ISSN: 0974-5823 Vol. 7 No. 1 January, 2022

International Journal of Mechanical Engineering

ECDSA fast nonce generation methods for vehicle communication systems

Eun-Gi Kim[*1]

[*1]Professor, Dept. of Information and Communication Engineering, Hanbat National Univ.,

125 Dongseo-daero Yuseong-gu Daejeon City, 34158, Republic of Korea

Abstract.

BACKGROUND/OBJECTIVES: Research on a vehicle communication system to improve the driving stability of a running vehicle is actively being conducted.

METHODS/STATISTICAL ANALYSIS: The digital signature, which is widely used in vehicle communication systems, supports the verification of the creator and contents of the transmitted message. The creator of the message transmitted from the vehicle calculates the digital signature using its own private key, and the third party verifies the received message using the signer's public key.

FINDINGS: The digital signature scheme, which started in the mid-1970s, has evolved from early developed RSA to DSA (Digital Signature Algorithm), and ECC (elliptic curve cryptography) based algorithm. In addition, NIST has announced dilithium, falcon, and rainbow as candidates for signature algorithms to be used in post-quantum cryptography. In a vehicle communication system to support the stable operation of a vehicle in motion, each vehicle periodically broadcasts information on its position, speed, and direction. A digital signature scheme is used to prevent falsification or modification of broadcasting messages. Currently, the most used digital signature method in vehicle communication systems is ECDSA. In the ECDSA method, a random number is required to calculate the signature value of a message. Thus, to compute the ECDSA digital signature for multiple consecutive messages, one must generate a random number when computing the signature value of each message. In this work, we propose a fast nonce generation scheme that can be used in such cases and analyze its performance.

IMPROVEMENTS/APPLICATIONS: The fast nonce generation scheme proposed in this study is thought to be effectively used in vehicle communication systems.

Keywords: ECDSA, vehicle, communication, nonce, signature

1. INTRODUCTION

As the use of electronic documents increases, the importance of digital signatures to help verify forgery of electronic documents is increasing. Digital signature is generated by encrypting the hash value of the document to be signed with the signer's secret key. A user who wants to verify an electronic document can compare the decryption value of the digital signature with the signer's public key and the hash value of the document, and if the two values are the same, the content of the document is not forged[1].

Since the RSA (Rivest, Shamir, Adleman) public key cryptographic algorithm was developed in the 1970s, various digital signature methods have been proposed[2]. Some of the representative digital signature methods initially proposed include RSA and DSA (Digital Signature Algorithm). NIST adopted the DSA approach as the FIPS 186 (Federal Information Processing Standardization 186) standard in 1994 and later amended it to establish standards such as FIPS 186–1 (1998), FIPS 186–2 (2000), FIPS 186–3 (2009), FIPS 186–4 (2013), and 186–5 (2019, draft)[3].

However, the DSA method based on the discrete logarithm problem is currently being demanded to reduce or prohibit its use due to problems such as parameter generation overhead and key length[4]. RSA, which is based on the integer factorization problem, has a key length (2048 ~ 3072 bits) problem and various vulnerabilities, so it is recommended to gradually stop using it[5].

An elliptic curve-based public key security algorithm is currently being actively used as a way to solve this problem of DSA or RSA security algorithms. The ECC (elliptic curve cryptography) method has various advantages over RSA, such as fast key generation, key length, and fast signature generation[6]. Digital signature methods using ECC include ECDSA (Elliptic Curve Digital Signature Algorithm) and EdDSA (Edwards-curve Digital Signature Algorithm)[3,7].

Beginning in 1985 with the proposed ECC algorithm, ECDSA began to be used in 2005. ECDSA has advantages over the existing RSA method, but various vulnerabilities are found as the use increases. Recently, EdDSA algorithms have been developed as a new digital signature method that can complement the vulnerabilities of ECDSA. The EdDSA method has advantages over the existing ECDSA method, such as superior processing performance, not using a random nonce that may harm the stability of the algorithm, and being more resistant to various side channel attacks[8,9].

Currently, the ECDSA algorithm is actively used in various fields. For example, ECDSA is also used in the WAVE specification, which is a standard for vehicle communication established by IEEE. The WAVE specification consists of IEEE 802.11p, which modified the IEEE WLAN (wireless LAN) specification to suit automotive communication, and WAVE protocols operating above it[10]. In the WAVE system, a moving vehicle periodically transmits information about its own speed and direction in broadcasting mode to improve driving stability. At this time, the transmitted messages are signed with ECDSA to prevent forgery and alteration[11]. Accordingly, a vehicle must sign several messages at a high speed, and must be able to quickly verify the signature values of messages received from neighboring vehicles. In this study, we propose a method to quickly sign consecutive messages with ECDSA and analyze the performance of the proposed method.

