Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 6
ISET 2024

Recent Progress in ECDSA and DSA Digital Signature Algorithms

Senyi Gong

DUT-RU International School of Information Science & Engineering at DUT, Dalian University of Technology, Da Lian, China

2693896858@mail.dlut.edu.cn

Abstract. Digital signatures are essential for verifying the authenticity and integrity of information, thus facilitating digital authentication. The Elliptic Curve Digital Signature Algorithm (ECDSA) and the Digital Signature Algorithm (DSA) play critical roles in the field of digital signatures. This paper explores the conceptual frameworks and mathematical underpinnings of these algorithms, providing an in-depth analysis of their advantages and disadvantages with a focus on computational time complexity and key length requirements. The distinct applications of ECDSA and DSA in various fields are examined, highlighting how the selection of an algorithm influences security and efficiency in different scenarios. This analysis also considers the algorithms' suitability for specific security applications, offering insights into their strategic deployment. By comparing these characteristics, the study aims to guide the selection of appropriate digital signature technology, tailored to meet operational needs and security specifications. This comparative approach not only elucidates the practical implications of each algorithm but also contributes to a broader understanding of their roles in securing digital transactions.

Keywords: ECDSA Algorithm; DSA Algorithm; Digital Signature; Elliptic Curve Cryptography.

1. Introduction

Digital signatures are paramount in network information transmission, particularly due to their role in non-repudiation and verifying file integrity. The historical significance of digital signatures began with Whitfield Diffie and Martin Hellman's seminal 1976 paper, "New Directions in Cryptography," which proposed digital signatures to replace traditional written contracts with electronic communication. This innovative system not only mirrored the mathematical properties of written signatures but also facilitated easy verification of authenticity [1].

Building on the concept of individual identity verification, Rivest, Shamir, and Adleman introduced the RSA algorithm, leveraging the computational difficulty of factoring large integers. The RSA algorithm employs the product of two large prime numbers as a public key, part of which forms the private key. During encryption, plaintext is secured using the public key, while decryption is straightforward with the private key [2].

Given the security provided by the RSA algorithm and the challenges in decomposing large numbers, the question arose whether alternative encryption methods based on other complex mathematical problems could be viable. In 1985, ElGamal introduced a cryptography and signature system based on the discrete logarithm problem, leading to various derivative systems such as the linear Schnorr signature system. However, the shorter signature length of the Schnorr system compromises its security compared to ElGamal's proposal [3, 4].

In August 1991, NIST (National Institute of Standards and Technology) developed the Digital Signature Algorithm (DSA), drawing on the discrete logarithm problem for its security, similar to the Schnorr and ElGamal systems [5]. Although the DSA shares similar security attributes with the RSA, a distinctive feature of DSA is that the prime numbers in its public key are public, allowing even those without the private key to verify signatures and detect tampering. This feature is not possible with RSA, which is based on the complex problem of large number factorization.

In 1992, Vanstone introduced the Elliptic Curve Digital Signature Algorithm (ECDSA) based on elliptic curve cryptography. ECDSA effectively addressed the issue of signature forgery, ensuring that each public key is uniquely paired with a private key, thereby eliminating the possibility of forged signatures with duplicated public keys [6].

Current research on digital signatures predominantly focuses on public key cryptography systems, originally proposed by Diffie and Hellman. The mainstream digital signature algorithms explored and applied today can be categorized into three groups: RSA and its variants, which secure digital signatures through the challenge of large integer factorization; algorithms relying on the discrete logarithm problem in finite fields; and those based on elliptic curve cryptography (ECC), such as DSA and ECDSA, which depend on the complexity of elliptic curve problems. The following text will further analyze DSA and ECDSA along with their optimization algorithms.

2. DSA Algorithm

DSA was proposed in 1991 by the National Security Bureau and the National Bureau. It also proposed the DSS. DSA, as a special algorithm dedicated to signature, in addition to the key, a different random number will be used in each signature, and the length of the key should be at least 1024 bits and above, so as to ensure the security of the clear text in a medium and long term time line.

2.1. The DSA Algorithm Procedure

The signature process in the DSA algorithm process is shown in Figure 1 below, and the validation process is shown in Figure 2 below.

