JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 26 No. 11, pp. 111-120, November 2021
https://doi.org/10.9708/jksci.2021.26.11.111

Improved Digital Signature Algorithm Based on Batch Verification

Hye-jin Kim*

*Assistant Professor, Dept. of General Education, Kookmin University, Seoul, Korea

[Abstract]

This paper proposes an efficient SM2 digital signature. The batch verification algorithm is especially suitable for application scenarios that require verification of a large number of digital signatures such as electronic money. The algorithm does not verify immediately after each signature, but verifies multiple signatures at the same time. Because in the SM2 digital signature verification process, the dot multiplication operation is a very time-consuming operation, the batch verification algorithm significantly shortens the entire verification process by reducing the time-consuming dot multiplication operation in the verification process, and greatly improves the verification efficiency. Experimental data shows that in the case of the same number of messages, the efficiency of the batch verification algorithm is much higher than the efficiency of a single verification algorithm. For example, when the number of signatures reaches about 1 million, a single verification algorithm takes about 1 hour, while a batch verification algorithm only needs 2 seconds.

▸ **Key words**: Digital, Signature, Batch verification, SM2, Authenticity

[요 약]

본 논문은 효율적인 SM2 디지털 서명을 제안한다. 배치 검증 알고리즘은 특히 전자화폐와 같은 다수의 디지털 서명을 검증해야 하는 애플리케이션 시나리오에 적합하다. 이 알고리즘은 각 서명 직후에 확인하지는 않지만 동시에 여러 서명을 확인한다. SM2 디지털 서명 검증 프로세스에서 도트 곱셈 연산은 매우 시간이 많이 걸리는 작업이기 때문에 배치 검증 알고리즘은 검증 과정에서 시간이 많이 걸리는 도트 곱셈 연산을 줄여 전체 검증 과정을 크게 단축하고 검증 효율성을 크게 향상시킨다. 실험 데이터는 동일한 수의 메시지의 경우 배치 검증 알고리즘의 효율성이 단일 검증 알고리즘의 효율성보다 훨씬 더 높다는 것을 보여준다. 예를 들어 서명 수가 약 100만 개에 도달한 경우, 단일 검증 알고리즘 처리 시간은 약 1시간이 걸리는 반면 배치 검증 알고리즘은 2초만 있으면 된다.

▸ **주제어**: 디지털, 서명, 배치 검증, SM2, 진위

• First Author: Hye-jin Kim, Corresponding Author: Hye-jin Kim
*Hye-jin Kim (khj5187@kookmin.ac.kr), Dept. of General Education, Kookmin University
• Received: 2021. 10. 21, Revised: 2021. 11. 25, Accepted: 2021. 11. 25.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

Digital signatures are often used for identity authentication and non-repudiation. For example, the issue of bank pay, suppose that user B is the manager of bank X, and user A is an accountant of bank X (with 1 million employees). User A needs to the salary data and signatures of all employees (about 1 million) are sent to Bank X so that user B can transfer the salary to all employees. At this time, user B needs to verify the salary data and signatures of each employee sent by user A. It needs to do 1 million verifications, resulting in a lot of calculation work, which significantly reduces the efficiency of the bank transfer system. At this time, the batch verification of digital signatures is particularly important. Naccache et al. [1] first proposed the Batch Verification algorithm. The basic idea of the batch verification algorithm is to form a new set of multiple signatures (possibly from different signers) and verify the new set. If the set is verified, all signatures in the set are accepted, otherwise, Reject all signatures in the set. At present, for digital signatures such as DSA (Digital Signature Algorithm), RSA (Rivest, Shamir, Adleman) and ECD-SA (Elliptic Curve Digital Signature Algorithm), researchers have proposed corresponding batch verification algorithms [2-7]. Among them, Harn [8] designed an interactive batch verification algorithm for DSA, Hwang et al. [9] proposed a batch verification algorithm for DSA and RSA, and Cheon et al. [10] designed a fast batch verification algorithm for ECDSA* Multiple digital signature schemes.