Chapter 2 describes typical elliptic curve based digital signature algorithms such as ECDSA and EdDSA, and Chapter 3 describes the design and performance of our fast-nonce generation scheme proposed in this study. Finally, Chapter 4 describes the conclusions and future research plans.

2. ECDSA (ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM)

This chapter describes the digital signature scheme of elliptic curve cryptography.

2.1. ECDSA OPERATIONS

ECDSA is the most commonly used ECC-based digital signature algorithm. Figure 1 shows the operation of ECDSA[2].

- Generate random number $k \in [1, n-1]$ (k is nonce)
- Compute $R(x, y) = kG$
- Set $r = x \pmod n$
- Compute $s = \frac{1}{k}(hash(M) + d_A r) \pmod n$

The signature for $M = (r, s)$

(a) Sign operation for message M

To verify a message M with (r, s)

- Compute $R'(x', y') = \frac{1}{s}(hash(M)G + rQ_A) \pmod n$
- If $(x' = r)$ accept
  else reject

(b) Verification operation for message M

Figure 1. ECDSA operations

As shown in the figure 1, the signer creates a private key (d_A) and calculates the public key ($Q_A = d_A G$, G: generator). Signer generates random number k (nonce) to sign message M and calculates R ($R(R_x, R_y) = kg$). The x-value of R, r, is used for the calculation of s, and this calculated (r, s) is the signature of message M. The user who wants to verify the message M calculates R' using M and the signature (r, s), and performs an operation to confirm that the x value of R' is the same as the received r. If these two values matches, message verification succeeds according to Equation 1.

$$\begin{aligned} R'(x', y') &= \frac{1}{s}(hash(M)G + rQ_A) \\ &= \frac{k}{hash(M)+d_A r}(hash(M)G + rQ_A) \\ &= \frac{k}{hash(M)+d_A r}(hash(M)G + rd_A G) \\ &= \frac{k}{hash(M)+d_A r}(hash(M) + rd_A)G \\ &= kG \\ &= R(x, y) \end{aligned} \quad (1)$$

2.2. ECDSA VULNERABILITY

The nonce used in ECDSA's signature must not be disclosed. When messages M, signature, and nonce k are released, the signer's secret key (d_A) may be exposed. Equation 2 shows that the signer's secret key d_A can be calculated from the signature (r, s), nonce k, and hash of message M.

$$\begin{aligned} s &= k^{-1}(H(M) + d_A r) \\ ks &= H(M) + d_A r \\ ks - H(M) &= d_A r \\ d_A &= r^{-1}(ks - H(M)) \end{aligned} \quad (2)$$

In addition, in ECDSA, nonce k should be newly created in the signature of each message and should not be reused. If the same nonce k is used for the signatures of messages M1 and M2, the nonce k can be calculated according to the Equation 3.

$$s1 = k^{-1}(H(m1) + d_A r)$$

$$s2 = k^{-1}(H(m2) + d_A r)$$

$$s1 - s2 = k^{-1}(H(m1) - H(m2))$$

$$k(s1 - s2) = (H(m1) - H(m2))$$

$$k = (s1 - s2)^{-1}(H(m1) - H(m2)) \quad (3)$$

If the nonce k used to sign the message M1 is calculated according to the Equation 3, any user can be able to calculate the signer's secret key d_A in accordance with Equation 2. A deterministic ECDSA method was proposed to prevent such nonce reuse attack. In the existing ECDSA method, the signer generated k randomly, but in deterministic ECDSA, k is determined by the calculation of HMAC with hash(M) and private key input. (k = HMAC (... || private key || hash(M))) Therefore, if the contents of the message to be signed are different, the nonce also has a different value, preventing nonce reuse attack.

