# Design and Implementation of Multiple ECQV Implicit Certificate Generation Algorithms

[1]Eun-Gi Kim

***Abstract: Background/Objectives: A method for generating multiple certificates with a single CA-authenticated certificate in a vehicle communication system having an ECQV-based certificate.***

***Methods/Statistical analysis: Using the features of ECC (Elliptic-Curve Cryptography), the author proposes a method to generate a multiple ECQV implicit certificates with a single certificate generation request. The vehicle sends the public key and the number of required certificates to the CA. The CA passes the ECQV certificate and the values needed to generate multiple certificates to the vehicle. Findings: In this paper, we implemented the proposed multiple ECQV implicit certificate scheme and analyzed its performance on the embedded board. In this case, the security library used in the implementation is Openssl, and the embedded board for performance analysis is FaLinux's ez-s3c6410. The setup and request time for generating the certificate are almost the same as the existing method. In the case of generating multiple certificates according to the existing ECQV method, one request processing time is required when generating each certificate. In contrast, in the proposed method, even when generating multiple certificates, only one request processing time is required. Experimental results show that certificate generation time is shorter than public key generation time from certificate, and secret key generation time is much shorter than public key generation time. This is because in public key generation in ECQV, the arithmetic operation of the points in the elliptic curve is required, whereas the secret key requires a simple integer arithmetic operation. Improvements/Applications: We propose a method to generate multiple ECQV implicit certificates by generating only one public key. The performance of the proposed method is analyzed and confirmed to be superior then the conventional method.***

***Keywords: Vehicle, Communication, Security, Certificate, ECQV, WAVE***

## I. INTRODUCTION

Recently, research on vehicle communication has been rapidly increasing, and it is expected that more and more vehicles will be connected to the network in the future. When many vehicles are connected by a network, there are various network attacks that may interfere with the normal communication of the vehicles. In order to prevent such attacks, the encryption and authentication functions of the messaging are basically required[1,2].

A certificates can be efficiently used for mutual authentication of vehicles transmitting and receiving data and encryption/decryption of transmitted messages. Currently, certificates made according to the X.509 standard are used most frequently. In this method, a user submits his/her identifier and public key to a certificate authority (CA). This certificate will be signed by using CAs own private key, and this document containing the "name, public key, signature" will become a certificate[3].

The well-known public key security algorithms used for signatures are RSA (Rivest, Shamir, Adleman) and ECC. In particular, ECC methods based on an elliptic curve has recently been widely used as a solution to the problem of increasing the key length, which is a disadvantage of RSA. The ECC scheme, which can effectively encrypt data using a key having a small length, is recognized as a method that can be effectively utilized in a mobile terminal that is constrained by a performance of a processor or a battery[4].

As the ECC-based security method evolves, an ECQV (Elliptic Curve Qu-Vanstone) implicit certificate, which is different from the existing certificate, has been proposed. The ECQV implicit certificate does not store the public key value in the certificate differently from the existing X509 certificate, and the receiving side calculates the public key using the contents of the certificate. This ECQV implicit certificate scheme has advantages of supporting small size and fast processing speed compared with existing X509 based certificates[5,6].

In case of the vehicle communication system, a message authentication function and an encryption/decryption function capable of judging whether or not the transmitted and received messages are falsified should be supported. To support these functions, the IEEE vehicle communication standards describes the use of implicit certificates based on ECQV for real-time processing of communication functions[7,8,9]. In order to support the anonymity of the communicating vehicle, one vehicle has several certificates and periodically uses different certificates[10].

When an vehicle operates in the ECQV implicit certificate scheme, the vehicle should have a number of ECQV certificates at the beginning of the operation, and so a plurality of public keys must be created and submitted to the CA[11,12]. In this paper, we propose a method to generate multiple ECQV certificates by generating only one public key and submitting it to CA, and analyzed the performance by implementing it.

Section 2 describes the studies related to certificates, and section 3 describes how to generate multiple ECQV certificates. Section 4 describes the performance analysis and section 5 concludes the paper.

**Revised Manuscript Received on January 03, 2019.**

**Eun-Gi Kim**, Department of Information and Communication Engineering, Hanbat National University, Korea (ROK)

## II. CERTIFICATE RELATED RESEARCHES

### 2.1. Traditional certificate scheme

Figure 1 shows the X509 certificate generation process that is currently in use. As shown in the figure, the hash value is calculated by inputting the "user name, public key", and the digital signature is the encrypted hash value encrypted with the CA's secret key. A certificate is one that adds an electronic signature to the original document that contains the user name and public key. Commonly used hash algorithms are SHA2/3 and SHA256. RSA and ECDSA (elliptic curve digital signature algorithm) are used to encrypt the hash values.