In the rapidly developing modern application field of the Internet, in order to improve the speed of batch verification of digital signatures, researchers have proposed different schemes. Bayat et al. [11] proposed a security authentication scheme with batch verification function, which applies batch verification to In the Internet of Vehicles, when vehicles communicate with vehicles, and when vehicles communicate with fixed roadside units or cloud platforms, the use of batch verification can process data more quickly, reduce accidents, and improve traffic conditions. SM2 algorithm [12] is based on The ECC elliptic curve cryptography theory is independently developed and designed, and the prime number domain 256-bit elliptic curve is used as the standard curve to generate the key pair. In the ECDSA digital signature algorithm, no processing is performed on the signed message M, while the SM2 digital signature algorithm treats the signed message M Processed, including the user's distinguishable identifier, some elliptic curve system parameters and the hash value of the user's public key, which makes the security and repudiation stronger. Therefore, SM2 is superior to ECDSA and ECDH (Elliptic Curve Diffie- Hellman key exchange) and other algorithms. At present, SM2 is also used in blind signatures, proxy signatures, threshold cryptosystems and secure cooperation between two parties. However, no one has conducted research on SM2's batch verification algorithm. This article proposes SM2 digital signatures A batch verification algorithm for the same signer and different signers.

This paper proposes a safe and efficient SM2 batch verification algorithm. The algorithm does not verify immediately after each signature, but verifies multiple signatures at the same time. Because in the SM2 digital signature verification process, the dot multiplication operation is very expensive Therefore, this paper shortens the verification time by reducing the dot multiplication operation, thereby improving the efficiency of the algorithm. The experimental data shows that the efficiency of the batch verification algorithm is much higher than the efficiency of the single verification algorithm when the number of messages is the same. When the number reaches about 1 million (220), a single verification algorithm takes about 1h, while a batch verification algorithm only takes 2s.

II. Theoretical basis

In 1991, the U.S. government proposed the digital signature standard DSS (Digital Signature Standard) as a federal standard, using the digital signature algorithm (DSA) to sign electronic documents. The DSA digital signature algorithm is an ElGamal signature scheme based on the discrete logarithm problem [13], it requires at least 2 modular exponentiation operations to verify each signature. Since modular exponentiation is a very time-consuming calculation, it is necessary to use dedicated hardware or efficient software algorithms to accelerate the verification process.

In 1994, Naccache et al. [1] first proposed an interactive DSA batch verification algorithm. In the same year, Lim et al. [14] pointed out that this interactive DSA batch verification algorithm is insecure. In this scheme, the adversary can forge a set of signatures to satisfy the batch verification formula. In 1995, Harn [8] designed an interactive batch verification algorithm for the DSA digital signature algorithm. The signer interacts with the verifier to generate n signatures, and the verifier verifies the n signatures. In 1998, Harn [15] once again proposed a non-interactive batch verification algorithm for the DSA digital signature algorithm. It makes full use of the signer's public key as a verification link, and abandons the process of interacting with the verifier when the signer generates a signature. In 2001, Hwang et al. [16] pointed out that the signer in the document [16] can forge personal signatures and make the wrong batch verification effective.

In 1998, Harn [17] proposed a batch verification scheme for the RSA digital signature algorithm, using the same private key to sign multiple messages, and batch verification of multiple signatures. In 2000, Hwang et al. [18] Two methods are used to prove that in the scheme of literature [17], the verifier cannot verify whether the signature is legal. In 2001, Hwang et al. [16] proposed two simple batch verification algorithms for multiple digital signatures: BV-DSA and BV -RSA, where the BV-RSA scheme and the document [18] scheme have the same security flaws, that is, dishonest signers can forge personal digital signatures to make false batch verification effective. In 2006, Bao et al. [19] aimed at documents The BV-RSA scheme in [9] was cryptanalyzed and improved. In 2017, Kittur et al. [20] proposed a fast verification scheme for the RSA digital signature algorithm, and studied the batch verification of various RSA digital signature algorithms. Method, pointed out that batch verification of digital signatures is a promising research direction in IoT applications.

Compared with the DSA and RSA algorithms, the ECDSA digital signature verification algorithm has the advantages of fast speed, high strength and short signature [21]. ECDSA* is an improvement of ECDSA, which can be applied to Naccache et al. [1] for DSA digital In the batch verification algorithm proposed by the signature. In 2005, Adtipa et al. [22] proposed a scheme for accelerating the verification of ECDSA digital signatures, which can accelerate the efficiency of verifying ECDSA signatures and make them more efficient without increasing the complexity of the algorithm. An increase of more than 40%. In 2007, Cheon et al. [10] designed a scheme for rapid batch verification of multiple digital signatures for ECDSA* digital signatures, and proposed different batch verification algorithms for the same signer and different signers. In 2012, Bernstein et al. [23] proposed a rapid batch forgery recognition strategy for elliptic curve signatures, thereby reducing the cost of batch forgery recognition of elliptic curve signatures. In 2014, Karati et al. [24] proposed a new ECDSA digital signature batch verification algorithm. In 2019, Kittur et al. [25] proposed a more efficient batch verification algorithm for ECDSA*, which performs better on different batch sizes.