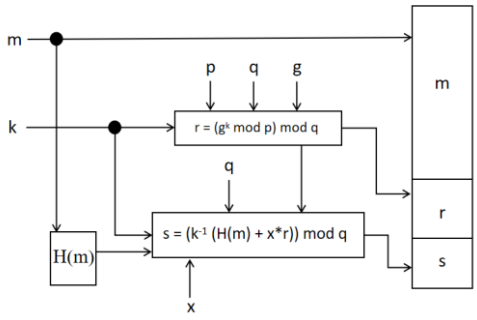

Figure 1. Signature (Photo credit: Original).

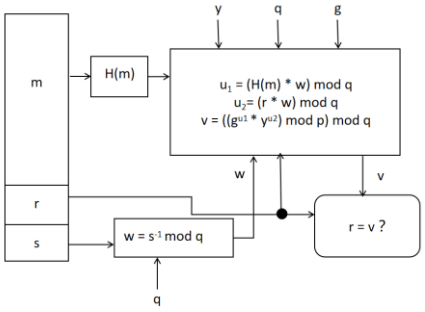

Figure 2. Verification (Photo credit: Original).

2.2. Preready

Select a prime of 512 to 1024 bits with a bit length of a multiple of 64 as p.

Select a prime number, q, so that q satisfies: q, as a factor of p-1.

Calculate a base number, g, so that g satisfies:g = h((p-1)/q)mod p, The base number h satisfies: and h <p-1 and h（p-1）/qmod p > 1.

Generate a global parameter (p, q, g) that can be shared by a set of users.

2.3. Key Generation

Generates a non-leaky, private key kept in the hands of the sender. And a public key that can be obtained by the information recipient.

2.3.1. Private key generation.

Generate a key, x, so that x satisfies: x <q.

2.3.2. Public Key Generation.

Part y of the public key is generated, Make y satisfied: y = gxmod p.

The combined public key (p, q, g, y).

2.4. Signature and Verification

2.4.1. Signature.

After obtaining the information m, first randomly obtain a number k, so that k satisfies: 0 <k <q.

Calculate the r in the signature of the information m so that r satisfies: r = (gkmod p) mod q.

Calculate the s in the signature of the information m, so that the s satisfies: s = (k-1(H(m) + x*r)) Modq, where the H (x) function is the one-Way Hash function and the H (m) is the hashing of the message m.

Get the signature of the information m: (r, s).

2.4.2. Validation.

After receiving the information m, the received signature (r, s) is calculated.

Range validation of r and s, if r satisfies: 0 <r <q and 0 <s <q.

First calculate the value of w, so that w satisfies: w = s-1mod q.

Calculate the index u of g1, send u1satisfied:u1= (H(m) * w) mod q.

Calculate the index u of the y2, send u2satisfied:u2= (r * w) mod q.

Calculate the validation value v so that v satisfies: v = ((gu1* yu2) mod p) mod q.

If v = r, the validity of the signature can be verified.

3. ECDSA Algorithm

In 1992, Vanstone proposed the elliptic curve digital signature algorithm ECDSA. The ECDSA full name is given by Elliptic curve digital signature algorithm, this signature algorithm is based on the elliptic curve password of the digital signature algorithm resimulation, its security is based on the problem of elliptic curve discrete log. However, the discrete logarithm problem of the elliptic curve is hardly a rising gradient compared to the discrete logarithm problem, so the unit bit intensity of the elliptic curve cryptography system is much higher than that of the traditional discrete logarithm system [7].

3.1. Algorithm Process

The signature process in the algorithm process of ECDSA is shown in Figure 3 below, and the validation process is shown in Figure 4 below.

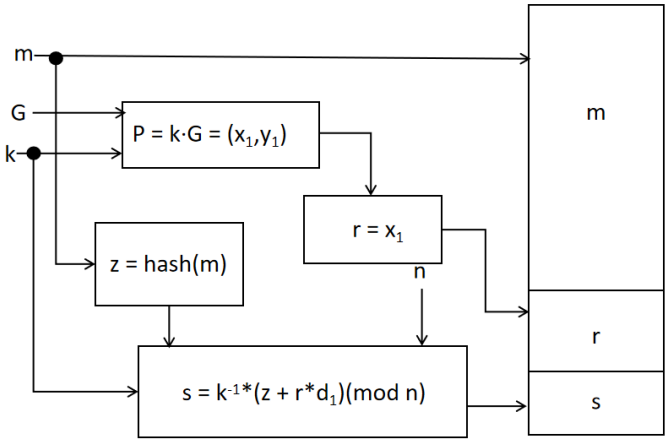

Figure 3. Signature (Photo credit: Original).