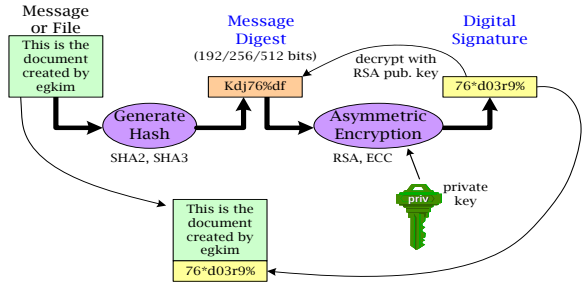


***Figure 1. General certificate generation procedures***

Figure 2 shows the certificate validation process. The certificate receiver calculates the hash value by inputting the value of "user name, public key". In addition, a value obtained by decrypting the signature value in the certificate with the public key of the certificate owner is calculated. If the two values are equal, it is confirmed that the contents of the certificate are valid.

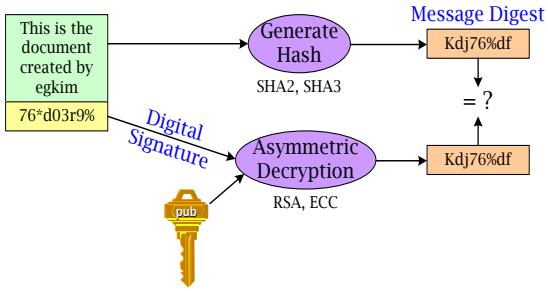


**Figure 2. General certificate verification procedures**

### 2.2. ECQV implicit certificate scheme

The entire process of the ECQV implicit certificate scheme, as shown in Figure 3, performs the $Setup$ operation first. The user performs a $Cert\_Request$ requesting the generation of a certificate, and the CA that receives the certificate request performs a $Cert\_Generate$ to generate an implicit certificate. A user who has used an implicit certificate can perform a $Cert\_PuKey\_Extract$/ $Cert\_PrKey\_Extract$ to calculate a public key and a secret key.

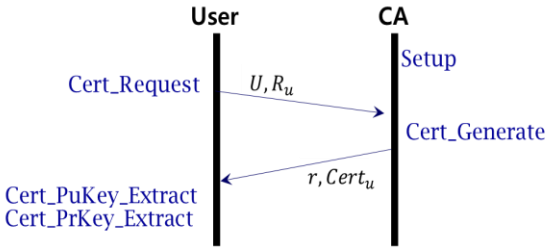


***Figure 3. Overall procedures of implicit certificate scheme***

■ $Setup$

The CA initially sets the parameters required for ECC.

- EC coefficients $a$ and $b$.
- Base point generator $G$.
- Order of the base point generator $n$.
- Cofactor $h$.
- CA key pair (private key $d_{CA}$, public key $Q_{CA} = d_{CA}G$)

■ $Cert\_Request$

The user selects an arbitrary value $k_u$ and calculates a $R_u$

- Select random $k_U \in_R [1, \ldots, n-1]$
- $R_u = k_u G$

■ $Cert\_Generate$

CA receives $u$ (user id), $R_u$ as input, calculates $r$, $Cert_u$ and delivers it to the user.

- Select random $k \in_R [1, \ldots, n-1]$
- $P_u = R_u + kG$
- $Cert_u = Encode(P_u, u, *)$
- $e = H_n(Cert_u)$
- $r = ek + d_{CA} \pmod n$

■ $Cert\_PuKey\_Extract$/ $Cert\_PrKey\_Extract$

The user calculates the public key and secret key from received $r$, $Cert_u$

- $e = H_n(Cert_u)$
- Calculated public key: $Q_u = eP_u + Q_{CA}$
- Calculated private key: $d_u = ek_u + r \pmod n$

An ECQV implicit certificate has the following advantages over an existing certificate:

- The ECQV certificate has a smaller size than the existing certificate because the ECQV certificate does not have the digital signature, while the X509 based certificate must store the digital signature and the public key.
- In the X509 based certificate, the digital signature of the certificate must be verified in order to check whether the public key value is normal. In the ECQV certificate, the public key value can be extracted and processed faster.

## III. MULTIPLE IMPLICIT CERTIFICATE GENERATION SCHEME

In the vehicle communication system, in order to guarantee the anonymity of the communicating vehicle, the vehicle has several certificates and periodically changes the certificate. The use of ECQV implicit certificates is recommended in order to reduce the size of the certificates and enable faster processing. To install multiple certificates on a vehicle, multiple temporal public key $R_u$ (in 2.2, $Cert\_Request$ procedure) must be calculated and delivered to the CA.