Table 1. Summary of development

Authors	Year	Improvement
Naccache et al.	1994	interactive DSA batch verification algorithm
Harn	1998	non-interactive batch verification algorithm
Hwang et al.	2000	batch verification scheme
Karati et al.	2014	new ECDSA digital signature batch verification algorithm
Kittur et al.	2017	fast verification scheme
Kittur et al.	2019	more efficient batch verification algorithm

III. SM2 digital signature batch verification algorithm

Define p as a large prime number and F_p as a finite field. Choose $a,b\in F_p$ as the parameter of the elliptic curve E. Define (x,y) as a point on the elliptic curve E, and use it as the generator of the group G The order of group G is n. $H_v(\ )$ is defined as a hash algorithm with a digest length of v bits.

3.1 Batch verification algorithm for the same signer

Assuming that the signer is A, its public-private key pair is (d_A,P_A), for different messages $M_i(i=1,2,\cdots,l)$, the generated signature message is $(r_i,s_i)(i=1,2,\cdots,l)$. Assuming that the verifier is B, then B needs to send the signature $(M_i,r_i^{'},s_i^{'})$ to A to verify and determine whether the signer is A, where $r_i=r_i^{'},s_i=s_i^{'}$. The verifier B uses the cryptographic hash function to obtain the hash value Z_A of the signer A, and concatenates it with M_i to obtain $\overline{M_i}=Z_A\parallel M_i$. The verifier B uses the cryptographic hash function for different Mis to produce different $e_i^{'}(i=1,2,\cdots,l)$, and sum them to get d. The verifier B calculates the (r_i,s_i) sent by the signer A to get t_(i∘). In the process of digital signature verification, the dot multiplication operation is a very time-consuming operation, that is, to calculate the point on the elliptic curve $(x^{'},y^{'})=s^{'}G+tP_A$. Therefore, this article first calculates the cumulative sum of $s_i^{'}$ andt_i, and then calculates the point on the elliptic curve $(x^{'},y^{'})$. Only one dot multiplication operation is required, thereby improving the efficiency of digital signature verification.

Assuming that the signer is A and its public-private key pair is (d_A,P_A), for the message M_i to be signed, randomly select $k_i\in[1,n-1]$, calculate:

$$(x_i,y_i)=k_iG$$
$$r_i=(e_i+x_i)\mathrm{mod}n \quad (1)$$

$$s_i=\left((1+d_A)^{-1}\cdot(k_i-r_id_A)\right)\mathrm{mod}n \quad (2)$$

Signer A sends $(r_i,s_i)(i=1,2,\cdots,l)$ as the signature of message M_i to verifier B.

Verifier B receives the signatures of all messages $(M_i^{'},r_i^{'},s_i^{'})$ $((i=1,2,\cdots,l)$, i is the same in steps 1 to 5), and performs batch verification.

Step 1 Check whether $r_i^{'}\in[1,n-1]$ is established, if not, the verification fails.

Step 2 Check whether $s_i^{'}\in[1,n-1]$ is established, if not, the verification fails.

Step 3 Calculate $\overline{M_i}^{'}=Z_A\parallel M_i^{'}$, where $Z_A=H_{256}$ $(ENTL_A\parallel ID_A\parallel a\parallel b\parallel x_G\parallel y_G\parallel x_A\parallel y_A)$. ID_A indicates that the same signer A has a distinguishable identifier with a length of $entlen_A$ bits. $ENTL_A$ is 2 bytes converted from integer $entlen_A$.

Step 4 Calculation $e_i^{'}=H_v(\overline{M_i}^{'})$.

Step 5 Calculation $t_i=(r_i^{'}+s_i^{'})\mathrm{mod}n$, if $t_i=0$, then the verification fails.

Step 6 Calculation $d=\sum_{i=1}^{l}e_i^{'}$ (3)

Step 7 Calculation $w=\sum_{i=1}^{i}t_i$. (4)

Step 8 Calculation $R=\sum_{i=1}^{l} r_i^{'}$.

