorithm 1. The inputs consist of a random number \( r \) and the matrix size \( \zeta_i \), which are used to generate an \( \zeta_i \times \zeta_i \) matrix \( M_i \) and the inverted matrix \( M_i^{-1} \) of matrix \( M_i \). The random number \( r \) is adopted as the seed of a pseudo-random number generator, ensuring that the generated matrices exhibit no disparities when using the same random seed. The pseudo-random order can be generated and reflected in the matrix \( I \). Subsequently, the \( \zeta_i \times \zeta_i \) matrices \( M_i \) and \( M_i^{-1} \) can be derived based on the arrangement specified by the matrix \( I \).

**Algorithm 1 Random invertible matrix generation method**

**Input:** a random number \( r \) and the matrix size \( \zeta_i \)

**Output:** a \( \zeta_i \times \zeta_i \) matrix \( M_1 \) and a \( \zeta_i \times \zeta_i \) matrix \( M_2 \)

1: Set \( r \) as the random seed
2: Create an \( 1 \times \zeta_i \) increment matrix \( I \)
3: Create an \( \zeta_i \) \( \times \zeta_i \) zero matrix \( M_1 \) and another \( \zeta_i \) \( \times \zeta_i \) zero matrix \( M_2 \)
4: for \( i = 0 \) to \( \zeta_i - 1 \) do
5: Generate a pseudo-random integer number \( j \) which is between 0 and \( \zeta_i - 1 \)
6: Swap(I[j], I[j])
7: end for
8: for \( i = 0 \) to \( \zeta_i - 1 \) do
9: Set \( M_1[I[i]] = 1 \)
10: Set \( M_2[I[i]] = 1 \)
11: end for
12: return \( M_1 \) and \( M_2 \)
IV. EVALUATION AND DISCUSSIONS

The lengths and computation time are compared for evaluating the proposed PQCMC in the following subsections.

A. The Comparison of Lengths

Table I offers a comparison of various McEliece methods [14],[18]-[22], focusing on the proposed PQCMC implicit certificate scheme. The objective is to illustrate the dimensions of the public key $L_{E}(2)$ (with dimensions $\zeta \times \zeta^2$), the reconstruction value $B$ (with dimensions $\zeta \times \zeta^2$), and the signature $s$ (with dimensions $\zeta \times 1$). Notably, the results indicate that the proposed PQCMC is capable of generating implicit certificates without requiring a signature, thereby obviating the need for the length of $s$. While the length of the reconstruction value in the proposed implicit certificate might exceed that of the public key in an explicit certificate, the proposed PQCMC enables the issuance of pseudonymous certificates to enhance privacy.

### TABLE I. LENGTH COMPARISON

<table>
<thead>
<tr>
<th>McEliece Method</th>
<th>$L_{E}(2)$</th>
<th>$B$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>[14]</td>
<td>$554, 1024$</td>
<td>$253 KB$</td>
<td>$354 KB$</td>
</tr>
<tr>
<td>[19]</td>
<td>$1219, 1702$</td>
<td>$242 KB$</td>
<td>$354 KB$</td>
</tr>
<tr>
<td>[20]</td>
<td>$1606, 2048$</td>
<td>$438 KB$</td>
<td>$512 KB$</td>
</tr>
<tr>
<td>[21]</td>
<td>$7151, 2048$</td>
<td>$925 KB$</td>
<td>$1233 KB$</td>
</tr>
<tr>
<td>[22]</td>
<td>$3504, 4096$</td>
<td>$1802 KB$</td>
<td>$2048 KB$</td>
</tr>
<tr>
<td>(Random invertible matrix method)</td>
<td>$12288, 19440$</td>
<td>$4815 KB$</td>
<td>$5986 KB$</td>
</tr>
</tbody>
</table>

B. The Comparison of Computation Time

The typical time complexity for generating a random invertible matrix is $O(n^3)$ [23]. However, Kong et al. proposed an improved method based on g subgroups that reduces the time complexity to $O\left(\frac{n^3}{\theta}\right)$ [23], as shown in Table II. In this study, Algorithm I achieves a time complexity of $O(n)$, enabling the efficient generation of a random invertible matrix.

### TABLE II. THE TIME COMPLEXITY COMPARISON

<table>
<thead>
<tr>
<th>Random invertible matrix generation method</th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>The proposed method</td>
<td>$O(n)$</td>
</tr>
</tbody>
</table>

V. CONCLUSIONS AND FUTURE WORK

The proposed PQCMC, serving as a post-quantum cryptography-based implicit certificate scheme, can issue pseudonymous certificates for enhancing privacy. However, a limitation of the proposed PQCMC lies in the length of the reconstruction value in V2X communication. To circumvent this limitation, a potential approach is to preload the implicit certificate onto devices and embed the certificate’s digest into the secure protocol data unit, resulting in reduced lengths.

REFERENCES


