# Advances in Digital Signature Algorithms: Performance, Security and Future Prospects

*Shuhan* Lyu*

College of Letters and Sciences, University of California Santa Barbara, 93106 Santa Barbara, America

**Abstract.** With the rise of online transactions and digital communications, digital signatures have become essential for verifying document integrity and authenticity across fields like finance, government, and healthcare. This article examines several important digital signature algorithms, analyzing their principles and performance. Each algorithm offers unique advantages: the Rivest – Shamir – Adleman (RSA) provides security through integer factorization, the Digital Signature Algorithm (DSA) through discrete logarithms, and the Elliptic Curve Digital Signature Algorithm (ECDSA) with the efficient elliptic curve cryptography. This paper also discusses recent optimizations, including the Batch and Multi-prime RSA for greater efficiency, the Batch DSA for improved verification speed, and the EdDSA for the enhanced ECDSA performance on Edwards curves. The results show that while optimizations improve computational efficiency, challenges persist, especially with potential quantum threats that could undermine these algorithms' security. Future work should explore hybrid cryptographic schemes and post-quantum cryptography to strengthen digital signature resilience in evolving digital environments.

## 1 Introduction

Digital signatures are algorithms that can authenticate the integrity of online documents. Their role has become increasingly crucial across fields, including contract processing, government paperwork handling, financial transactions, and healthcare insurance. Digital signatures are considered more secure than physical signatures as they are less prone to theft or damage. Moreover, they offer the advantage of faster transmission with access to the Internet.

A digital signature usually involves three components. A key generation algorithm that creates a pair of public and private keys; a signing algorithm that uses the user's private key and the hash value of the document to create a unique signature; and finally, a verifying algorithm that uses the public key to confirm the authenticity of the document received by generating another signature. The document's integrity is guaranteed when the two signatures match.

Rivest – Shamir – Adleman (RSA), Digital Signature Algorithm (DSA), and Elliptic Curve Digital Signature Algorithm (ECDSA) are the three main algorithms of digital

*Corresponding author: shuhanlyu@ucsb.edu

signatures. Each of them is based on a hard problem that cannot be solved efficiently with current computing power, and they vary in performance in computational complexity, time consumption, and security. The difficulty of large prime factorization guarantees the security of the RSA algorithm. If there are breakthroughs in computational power (such as quantum computers) or the discovery of new algorithms that make large number factorization easier, RSA's security would be significantly weakened. Similar to RSA, DSA is also a public-key encryption scheme. The difference is that DSA relies on the complexity of solving discrete logarithm problems. Compared to RSA, DSA, proposed by the U.S. National Institute of Standards and Technology (NIST) in 1991, has lower computational complexity, which makes it more suitable for batch signature verification. However, the key generation of DSA is less efficient, particularly in resource-constrained environments. The ECDSA generates signatures using elliptic curve cryptography or the discrete logarithm problem over finite fields. With the same security level, ECC keys are shorter and require fewer computational resources, making them suitable for mobile devices and embedded systems. However, its algorithm is more complex compared to other methods and, therefore, needs a more sophisticated algorithm. With the development of blockchain technology, some signature methods that integrate blockchain technology have also been developed. They combine blockchain's decentralized features to prevent signature forgery and support smart contracts for efficient automated processes.

## 2 Methods

### 2.1 RSA

RSA is a public key encryption that is frequently used to create digital signatures. The essential principle behind it is that it is very difficult to factor a large integer created by multiplying two prime numbers. The whole algorithm runs under the assumption that factoring such numbers cannot be done in a reasonable time under current computing power.

Since RSA is a public key encryption, the generation involves a public key and a private key. First, two large primes, p, and q, are chosen randomly; these prime numbers should be chosen randomly. And the larger the primes, the harder it is to break the encryption [1]. The key generation starts with the computation of n, which follows as $n = p \times q$, which is part of both the public key and private key. When calculating n, $\varphi(n) = (p-1) \times (q-1)$ can be used because n is the multiplication of two prime numbers. $\varphi(n)$ helps define the range for the encryption and decryption exponents. e is chosen to be co-prime with $\varphi(n)$ and also satisfying $1 < e < \varphi(n)$. d as part of the private key is calculated such that it satisfies $\varphi(n)$: $d \times e = 1 \bmod \varphi(n)$. The document will be converted into binary form and then hashed with SHA-256 to give a value of m, which will be used to generate the signature.