Step 9 Calculation $u=\sum_{i=1}^{l} s_i^{'}$. (5)

Step 10 Calculate the points on the elliptic curve according to formula (4) and formula (5):

$$(x^{'},y^{'})=uG+wP_A \quad (6)$$

Step 11 Calculate the value of $R^{'}$according to formula (3) and formula (6): $R^{'}=(d+x^{'})\mathrm{mod}n$, check whether $R^{'}=R$ is established, if it is established, the verification is passed; otherwise the verification is not passed.

Because $M^{'}=M,(r^{'},s^{'})=(r,s)$,then $e^{'}=e$。And because $r_i=(e_i+x_i)(i=1,2,\cdots,l)$, then:

$$\sum_{i=1}^{l} r_i=\sum_{i=1}^{l} e_i+\sum_{i=1}^{l} x_i$$

The batch verification formula for the same signer is:

$$uG+wP_A+\sum_{i=1}^{l} e_i-\sum_{i=1}^{l} r_i$$

Just verify:

$$uG+wP_A=\sum_{i=1}^{l} x_i$$

According to equation (1), equation (2), equation (4), equation (5) and $r_i^{'}=r_i$, $s_i^{'}=s_i$ we get:

$$uG+wP_A=$$
$$\sum_{i=1}^{l} s_i^{'}G+\sum_{i=1}^{l} t_iP_A=$$
$$\sum_{i=1}^{l} s_i^{'}G+\sum_{i=1}^{l} (r_i^{'}+s_i^{'})\cdot d_AG=$$
$$\left(\sum_{i=1}^{l} s_i+\sum_{i=1}^{l} r_id_A+\sum_{i=1}^{l} s_id_A\right)\cdot G=$$
$$\left(\sum_{i=1}^{l}(1+d_A)CDOT\ s_i+\sum_{i=1}^{l} r_i\cdot d_A\right)\cdot G=$$
$$\left(\sum_{i=1}^{l}(1+d_A)\cdot \frac{k_i-r_id_A}{1+d_A}+\sum_{i=1}^{l} r_i\cdot d_A\right)\cdot G=$$
$$\sum_{i=1}^{l} k_i\cdot G=$$
$$kG=(x,y)=\textbf{Right}$$

This proves the correctness of batch verification of digital signature schemes generated by the same signer.

Security analysis theorem 1: In the case of the same signer, the SM2 digital signature batch verification algorithm is secure and has the same strength as a single verification algorithm.

This article uses the idea of proof by contradiction to prove the security of the SM2 batch verification algorithm under the same signer, that is, if the SM2 batch verification algorithm is not secure, it can be deduced that the SM2 standard single verification algorithm is not safe, so the SM2 batch verification algorithm is safe. Specific The proof is as follows:

In the same signer batch verification scheme, verification is required:

$$R=\sum_{i=1}^{l} r_i=\sum_{i=1}^{l} r_i^{'}=R^{'}$$

Where $r_i^{'}=s_i^{'}G+t_iP_A+e_i^{'}$。

Assuming that the adversary successfully tampered with the signed message Mi, and can satisfy:

$$\sum_{i=1}^{l} e_i=\sum_{i=1}^{l} e_i^{'}$$

But it is not possible to change the signed $r_1,r_2,\cdots,r_l$. Because the signature process can generate $r_1,r_2,\cdots,r_l$ for different messages M_i, the adversary must reveal that $r_1,r_2,\cdots,r_l$ are part of the SM2 signature.

In the verification process, this paper calculates the cryptographic hash function value $e_i^{'}$ of all messages M_i, calculates the points (x_i,y_i) on the

elliptic curve according to formula (6), and gives these x_i coordinates and verification conditions $R=R'$, For the corresponding y coordinates $y_1,y_2,\cdots,y_i,r_1,r_2,\cdots,r_l$, there is a unique solution. As long as l is limited to a small constant value, the adversary only needs moderate calculations to determine $y_1,y_2,\cdots,y_i$, which means that the adversary can reveal the coordinates x_i and r_i, and then know the coordinates of all points of (x_i,y_i).

The above argument is based on the uniqueness theorem of the solution, $r_1,r_2,\cdots,r_l$ meet the standard SM2 digital signature verification conditions. If the adversary can tamper with the data in the SM2 digital signature batch verification algorithm, then the adversary can also tamper with the algorithm in the standard SM2 digital signature. Data, so the security of SM2 digital signature batch verification algorithm is not lower than that of SM2 single digital signature verification algorithm.