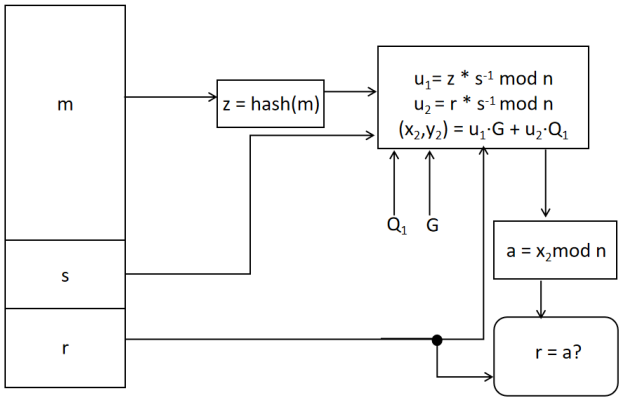

Figure 4. Verification (Photo credit: Original).

3.2. Pre-Process Preparation

Select a safe, elliptic curve, Ep (A, b) and a base point G located on the elliptic curve.

Take n so that n satisfies: n is the order of the base point G.

randomly select an integer k so that k satisfies: k <n.

3.3. Key Generation

Private key generation: To randomly generate a private key, d1, send d1satisfied:d1< n.

Public-key generation: Use the base point G and the public key d1Calculate public key Q1, send d1satisfied:Q1= d1·G.

3.4. Signature and Verification

3.4.1. Signature.

Calculate point P so that P satisfies: P = k · G = (x1,y1).

Calculate the integer r, so that r = x1, among x1x in point p1Of the integer form of the coordinate values of.

After obtaining the information m, calculate the hash value z = hash (m), and hash (m) is the hash function.

Calculate a part of the signature s so that s satisfies: s = k-1*(z + r*d1) (Modn), where k-1Is the multiplicative inverse of k about the module n.

If the value of r or s is 0, you need to choose the k value for calculation until r and s are not 0.

Get the signature of the information m: (r, s).

3.4.2. Validation.

After receiving the information, the information receiver first calculates the hash function z = hash (m).

count u1, send u1satisfied:u1= z * s-1mod n.

count u2, send u2satisfied:u2= r * s-1mod n.

precalculated position (x2,y2) = u1·G + u2·Q1.

Calculate an integer a, so that a satisfies: a = x2The modn, where the x2For the x in the dot2Integer conversion of the coordinate values.

If r = a, the validity of the signature can be verified.

4. Comparison of The ECDSA Algorithm with The DSA Algorithm

4.1. Comparison Of ECDS Algorithm and DSA Algorithm on Mathematical Basis

4.1.1. Key length comparison.

Because ensuring the security of information is the most basic purpose of the signature algorithm, so in the case of ensuring the same security, the shorter the key length, then the higher the transmission efficiency of the key. If you select the elliptical curve key with the key length of 161-256 bits in the ECDSA algorithm, then if the DSA algorithm wants to guarantee the key of the same security as the key generated by the ECDSA algorithm, it needs to select the key of 1024 -3072 bit length. It is clear that the ECDSA requires shorter keys for the same security purposes.

The requirements of the different key systems for the key length and size are shown in Table 1 below.

Table 1. Comparison of security key lengths between ECC and DSA.

ECC key length / bit	DSA key length / bit	Key Length Ratio (ECC: DSA)
106	512	1：5
132	768	1：6
160	1024	1：7
210	2048	1：10
600	21000	1：35

4.1.2. Comparison of the time complexity of the two algorithms in the signature step.

As a good expression of algorithm efficiency, to compare the computational complexity of the algorithm, the method selected in this paper is to compare the computational complexity of ECDSA and DSA algorithms by comparing the time complexity.

In the ECDSA algorithm, the point multiplication algorithm is mainly used, such as the calculation of the point P mentioned above, which is obtained by the point multiplication by using the random number k and the basis point G. The point multiplication algorithm can be seen as a process of repeatedly adding itself to a point on the elliptic curve. Example: 2 · P = P + P are shown in Figure 5 below.