2.3. EDDSA (EDWARDS-CURVE DIGITAL SIGNATURE ALGORITHM)

The EdDSA digital signature algorithm is simpler and faster than ECDSA and is a method that can prevent nonce reuse attacks. Figure 2 shows the operations of EdDSA.

private key: pk ∈ [1, q − 1]
public key: pubKey = pk × G (G: generator)
(a) signer key generation

EdDSA_sign(msg, pk) → { R, s }
r = hash[hash(pk)+msg] (mod q)
Calculate $R = rG$
h = hash[R + pubKey + msg] (mod q)
s = $(r + h \times pk)$ (mod q)
(b) Sign operation for message M

EdDSA_verify(msg, pubKey, signature (R, s))
h = hash[R + pubKey + msg] (mod q)
P1 = sG
P2 = $R + h \times pubKey$
P1==P2 → valid, P1≠P2 → invalid
(c) Verification operation for message M

Figure 2. EdDSA operations

As you can see in the figure 2, the signer generates a private key and a public key before signing. The signer who wants to sign the message M calculates the value of "hash(private key || M)" and uses the result as a nonce r. After that, calculate R, h and use it to calculate s. R and s calculated in this way are used as the signature value of message M.

The user who wants to verify the signature calculates the h, P1, and P2 values using the signer's public key, message, R and s.

$$P1 = sG$$
$$= (r + h \times pk) \times G$$
$$= rG + h \times pk \times G$$
$$= R + h \times pubKey$$
$$= P2 \quad (4)$$

If the calculated P1 and P2 are the same, the signature is verified as normal according to the Equation 4.

In the WAVE system that supports vehicle-to-vehicle communication, all vehicles running are periodically broadcasting on their speed or direction information for the safety operation of vehicles. In this case, all messages transmitted must be signed to prevent malicious messages from being transmitted. As we saw in Section 2.2., all signed messages must have different nonce values, and eventually, the sender must continuously generate random numbers. Random number is generated by processing digital information created from entropy of the system with CSPRNG (cryptographically secure pseudorandom number generator). The time required to generate the random number varies depending on the system configuration or state. In this study, we proposed a method to generate one random number and use it to support signing of multiple messages.

3. FAST NONCE GENERATION SCHEME

In this study, we proposed an efficient operation method when signing several consecutive messages with ECDSA. The proposed method in this study creates a nonce when signing the first message, and subsequently uses the value calculated during the signature process of the message to sign the next message. Therefore, it has the advantage of generating only one nonce even when the system signs multiple messages consecutively.

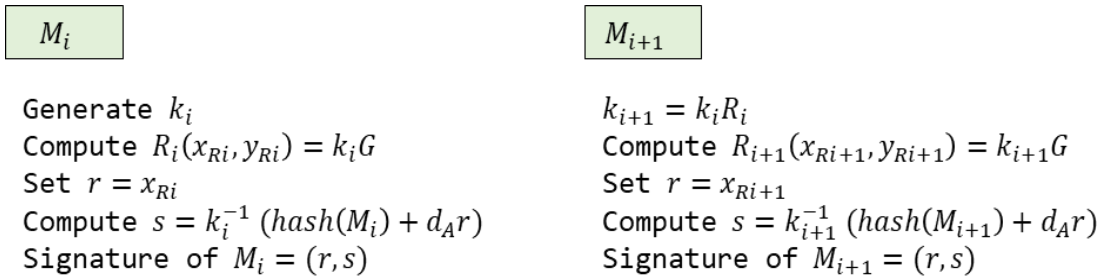
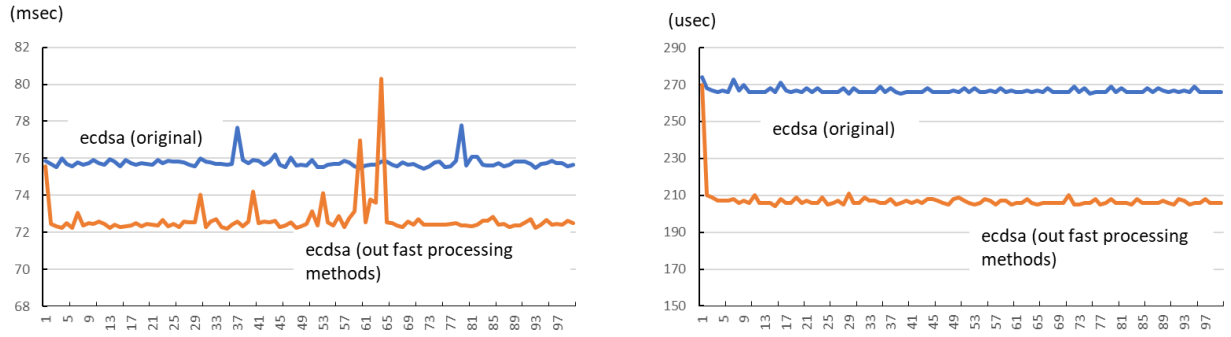


Figure 3. Proposed ECDSA signing algorithm

Figure 3 Shows the operation of the ECDSA signature algorithm proposed in this study. As can be seen from the figure 3, the signing process of the first message (M_i) is the same as the existing ECDSA operation, and a random nonce k_i is generated during this process. The value R_i calculated during the signing process of message M_i is used to calculate k_{i+1}, the nonce value to be used for signing the next message. Thus, random nonce is generated only when the signature of the first message is calculated, and subsequently the calculated value is used as nonce. If there are many messages to be signed, after performing a certain amount of message signing, a new nonce can be created to restart the process of Figure 3.