3.2 Batch verification algorithm for different signers

Assuming that the signer is A_i, its public-private key pair is $(d_{A_i},,P_{A_i})$, for different messages M_{A_i}, the generated signature message is $(r_{A_i},s_{A_i})(i=1,2,\cdots,l)$. Assuming that the verifier is B, then B needs to verify the signature sent by $(M_{A_i},r_{A_i},s_{A_i})$ to determine whether its signer is A_i. In the process of digital signature verification, the dot product operation is a very time-consuming operation, that is, calculating the elliptic curve point $(x'_{A_i},y'_{A_i})=s'_{A_i}G+t_{A_i}P_{A_i}$. Therefore, this paper first calculates the cumulative sum of s'_{A_i} and $t_{A_i}P_{A_i}$ respectively, and then calculates the points on the elliptic curve (x'_{A_i},y'_{A_i}), only one dot multiplication operation is required, thereby improving the efficiency of digital signature verification.

Assuming that the signer is $A_i(i=1,2,\cdots,l)$, and its public-private key pair is (d_{A_i},P_{A_i}), for the message to be signed M_{A_i} randomly select $k_{A_A}\in[1,n-1]$,calculate:

$$(x_{A_i},y_{A_i})=k_{A_i}G$$
$$r_{A_i}=(e_{A_i}+x_{A_i})\mathrm{mod}n \quad (7)$$

$$s_{A_i}=\left((1+d_{A_i})^{-1}\cdot\ (k_{A_i}-r_{A_i}d_{A_i})\right)\mathrm{mod}n \quad (8)$$

The signer A_i sends (r_{A_i},s_{A_i}) as the signature of the message M_{A_i} to the verifier B.

Verifier B receives the signatures of all messages $(M_{A_i},r_{A_i},s_{A_i})((i=1,2,\cdots,l)$, i is the same in steps 1 to 5), and batches verify.

Step 1 Check whether $r'_{A_i}\in[1,n-1]$ is established, if not, the verification fails.

Step 2 Check whether $s'_{A_i}\in[1,n-1]$ is established, if not, the verification fails.

IDA_i indicates that different signers A_i have a distinguishable identifier with a length of $entlen_{A_i}$ bits: $entlen_{A_i}$ is 2 bytes converted from integer $ENTL_{A_i}$.

Step 4 Calculation $e'_{A_i}=H_v$.

Step 5 Calculation $t_{A_i}=(r'_{A_i}+s'_{A_i})\mathrm{mod}n$, if $t_{A_i}=0$, then the verification fails.

Step 6 Calculation $d=\sum_{i=1}^{l} e'_{A_i}$. (9)

Step 7 Calculation $R=\sum_{i=1}^{l} r'_{A_i}$.

Step 8 Calculation $u=\sum_{i=1}^{l} s'_{A_i}$. (10)

Step 9 Calculate the points on the elliptic curve according to formula (10):

$$(x'_{A_i},y'_{A_i})=uG+\sum_{i=1}^{l} t_{A_i}P_{A_i} \quad (11)$$

Step 10: Calculate the value of R'according to formula (9) and formula (11): $R'=(d+x'_{A_i})\mathrm{mod}n$, check whether $R'=R$ is established, if it is established, the verification is passed; otherwise Verification failed.

Because $M_{A_i}^{'} = M_{A_i}, (r_{A_i}^{'}, s_{A_i}^{'}) = (r_{A_i}, s_{A_i})$, then $e_{A_i}^{'} = e_{A_i}$. And because $r_{A_i} = (e_{A_i} + x_{A_i})(i = 1, 2, \cdots, l)$, then:

$$\sum_{i=1}^{l} r_{A_i} = \sum_{i=1}^{l} e_{A_i} + \sum_{i=1}^{l} x_{A_i}$$

The batch verification formula for different signers A_i is:

$$uG + \sum_{i=1}^{l} t_{A_i} P_{A_i} + \sum_{i=1}^{l} e_{A_i}^{'} ? = \sum_{i=1}^{l} r_{A_i}^{'}$$