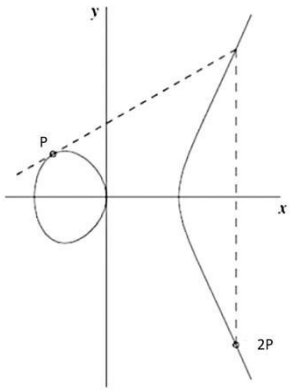

Figure 5. Example of the point-multiplication algorithm (Photo credit: Original).

If we discuss the point multiplication algorithm in the signature process, the theoretically required time complexity is O (n ^ 2 * log (p)) where n is the order of the elliptic curve and p is the modulus. However, because there is still a hash operation in the signature process, the theoretical time complexity is greater than the value shown above. Due to the use of efficient elliptic curve algorithms and novel hardware support in practical applications, the signature operation of ECDSA is usually very fast.in compliance withIn the finite-domain GF (2m) The optimized ECDSA algorithm successfully reduces the required time in the point multiplication operation [8]. Literature implements the hardware system architecture and component module of ECDSA signature algorithm based on verilog hardware description language design [9].

In the DSA algorithm, the hash function and the modular power operation are mainly used, and only the time complexity of the modular power operation reaches O (n2).With the same security, the DSA algorithm is much longer key length than the ECDSA algorithm, and more values need to be processed in the signature process, so its signature operation may be slower than the ECDSA algorithm. However, because the research of DSA algorithm is more mature, it also has a lot of hardware and software optimization strategies, such as optimizing the algorithm and prime selection, so as to improve the efficiency of digital signature encryption, Or DSA algorithm optimization for the high-speed implementation of SHA series encryption algorithm based on FPGA [10, 11].

4.2. Advantages of ECDSA Algorithm Over DSA Algorithm

The security of ECDSA is based entirely on the difficulty of the discrete logarithmic problem, which makes it very difficult for attackers to find the discrete logarithm of points on the elliptic curve in a finite domain. Therefore, after knowing the public key and the signature, it is difficult to find the corresponding private key in a short time. Compared to the DSA algorithm, based on the ECDSA algorithm of elliptic curve discrete log problem, because the unit bit strength of elliptic curve cryptography is higher than that of other public key systems, and ECDSA algorithm is small, ECDSA algorithm is better than DSA algorithm in the efficiency of signature generation, and ECDSA algorithm has relatively low demand for hardware and software resources.

4.3. Disadvantages of The ECDSA Algorithm

For ECDSA algorithm: this algorithm is easy to bypass attack. In the hardware implementation, the computational process of the ECDSA algorithm may produce specific power consumption patterns or electromagnetic radiation modes that may leak sensitive data in the private key or signature process. In the software implementation, the execution time of the ECDSA algorithm may be affected by the input data, and this change in the execution time may also be exploited by attackers to guess the private key or the signature result.

4.4. Advantages of DSA Algorithm Over ECDSA Algorithm

Although ECDSA algorithm is a younger algorithm, DSA algorithm has been through long-term research and practical demonstration, and its security and maturity have been widely used and recognized, and DSA has also shown excellent performance in preventing lattice attacks [12].

4.5. The Dsa Algorithm Shortcomings

For the DSA algorithm: in the practical application, the DSA algorithm often uses the common modulus number may bring some threats. If an attacker can access enough signatures and corresponding messages, they may use the information to try to guess the private key. This attack, although complex, is theoretically possible.

5. Algorithm Application

As an important technology to ensure information integrity and undeniability, digital signature has been widely used in many fields. Table 2 below will list the applied areas respectively based on the advantages of the ECDSA algorithm and the DSA algorithm itself.

Table 2. Application of ECDSA algorithm and DSA algorithm in various fields.