Figure 4 shows the performance of the fast-nonce generation scheme proposed in this study. Performance analysis was performed on a general desktop computer and embedded board, and the parameters used are as follows.

- Cryptography algorithm: ECDSA
- EC domain parameters: prime256v1 (NIST P-256, secp256r1)
- Simulation code: C with openssl library



(a) Desktop environments (Intel Core i7-6700K, 15G DDR3 RAM)

(b) Embedded board environments (BeagleBone Black Wireless, ARM® Cortex-A8)

Figure 4. Performance of our ECDSA algorithms

As you can see in the figure 4, the performance proposed in this study is different in desktop computers and embedded boards. The proposed method in this study showed about 25% performance improvement on the desktop, but about 5% performance improvement on the embedded board. This is thought to be due to differences in the method of generating random nonces in the two systems. In addition, the desktop computer showed more stable performance than the embedded board.

4. CONCLUSION

As the use of electronic documents increases, the use of digital signatures to help determine the authenticity of electronic documents is becoming more active. Digital signature provides a function to sign an electronic document with the signer's secret key and later verify it with the signer's public key.

For digital signature methods, DSA, ECDSA, and EdDSA were developed after RSA was created in the 1970s. Among them, ECDSA or EdDSA methods that use ECC algorithms are currently increasing their use as a way to overcome the shortcomings of RSA or DSA methods. In particular, ECDSA is the most commonly used ECC-style electronic signature algorithm today.

In this study, we proposed a modified ECDSA signature algorithm that could be used when continuous messages should be signed at high speed, and the performance of the proposed algorithm was analyzed. In the future, we plan to conduct research on the vulnerability of the algorithm proposed in this study.

5. REFERENCES

1. Rivest RL, Shamir A, Adleman L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM. 1978 Feb; 21 (2): 120-6. Available from: https://people.csail.mit.edu/rivest/Rsapaper.pdf
2. Johnson J, Kaliski B. Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. Internet Standard, RFC 3447. 2003 Feb. Available from: https://tools.ietf.org/html/rfc3447
3. FIPS PUB 186-1~4. Digital Signature Standard (DSS). Information Technology Laboratory NIST. 1998 Dec ~ 2013 July.

Available from: https://csrc.nist.gov/publications/detail/fips/186/4/final
4. Dimitrios Poulakis. New lattice attacks on DSA schemes. Journal of Mathematical Cryptology. 2016 May; 10 (2): 135-44. Available from: https://eprint.iacr.org/2016/058.pdf
5. Nemec Matus, Sys Marek, Svenda Petr, el al. The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017 Nov; 1631-48. Available from: https://acmccs.github.io/papers/p1631-nemecA.pdf
6. Menezes A, Okamoto T, Vanstone, SA. Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory. 1993 Sep;39 (5): 1639-46. Available from: https://ieeexplore.ieee.org/document/259647
7. Osefsson S, Liusvaara I. Edwards-Curve Digital Signature Algorithm (EdDSA). Internet Standard, RFC 8032. 2017 Jan. doi:10.17487/RFC8032. Available from: https://tools.ietf.org/html/rfc8032
8. Daniel J Bernstein. Ed25519: high-speed high-security signatures. 2017 Jan. Available from: https://ed25519.cr.yp.to/
9. Kaiyu Zhang, Sen Xu, Dawu Gu, et.al. Practical Partial-Nonce-Exposure Attack on ECC Algorithm. 13th International Conference on Computational Intelligence and Security. 2017 Dec; 248~52.
10. ITS joint program office. IEEE 1609 - Family of Standards for Wireless Access in Vehicular Environments (WAVE). US Department of Transportation. 2013 Apr. Available from: https://www.standards.its.dot.gov/Factsheets/Factsheet/80
11. IEEE Standards Association. IEEE Standard for Wireless Access in Vehicular Environments Security Services for Applications and Management Messages. IEEE Std 1609-2-2016. (Revision of IEEE Std 1609-2-2013)