Just verify:

$$uG + \sum_{i=1}^{l} t_{A_i} P_{A_i} = \sum_{i=1}^{l} x_{A_i}$$

According to equation (7), equation (8), equation (10) and $r_{A_i}^{'} = r_{A_i}, s_{A_i}^{'} = s_{A_i}$, we get:

$$uG + \sum_{i=1}^{l} t_{A_i} P_{A_i} =$$
$$\sum_{i=1}^{l} s_{A_i}^{'} G + \left(\sum_{i=1}^{l} (r_{A_i}^{'} + s_{A_i}^{'}) \cdot d_{A_i} \right) \cdot G =$$
$$\left(\sum_{i=1}^{l} s_{A_i} + \sum_{i=1}^{l} r_{A_i} d_{A_i} + \sum_{i=1}^{l} s_{A_i} d_{A_i} \right) \cdot G =$$
$$\left(\sum_{i=1}^{l} (1 + d_{A_i}) \cdot s_{A_i} + \sum_{i=1}^{l} r_{A_i} \cdot d_{A_i} \right) \cdot G =$$
$$\left(\sum_{i=1}^{l} (1 + d_{A_i}) CDOT \frac{k_{A_i} - r_{A_i} d_{A_i}}{1 + d_{A_i}} + \sum_{i=1}^{l} r_{A_i} d_{A_i} \right) \cdot G =$$
$$\sum_{i=1}^{l} k_{A_i} \cdot G = kG = (x_{A_i}, y_{A_i}) = \text{Right}$$

This proves the correctness of batch verification of digital signature schemes generated by different signers.

Theorem 2 In the case of different signers, the SM2 digital signature batch verification algorithm is secure and has the same strength as a single verification algorithm.

IV. Batch verification algorithm

In order to compare the operating efficiency of a single verification algorithm and a batch verification algorithm, this article uses C language programming to implement two solutions, and realizes the calculation on the elliptic curve by calling the OpenSSL cryptographic library. Based on the PC-side development, the operating environment is as follows:

This article executes 2 schemes 10 times, obtains the average value of the data, and uses ms as the time unit. In the same signer and different signer schemes, the same signer scheme at the key generation stage only needs to generate a pair of public and private key pairs, and Different signer schemes need to generate multiple pairs of public and private key pairs. Therefore, the key generation phase in different signer schemes takes much longer than the corresponding phase of the same signer scheme; in the signing and verification process, for the same signer and different signers, their parts consume almost the same time. Table 2 shows the comparison of the running time of a single verification algorithm and a batch verification algorithm for the same signer and different signers in the case of different numbers of messages.

In a single verification algorithm, the dot multiplication operation accounts for more than 95% of the verification time. In the digital signature verification process, the time of a single verification algorithm is linear with the number of messages. As the number of messages continues to increase, the verification time also continues to increase accordingly; In the batch verification algorithm, the most time-consuming dot product operation has nothing to do with the number of messages, and only needs one time. Although other calculations are linear with the number of messages, they take less time, so the verification time will not increase with the number of messages. Increase and rapid growth. When the number of signatures reaches about 1 million (220),

a single verification algorithm takes about 1h, while a batch verification algorithm only takes 2s. Therefore, when a large amount of message data is digitally signed at the same time, no matter it is the same signer Still different signers, using batch verification algorithms can greatly shorten the running time and significantly improve efficiency.

Table 2. Comparison of running time between single verification algorithm and batch verification algorithm

Plan		2^0	2^4	2^8	2^{12}	2^{16}	2^{20}
Same signer	Dot multiplication	3.6	65	1082	17162	269435	1779419
	Single verification	4.0	78	1140	18553	285318	1851231
	Batch verification	4.0	5	6	19	200	2125
Different signers	Dot multiplication	3.6	67	1077	17341	277403	1789623
	Single verification	4.0	79	1162	18626	288874	1857206
	Batch verification	4.0	5	6	18	203	2365

V. Conclusions

Digital signatures play an important role in identity authentication and non-repudiation. For example, in electronic money systems, merchants or consumers need to use digital signatures to sign or verify electronic money to ensure the security and correctness of electronic money information [26-30]. At the same time When performing digital signature verification on a large amount of electronic cash, it will greatly reduce the operating efficiency of the entire system, so the use of batch verification algorithms is particularly important. There are many mature batch verification algorithms for digital signatures such as DSA, RSA and ECDSA, but For the SM2 digital signature algorithm, there is no corresponding batch verification algorithm. For the first time, an efficient SM2 batch verification algorithm is proposed, which reduces the most time-consuming point multiplication operation from n times to 1 time (n is the number of signatures), which greatly shortens Verification time. The correctness and safety of the SM2 batch verification algorithm is proved, and experimental verification is carried out. From the experimental results, it can be seen that the efficiency of the batch verification algorithm is significantly improved compared to a single verification algorithm. When the number of signatures reaches about 1 million (220), a single verification algorithm takes about 1h, while a batch verification algorithm only takes 2s.

Currently, interest in the metabus system is increasing, and it is highly likely that this study will be used properly.