In this paper, we propose a method to generate a number of ECQV implicit certificates by using a feature of $(a+b)G = aG + bG$ ECC scheme, so that the vehicle can generate multiple certificate only a single $R_u$ calculation. Figure 4 shows the overall process of the proposed method.

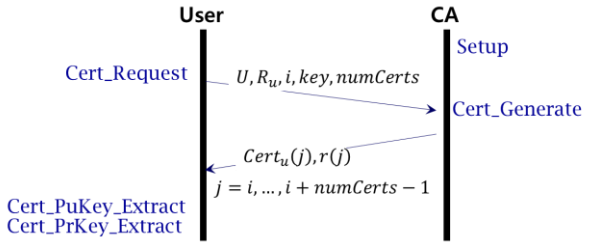


***Figure 4. Overall procedures of our designed implicit certificate scheme***

■ *Setup*

This is similar to the setup procedure in section 2.2.

■ *Cert_Request*

The user selects an arbitrary value $k_u$ and calculates $R_u$. Choose publicly known $i$, $key$, $f_{key}(i)$, and the number of certificates you want to generate, $numCerts$.

- Select random $k_U \in_R [1, \dots, n-1]$
- $R_u = k_u G$
- Publicly known $f$,$i$, $key$ (for calculating a $f_{key}(i)$)
- The number of needed certificate $numCerts$

■ *Cert_Generate*

CA calculates $numCerts$ values of $r(j)$, $Cert_u(j)$ and using $u$, $R_u$, $i$,$key$, $numCerts$ as inputs and delivers them to the user.

- Select random $k \in_R [1, \dots, n-1]$
- Loop $[(j = i);\ (j \le numCerts - 1);\ (j++)]$

$R_u(j) = R_u + f_{key}(i)G$

$P_u(j) = R_u(j) + kG$

$Cert_u = Encode(P_u(j), u, numCerts, key, \quad *)$

$e(j) = H_n(Cert_u(j))$

$r(j) = e(j)k + d_{CA} \pmod n$

■ *Cert_PuKey_Extract / Cert_PrKey_Extract*

The user calculates $numCerts$ values of public key and private key from received $r(j)$, $Cert_u(j)$

- Loop $((j = i);\ (j \le numCerts - 1);\ (j++))$

$k_u(j) = k_u + f_{key}(i)$

$e(j) = H_n(Cert_U(j))$

Calculated public key: $Q_u(j) = e(j)P_u(j) + Q_{CA}$

Calculated private key: $d_u(j) = e(j)k_u(j) + r(j) \pmod n$

Equation 1 shows that the computed public key $Q_u(j)$ and private key $d_u(j)$ values are normal.

$$\begin{aligned} Q'_u(i) &= d_u(i)G = \big(e(i)k_u(i) + r(i)\big)G \\ &= e(i)k_u(i)G + r(i)G \\ &= e(i)k_u(i)G + (e(i)k + d_{CA})G \\ &= e(i)k_u(i)G + e(i)kG + d_{CA}G \\ &= e(i)(k_u(i)G + kG) + d_{CA}G \\ &= e(i)P_u(i) + Q_{CA} = Q_u(i) \end{aligned} \quad (1)$$

## IV. PERFORMANCE VERIFICATION AND ANALYSIS

In this paper, we implemented the proposed multiple ECQV implicit certificate scheme and analyzed its performance on the embedded board. The security library used in this study is Openssl, and the embedded board for performance analysis is FaLinux's ez-s3c6410 board (CPU ARMv61, RAM 1G, Linux Kernel 4.10.17) [13].

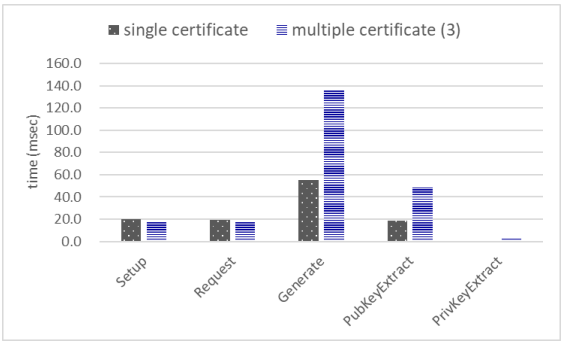


**Figure 5. Performance of our designed certificate scheme**

Figure 5 shows the results of performance comparison between generating a single certificate and generating three certificates according to the proposed method. The $f_{key}(i)$ is as follows.