Classify	Domain	Technology	Effect	Focument	Choose the reason
ECDSA algorithm	Mobile e-commerce	Digital signature was performed using the ECDSA algorithm	in-house transaction	Literature [13]	The shorter key length reduces the cost and improves the efficiency
ECDSA algorithm	3D printing	The ECDSA encryption algorithm	Improve the security and integrity of 3D printed documents	Literature [14]	Faster encryption and decryption speed, less storage space footprint, higher security characteristics
ECDSA algorithm	Wireless sensor network	Both the OTS and ECDSA algorithms are used	Improve network life	Literature [15]	The ECDSA algorithm produces less amount of data per 1 bit of data than the OTS algorithm
DSA algorithm	Mobile communication user authentication scheme	DSA encryption algorithm	Ensure the security of the mobile communication system	Literature [16]	The calculation required by the user is in the expected calculation stage, and no calculation is required for real-time communication
DSA algorithm	Mobile phone short message early warning	The DSA digital signature algorithm	Make the coding of the mobile phone SMS visible within the range	Literature [17]	Ensure the safe transmission of early warning information in the channel, and can realize the security authentication from the receiving end to the sending end

6. Conclusion

The Elliptic Curve Digital Signature Algorithm relies on the elliptic curve discrete logarithm problem and is extensively utilized in digital trading due to its high security and superior performance. Similarly, the Digital Signature Algorithm, which is based on the integer finite domain discrete log problem, has long been a staple in numerous authentication schemes, often appearing in its original form or as a variant. Despite numerous improvements and optimizations to these algorithms, the advent of quantum computing technology poses a significant threat to existing cryptographic systems, including digital signature algorithms. Notably, the Shor algorithm could potentially solve the discrete logarithmic problem on elliptic curves, thereby compromising the private keys used in digital signatures.

In the realm of digital signatures, the challenge lies in identifying new, robust mathematical problems to develop novel signature systems or enhancing the decryption difficulty of existing systems without compromising efficiency. This is essential to counter the advancing capabilities of quantum computing technology.

References

[1] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Trans. Inf. Theory 22(6) (1976) 644-654.

[2] R.L. Rivest, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 26(2) (1978) 96-99.

[3] T. Elgamal, A public-key cryptosystem and signature scheme based on discrete logarithms, IEEE Trans. Inf. Theory 31(4) (1985) 469-472.

[4] C.P. Schnorr, Efficient signature generation by smart cards, J. Cryptol. 4(3) (1991) 161-174.

[5] X. Zhu, H. Xu, Z. Zhao, et al., "An Environmental Intrusion Detection Technology Based on WiFi," Wireless Personal Communications, vol. 119, no. 2, pp. 1425-1436, 2021.

[6] R.L. Rivest, M.E. Hellman, J.C. Anderson, et al., Responses to NIST's proposal, Commun. ACM 35(7) (1992) 41-54.

[7] Shenzhen Qianhai WeBank Co., LTD., A data processing method based on collaborative calculation, Chinese Patent CN202310238511.6, (2023-06-27).

[8] Y. Qin, W. Xu, Optimization of the ECDSA algorithm on the finite domain GF(2^m), Comput. Eng. Appl. 42(29) (2006) 136-138, 176.

[9] X. Wang, Hardware implementation design of ECDSA signature algorithm and FPGA simulation [dissertation], Xidian University, Shaanxi, (2018).

[10] K. Zhao, Z. Liu, Y. Tang, Optimized design of prime selection in DSA encryption algorithm, Sci. Technol. Inf. (2010) (33) 30, 275.

[11] L. Zhu, High-speed implementation of the hash function encryption algorithm [dissertation], Shanghai Jiao Tong University, Shanghai, (2007).

[12] F. Yu, Y. Jia, An Enhanced Lattice Attack to DSA and ECDSA Scheme, Netinfo Security 22(2) (2022) 11-20.

[13] D. Wang, Application of ECDSA in mobile e-commerce [dissertation], Northeastern University, Liaoning, (2005).

[14] S. Li, X. Dou, X. Zhang, A 3D printing plus cryptographic method based on ECDSA, Digit. Commun. World (2021) (4) 85-86.

[15] Q. Huang, Optimization of network performance in wireless sensor networks, Central South University, Hunan, (2014).

[16] S. Chen, R. Zhao, Mobile communication user authentication scheme based on DSA digital signature algorithm, In: Proc. 6th China Acad. Conf. Cryptogr. Beijing: Science Press, (2000) 129-132.

[17] R. Yang, H. Wu, J. Zhang, Application of DSA encryption algorithm in mobile phone SMS early warning, Inf. Secur. Commun. Confid. (2013) (11) 97-99.