REFERENCES

[1] Naccache D., M´ Raihi D., Vaudenay S., et al. “Can D·S·A be improved? Complexity trade-offs with the digital signature standard”, Proceedings of Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, 1994, pp.77-85. DOI: https://doi.org/10.1007/BFb0053426

[2] Jung Hyun Kim, “Proposal for Advanced Attribute-based Encryption in Mobile Cloud Computing”, Asia-pacific Journal of Convergent Research Interchange, Vol.1, No.4, pp. 45-51, December 31, 2015. DOI: http://dx.doi.org/10.21742/apjcri.2015.12.07

[3] Pavan Yadav, “Advanced Looping Broadcast Proxy Re-Encryption in Cloud computing”, Asia-pacific Journal of Convergent Research Interchange, Vol.2, No.1, pp. 21-28, March 31, 2016. DOI: http://dx.doi.org/10.21742/APJCRI.2016.03.04

[4] Jae Yoon Lee, Mounika Durbha, “Customary Broadcast Encryption with Advanced Encryption and Short ciphertexts”, Asia-pacific Journal of Convergent Research Interchange, Vol.2, No.2, pp. 27-33, June 30, 2016. DOI: http://dx.doi.org/10.21742/APJCRI.2016.06.04

[5] Bhargavi Nadella, "Data Encryption using Geometric Range", Asia-pacific Journal of Convergent Research Interchange, Vol.2, No.3, pp. 21-28, September 30, 2016. DOI: http://dx.doi.org/10.21742/APJCRI.2016.09.03

[6] Su Min Shin, Vandana Roy, “Hybrid key-Based Encryption in

Cloud Storage", Asia-pacific Journal of Convergent Research Interchange, Vol.2, No.3, pp. 29-34, September 30, 2016 DOI: http://dx.doi.org/10.21742/APJCRI.2016.09.04

[7] V. Sujatha, "Auditing of Storage Security on Encryption", Asia-pacific Journal of Convergent Research Interchange, Vol.3, No.2, pp. 1-9, June 30, 2017. DOI: http://dx.doi.org/10.21742/APJCRI.2017.06.01

[8] Harn L, "DSA type secure interactive batch verification protocol", Electronics Letters, Vol.34, No.4, pp.257-258, February 16, 1995 DOI: https://doi.org/10.1049/el:19950203

[9] Hwang M. S., Lee C. C., Tang Y. L., "Two simple batch verifying multiple digital signatures", Proceedings of Information and Communications Security, Xian, China, 2001, pp.233-237. DOI: https://doi.org/10.1007/3-540-45600-7_26

[10] Cheon J. H., Yi J. H., "Fast batch verification of multiple signatures", Proceedings of Public Key Cryptography, Beijing, China, 2007, pp.442-457. DOI: https://doi.org/10.1007/978-3-540-71677-8_29

[11] Bayat M., Barmshoory M., Rahimi M., et al. "A secure authentication scheme for VANETs with batch verification", Wire-less Networks, Vol.21, No.5, pp.1733-1743, December 23, 2014. DOI: https://doi.org/10.1007/s11276-014-0881-0

[12] Chinese Encryption Administration, Public key cryptography algorithm SM2 based on elliptic curves, ISBN 0003-2012, Beijing, March 21, 2012. (in Chinese) WEB: http://www.gmbz.org.cn/upload/2018-07-24/1532401863206085511.pdf

[13] Djebaili K., Melkemi L., "Security and robustness of a modified elgamal encryption scheme", International Journal of Information and Communication Technology, Vol.13, No.3, pp.375-387, September 5, 2018. WEB: https://ur.booksc.eu/dl/75214976/11930b

[14] Lim C. H., Lee P. J., "Security of interactive DSA batch verification", Electronics Letters, Vol.30, No.19, pp.1592-1593, September 15, 1994. DOI: https://doi.org/10.1049/el:19941112

[15] Harn L., "Batch verifying multiple DSA-type digital signatures", Electronics Letters, Vol.34, No.9, pp.870-871, April 30, 1998. WEB: http://h.web.umkc.edu/harnl/papers/1998%20J1.pdf

[16] Hwang M. S., Lee C. C., "Cryptanalysis of the batch verifying multiple DSA-type digital signature", Pakistan Journal of Applied Sciences, Vol.1, No.3, pp.287-288, September 30, 2001. DOI: https://dx.doi.org/10.3923/jas.2001.287.288

[17] Harn L., "Batch verifying multiple RSA digital signatures", Electronics Letters, Vol.34, No.12, pp.1219-1220, June 11, 1998. DOI: https://doi.org/10.1049/el:19980833