### 2.2 DSA

DSA is another widely used public key algorithm in the field of digital signature [2, 3]. Unlike RSA. DSA is based on the mathematical difficulty of solving the discrete logarithm problem. The DSA algorithm operates on the principle that while it's easy to compute powers in a finite group, finding the discrete logarithm (the reverse operation) is computationally infeasible.

Using DSA as document verification usually involves four steps: generating a key, generating a signature, signature verification, and key distribution.

For key generation, DSA requires the selection of a large number p as a prime, which represents the size of the group, and another number q, which is prime and divides p−1. A

number g for a cyclic group is selected, which helps in generating the public key. The private key x is a randomly determined number such that $0 < x < q$, and y as the public key is computed as $y = g^x \bmod p$. The private key x is kept secret, while y is shared publicly.

For signature generation, a random number k is selected for each signature such that $0 < k < q$, and the hash value of the document is computed. The signature consists of two values, r, and s. r is calculated as $r = (g^x \mid p) \mid q$, and s satisfies that $s = (k^{-1} \times (\text{hash}(m) + x \times r)) \bmod q$, where hash(m) is the hash of the message.

For signature verification, the recipient calculates two values, u1 and u2, based on the public key and the message's hash and checks whether $g^{u1} \times y^{u2} \bmod p$ matches the value of r. If these values are the same, the signature is verified.

### 2.3 ECDSA

ECDSA uses elliptic curve cryptography (ECC) to provide the same level of security as RSA or DSA but with significantly smaller key sizes [4, 5], which means it is especially suitable for environments with constrained resources, such as phones or embedded systems.

Like DSA, ECDSA works on the principle of the discrete log problem, but instead of working in a finite group of integers, ECDSA operates over an elliptic curve. This makes the computation more efficient while maintaining high security.

For key generation, a random private key d is selected, where d is a random integer in the range $[1, n-1]$ from the base point G derived n as its order. The public key Q is calculated as $Q = dG$, where G is a predefined point on the elliptic curve.

For signature generation, k is selected randomly at first, and then the pair (r, s) is calculated similarly to DSA but using the elliptic curve arithmetic. Specifically, r is calculated by taking the x-coordinate of the point kG, and s is calculated using the private key combined with the hash value of the message.

For signature verification, the basic logic is to manipulate the points on the elliptic curve. The verifier computes points based on the public key, the signature, and message's hash. If the calculated point matches the signature, the signature is valid.

### 2.4 Optimized RSA

The improvement of RSA includes many different aspects; in 1982, Quisquater and Conver proposed an improvement on the speed of RSA using the Chinese remainder theorem, which is called QCRSA. The improvements in RSA include batch RSA, multi-prime RSA, multipower RSA, and rebalanced RSA. The batch RSA reduces the complexity of computation during key generation by introducing a technique that can perform multiple encryption or signature generation with a single modular exponentiation calculation [6]. Classical RSA needs to calculate a new N and e for every message that needs to be encrypted. The most common algorithm for this computation is using the Chinese remainder theorem, which can reduce the large modular exponentiation operation into two small modular exponentiation operations and increase the efficiency based on the number of messages encoded. In the article Batch RSA, the author assigned n encryption exponents (e1, e2, …., en) of the messages to encrypt to a Huffman tree, which can minimize the computational path lengths. At each leaf node, the results of its two subtrees are taken, and the power is raised to the product of the exponents on the other side. The value will then be stored at the leaf node. When calculating the parent node, this value will be used to reduce the complexity.

The multi-prime RSA improves upon the original algorithm with a different approach [6]. Unlike classic RSA, multi-prime RSA constructs the N using multiple prime numbers,

which makes the algorithm even faster when applying the Chinese remainder theorem. The calculation can be divided into smaller module problems for each prime. Since more prime numbers are used to construct n, the exponents used in these calculations are reduced in size. In a 1024-bit multi-prime RSA system, each prime factor can be roughly one-third of the original modulus size compared to the 512-bit exponents in standard RSA, which directly reduces the computational procedures and significantly improves the speed. In addition, the multi-prime RSA also utilizes the Montgomery reduction to improve the efficiency of module calculation. Montgomery reduction changes how numbers are represented to make modular multiplication faster. Each value is adjusted by multiplying it with a large number R (chosen as a power of 2) before doing the main calculations. This adjustment, called the "Montgomery form," lets the algorithm perform modular multiplication without doing a slow division step. It uses bit shift to increase the efficiency.