$f_k(i) = AES_k(0^{128}\ XOR\ i)\ ||\ AES_k(1^{128}\ XOR\ i)$

$i < 2^{128}$

$x^y,\ x \in \{0, 1\}$ Array of value x, length y

As shown in Fig. 5, the setup and request time for generating a single certificate are almost the same as those proposed in this study. In the case of generating multiple certificates according to the existing ECQV method, one request processing time is required when generating each certificate. In contrast, in the proposed method, even when generating multiple certificates, only one request processing time is needed. Thus, as shown in Figure 5, only one request time is required to generate three multiple certificates according to the proposed method.

Figure 5 shows that the time required to extract the secret key from the ECQV certificate is shorter than the public key extraction time. This is because ECQV requires the arithmetic operation of the points in the elliptic curve to generate the public key, whereas the secret key requires a simple integer operation. In addition, in the proposed method, when $n$ certificates are generated, only "$n$ ×one certificate generation time" is required, and no additional time is required. This is also the case when generating a public key from a certificate (*Cert_PuKey_Extract*).

## V. CONCLUSION

A certificates are widely used for mutual authentication and message encryption of terminals that send and receive data. The currently used X509 certificate has a structure including "name, public key, signature" value. In recent years, as the elliptic curve based security method has evolved, an ECQV implicit certificate has been proposed which is different from the existing publicly used certificate. The ECQV implicit certificate has the advantage of being able to support a smaller size and faster calculation speed than existing X509 certificates, and is expected to be widely used in mobile communication systems. In particular, in the case of vehicle communication, it is recommended that a single car has multiple certificates and periodically changes the certificates to ensure the anonymity of the communicating car.

In this paper, we propose a method to generate multiple ECQV implicit certificates by generating only one public key and sending it to CA. In addition, we implemented the proposed method and confirmed that it is superior then the traditional methods in terms of processing speed and amount of data to be transmitted.

In the future, we plan to apply the results of this study to the standard of vehicle communication security standard.

## REFERENCES


1. G. Karagiannis et al. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards, and Solutions, J. IEEE Comm. Surveys & Tutorials, 2011; 13(4): 584-616.
2. Michael W. Whalen, Darren Cofer, and Andrew Gacek. Requirements and Architectures for Secure Vehicles, IEEE Software, 2016; 33(4): 22-25
3. D. Richard Kuhn, Vincent C. Hu et al. Introduction to public key technology and the federal PKI infrastructure, NIST Standards. SP 800-32, 2011 Feb.
4. Lawrence C. Washinton. Elliptic Curves Number Theory and Cryptography, 2nd ed. CRC Press, 2008
5. Chang-Seop Park. A Secure and Efficient ECQV Implicit Certificate Issuance Protocol for the Internet of Things Applications, IEEE Sensors Journal, 2017; 17(7): 2215-2223
6. EC 4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV), Standards for Efficient Cryptography, Certicom Research, 2013 Jan:
7. EEE Standard for Wireless Access in Vehicular Environments(WAVE) - Security Services for Applications and Management Messages, IEEE Standards 1609.2, 2016:
8. Roberto A. Uzcategui, Antonio Jose De Sucre, Guillermo Acosta-Marum. WAVE: A tutorial, IEEE Communications Magazine, 2009; 47(5): 126~133
9. John B. Kenney. Dedicated Short-Range Communications (DSRC) Standards in the United States, Proceedings of the IEEE, 2011; 99(7): 1162-1182
10. William Whyte. Andre Weimerskirch, Virendra Kumar, Thorsten Hehn, A Security Credential Management System for V2V Communications, Conference on IEEE Vehicular Networking, 2013 Dec: 1-8
11. Seol-Hee Sun, Eun-Gi Kim. A study on ECQV applied the butterfly key expansion algorithm, Conference on Korea Institute of Information and Communication Engineering, 2016 Oct; 20(2): 762-764. Available from https://www.eiric.or.kr/literature/ser_view.php?SnxGubun=INKO&mode=total&searchCate=literature&gu=INME051F6&cmd=qryview&SnxIndxNum=191354&rownum=&totalCnt=1&rownum=1&q1_t=QnV0dGVyZmx5IGtleSBleHBhbnNpb24=&listUrl=L3NlYXJjaC9yZXN1bHQucGhwP1NueEd1YnVuPUlOS08mbW9kZT10b3RhbCZzZWFyY2hDYXRlPWxpdGVyYXR1cmUmcTE9QnV0dGVyZmx5K2tleStleHBhbnNpb24meD0xMSZ5PTEw&q1=Butterfly+key+expansion
12. Tim Weil. VPKI Hits the Highway: Secure Communication for the Connected Vehicle Program, IEEE IT Professional, 2017;19(1): 59-63.
13. John Viega, Matt Messier, Pravir Chandra. Network Security with OpenSSL, O'Reilly Media, 2009 Feb.