[18] Hwang M. S., Lin I. C., Hwang K. F., "Crypt analysis of the batch verifying multiple RSA digital signatures", Informatica, Vol.11, No.1, pp.15-19, January 31, 2000. WEB: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.3241&rep=rep1&type=pdf

[19] Bao F., Lee C. C., Hwang M. S., "Crypt analysis and improvement on batch verifying multiple RSA digital signatures", Applied Mathematics and Computation, Vol.172, No.2, pp.1195-1200, January 15, 2006. DOI: https://doi.org/10.1016/j.amc.2005.03.016

[20] Kittur A. S., Jain A., Pais A. R., "Fast verification of digital signatures in IoT", Proceedigns of the International Symposium on Security in Computing and Communication, Manipal, India, 2017, pp.16-27. DOI: https://doi.org/10.1007/978-981-10-6898-0_2

[21] Johnson D., Menezes A., "The elliptic curve digital signature algorithm (ECDSA)", International Journal on Information Security, Vol.1, No.1, pp.36-63, January 31, 2001. DOI: https://doi.org/10.1007/s102070100002

[22] Antipa A., Brown D., Gallant R., et al. "Accelerated verification of ECDSA signatures", Proceedings of the International Workshop on Selected Areas in Cryptography, Kingston, ON, Canada, 2005, pp.307-318. DOI: https://doi.org/10.1007/11693383_21

[23] Bernstein D. J., Doumen J., Lange T, et al. "Faster batch forgery identification", Proceedings of International Conference on Cryptology in India, Kolkata, India, 2012, pp.454-473. DOI: https://doi.org/10.1007/978-3-642-34931-7_26

[24] Karati S., Das A., Roychowdhury D, et al. "New algorithms for batch verification of standard ECDSA signatures", Journal of Cryptographic Engineering, Vol.4, No.4, pp.237-258. November 30, 2014. DOI: https://doi.org/10.1007/s13389-014-0082-x

[25] Kittur A. S., Pais A. R. "A new batch verifications scheme for ECDSA* signatures", Sadhana, Vol.44, Article No.157, pp.157-169, June 7, 2019. DOI: https://doi.org/10.1007/s12046-019-1142-9

[26] Kui-hwa Park, "Impact of Digital Convergence Technology on the Logistics Value Chain", Asia-pacific Journal of Convergent Research Interchange, Vol.7, No.1, pp. 33-48, January 31, 2021. DOI: http://dx.doi.org/10.47116/apjcri.2021.01.04

[27] Yuan Lin, "Research on Interactively Digital Display for Cultural Heritage- Discovering the Hall of Mental Cultivation: A Digital Experience Exhibition", Asia-pacific Journal of Convergent Research Interchange, Vol.6, No.8, pp. 51-67, August 31, 2020. DOI: http://dx.doi.org/10.47116/apjcri.2020.08.06

[28] Sangbong Nam, Donoung Lee, "Developing Digital Musical Instrument, mPoi, by Taking Advantage of Sensor Interfaces: Focused on the Multimedia Music Piece, "Jwibul"", Asia-pacific Journal of Convergent Research Interchange, pp. 11-19, Vol.6, No.7, July 31, 2020. DOI: http://dx.doi.org/10.47116/apjcri.2020.07.02

[29] K. Asish vardhan, "Some Studies on Digital Image Segmentation Techniques", Asia-pacific Journal of Convergent Research Interchange, Vol.5, No.1, pp. 77-89, March 31, 2019. DOI: http://dx.doi.org/10.21742/apjcri.2019.03.08

[30] Byeong-Hyun Min, SiChul Kim, "Study on Growth Strategies of the Magazine Industry in the Digital Platform", Asia-pacific Journal of Convergent Research Interchange, pp. 65-72, Vol.3, No.4, December 31, 2017. DOI: http://dx.doi.org/10.14257/apjcri.2017.12.06.

Authors

Dr. Hye-jin Kim received the B.S. and M.Edu. degree in Wooseok University, Jeonju, Korea. She received the Ph.D. in Computer Science and Education from University of Bristol in 2017.

Dr. Kim worked as a lecturer in Jeonju Vision University Continuing Remote Education Center, Korea for 3 years. Now she is working as an Assistant Professor at Dept. of General Education, Kookmin University, Korea. Her research interests include U-learning, Education Technology, Artificial Intelligence, IoT, Remote Education Management Technology.