### 2.5 Optimized DSA

One way to enhance DSA is to create keys for messages within a group of random numbers that are all relatively prime to one another in the set. This distinct property of being relatively prime enables each signature to be processed independently in a batch verification system without any disruption among them. The standard DSA verification process involves exponentiation, for each encrypted message, which can be complex and time consuming. The batch DSA utilizes addition, under q of modular exponentiation to enhance the verification process speed and simplify calculations significantly. In addition to batch DSA enhancements, for speed improvement by avoiding the computation intensive the author suggests using a masking factor d to transform the calculation process into solving. The updated algorithm computes w of utilizing the inverse algorithm for verification purposes.

### 2.6 Optimized ECDSA

EdDSA operates as a modern digital signature designed to improve ECDSA [7, 8]. Its design leverages Edwards curves, known for their efficiency, security, and simplicity, particularly in high-performance and resource-constrained environments. Compared to ECDSA, it leverages the Edwards curve, known for faster addition and doubling operations, which streamline computations and result in significantly better performance than ECDSA's Weierstrass curve. It also uses deterministic key generation, such as hashing, which enhances security by preventing issues tied to randomness.

## 3 Discussion

### 3.1 RSA

Compared to regular RSA, the Batch RSA only requires one modular exponentiation for a batch of operations. This dramatically reduces computation costs when processing multiple messages or signatures, as only multiplications per message are needed compared to O(n) in standard RSA. Therefore, the Batch RSA's performance improvement is most prominent when the number of messages that need to be encrypted is very large. This makes Batch RSA especially suitable for centralized systems, such as mainframes handling bulk transactions or digital signatures in applications requiring high throughput. Besides the improvement in efficiency, Batch RSA's security also has been improved. Performing a single root extraction for each branch prevents unnecessary exposure of the private key,

reducing security risks. However, there are still some limitations with Batch RSA. Firstly, the improvement depends largely on the size of the batch. The method loses some of its efficiency if the number of operations does not meet an optimal batch size. Also, if encryption exponents are reused across messages, there is a risk of message reconstruction by an adversary, as highlighted in Batch RSA's limitations.

Compared to Batch RSA, the multi-prime RSA has a more general improvement on efficiency since it does not depend on the number of encrypted messages, since it reduces the size of exponents by including more prime factors in the modulus N, enhancing calculation speed significantly. In addition, it also utilizes a DSP architecture optimizations, such as Montgomery reduction, which helps to achieve better results in some embedded environments. However, adding more primes potentially introduces vulnerabilities, making multi-prime RSA theoretically more susceptible to attacks, especially if the additional primes aren't sufficiently large, and this algorithm also shows better performance on specific hardware like DSP processors due to its specific implementation.

Future implementations of RSA can explore alternative reduction methods or advanced pipe-lining techniques to handle redundancy better and improve execution time [9]. It could integrate with hybrid cryptographic schemes for batch RSA, leveraging its batch processing benefits with other encryption techniques like Elliptic Curve Cryptography (ECC) for environments requiring both efficiency and high security. In addition, in order to fight against the quantum computer that might appear in the future, which can solve the factorization of large numbers easily.

### 3.2 DSA

DSA ensures message integrity and authenticity by allowing only the private key owner to create a valid signature [10]. In contrast, anyone with the corresponding public key can verify it. The advantage of the enhanced version of DSA is optimized for efficient verification in digital environments, making it suitable for applications with high signature verification demands, such as digital certificates and secure email. However, DSA demands complex mathematical computations (modular exponentiations) in generating a signature, which can be resource-intensive compared to other algorithms like RSA. It also has the limitation that its security heavily relies on secure random number generation, as inadequate randomness can lead to vulnerabilities like the repeated random number attack, potentially exposing private keys.

In the future, CC-based DSA (ECDSA) is becoming more popular due to its smaller key sizes and reduced computational load, offering enhanced security with less resource usage—ideal for constrained environments. Also, improving pseudo-random number generation will enhance DSA's security and address its vulnerability to repeated random numbers. This enhancement would make DSA more robust against attacks targeting random number weaknesses.

### 3.3 ECDSA

The advantage of ECDSA is that it performs faster addition and doubling operations on Edwards curves, which improves overall signature generation and verification speeds, especially beneficial in high-throughput applications like blockchain and IoT, it also uses deterministic hash-based signatures, which reduces vulnerabilities tied to randomness. However, as a relatively newer technology compared to ECDSA, EdDSA has less widespread adoption, particularly in established systems like Bitcoin, where ECDSA is the standard. EdDSA also requires environments to support specific Edwards curves (like

Ed25519), which might not be compatible with all existing cryptographic systems that rely on traditional elliptic curves.

In the future, as block-chain applications grow, EdDSA's performance benefits may drive its adoption over ECDSA, particularly in networks requiring and efficient signing. With more research, EdDSA's compatibility could extend beyond Edwards curves, potentially developing into variants that work within traditional elliptic curve infrastructure while maintaining EdDSA's security and performance benefits.

## 4 Conclusion

This paper examined three primary algorithms for digital signatures—RSA, DSA, and ECDSA—analyzing their foundational principles, strengths, limitations, and recent improvements. RSA relies on the difficulty of factoring large integers, making it secure but potentially vulnerable to advances in quantum computing. Improvements like Batch RSA and Multi-prime RSA reduce computational complexity and increase efficiency, especially for high-throughput applications, but they also introduce challenges related to optimal batch sizes and security with additional primes. DSA, based on the discrete logarithm problem, excels in environments requiring fast verification and batch processing; however, its dependence on secure random numbers introduces potential vulnerabilities. Optimizations such as Batch DSA and improved random number generation enhance its verification speed and reliability. ECDSA offers strong security with smaller key sizes, suitable for constrained environments. The optimized EdDSA variant, with faster point operations on Edwards curves, provides performance gains but faces challenges in adoption and compatibility with traditional elliptic curves.

Digital signature technology continues to face challenges, especially with the advent of quantum computing, which threatens the foundational security assumptions of RSA and DSA. Future directions may include exploring hybrid cryptographic schemes that combine the efficiency of RSA's batch processing with the compact, secure structure of ECC. Moreover, enhancing pseudo-random number generation methods is critical to address vulnerabilities in DSA and ECDSA-based schemes. Despite covering various improvements and optimizations, this article is limited in scope, as it primarily focuses on established algorithms and does not address emerging post-quantum cryptographic methods that may offer long-term resilience against future threats. Further research into advanced cryptographic protocols will be essential to ensuring the robustness and adaptability of digital signatures in a rapidly evolving digital landscape.

## References

1. R. L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM. **21(2)** (1978), 120-126
2. T. Pornin, Deterministic usage of the digital signature algorithm (DSA) and elliptic curve digital signature algorithm (ECDSA) (No. RFC6979) (2013)
3. P. Q. Nguyen, I. E. Shparlinski, The insecurity of the digital signature algorithm with partially known nonces. Journal of Cryptology, **15** (2002), 151-176
4. D. Johnson, A. Menezes, S. Vanstone, The elliptic curve digital signature algorithm (ECDSA). International Journal of Information Security, **1** (2001), 36-63
5. J. Doerner, Y. Kondi, E. Lee, A. Shelat, Secure two-party threshold ECDSA from ECDSA assumptions. 2018 IEEE Symposium on Security and Privacy (SP), IEEE (2018), 980-997

6. J. Grossschadl, The Chinese remainder theorem and its application in a high-speed RSA crypto chip, in Proceedings of 16th Annual Computer Security Applications Conference (ACSAC'00), IEEE (2000), 384-393
7. S. Josefsson, I. Liusvaara, Edwards-curve digital signature algorithm (EdDSA) (No. RFC8032) (2017)
8. Q. Feng, K. Yang, M. Ma, D. He, Efficient multi-party EdDSA signature with identifiable aborts and its applications to blockchain. IEEE Transactions on Information Forensics and Security, **18** (2023), 1937-1950
9. Z. Cao, L. Liu, The Practical Advantage of RSA over ECC and Pairings. Cryptology ePrint Archive (2024)
10. N. Mehibel, M. H. Hamadouche, Efficient and secure digital signature algorithm (DSA). Emirates Journal for Engineering Research, **28(2)**, 3 (2023)