A Comprehensive Study of Digital Signatures: Algorithms, Challenges and Future Prospects

Jingkun Xu*

College of Business Administration, California State Polytechnic University – Pomona, 91709 Pomona, United States

Abstract. This article examines digital signature as a critical security measure for authentication and verification in electronic transactions, distinguishing them from simpler e-signatures by their use of advanced cryptographic techniques. Digital signature leverage asymmetric cryptography to provide higher security, with various standardised algorithms, such as the Digital Signature Algorithm (DSA) and Rivest-Shamir-Adleman (RSA), forming the foundation of secure systems today. The paper outlines the evolution of digital and electronic signature, from early telegraph-based approvals to the modern applications that facilitate secure digital contracts and communications. Key digital signature algorithms are analysed in detail, highlighting their strengths and weaknesses, including RSA, DSA, Elliptic Curve Digital Signature Algorithm (ECDSA), and Edwards-Curve Digital Signature Algorithm (EdDSA). Additionally, the article addresses the technical and implementation challenges of digital signature algorithms, such as high computational demands, complexities in key management, and difficulties in secure implementation. It concludes with an exploration of future trends, such as the development of efficient and quantum-resistant algorithms, improvements in cryptographic hardware, and new strategies to simplify secure key management. This comprehensive overview provides insights into both current digital signature practices and emerging solutions poised to enhance security and efficiency in an evolving digital landscape.

1 Introduction

Digital signature is one of the essential components that provide verification and permission under certain circumstances [1-3]. The concept of digital signature is similar or expresses signature's uniqueness by connecting to electronic signature. Basically, the e-signature, in some way, is part of the digital signature creation process. The digital signature uses cryptographic methods and asymmetric cryptography to sign data and provide authentication. Digital signature is important nowadays, and it is totally different from the electronic signature, or the e-signature because it provides a higher level of security, whereas the e-signature itself provides no level of security. When the e-signature is connected with digital signature or appears as a specific and iconic example of digital signature, e.g. Signature

* Corresponding author : jingkunxu@cpp.edu

appears in certain e-signature software applications; it can provide some level of security because the signature has been hashed based on the encryption within the software.

At first, the first e-signature that was accepted by the government as the handwritten one was via telegraph, then it was on a fax machine. In 2000, ESIGN Act was added into law, to legally accept that e-signature is the same as the handwritten one [4]. The acceptance of E-signature accelerated the development of digital signature applications, but it also expands more possibilities on digital signature application because the developer can add e-signature into the software. For example, the first lotus notes 1.0 was designed and published in 1989, way before the Act was added into law [4]. Since the application of digital signature was published a long time ago, the algorithms were also set up in 1982. There are two standardised Digital signature algorithms: Digital Signature Algorithm and Rivest Shamir Adleman. Those two algorithms were the most basic ones, and every other algorithm was created based on the platform these two algorithms provided. Many work-related environments adopt at least one electronic signature application from software developers e.g. the signature of the manager on the PDF file of a banking contract., and the idea of electronic signature is not only 'invading' the workplace, but it also appears more often on a daily basis e.g. the signatures of the tenants and the landlord on a leasing contract. In 2023,

The remainder of the paper is organised as follows. First, a detailed description of the two standardised algorithms would be discussed in the next section, along with some variant of algorithms e.g. different encryption but the same algorithms. Then, the advantages and disadvantages of every algorithm, including the variants, would be present in section 3, along with detailed comparison between algorithms and variants. Also, some challenges and obstacles that digital signature technology must overcome would be included in section 3, along with the proposed plan after the difficulties are resolved. Finally, the summary will be in section 4, along with the conclusion that was based on the information provided in the main content.

2 Method

2.1 Standardized algorithms

2.1.1 DSA

Digital Signature Algorithm is a Federal Information Processing Standard, and a public-key crypto-system that works for digital signatures. Undoubtedly, it was the most popular encryption method since it was accepted by the government, and it can be used as a legal certification. It provides an appropriate solution for applications requiring digital signatures [4]. It is part of the Digital Signature Standard (DSS) and is approved by the National Institute of Standards and Technology (NIST), establishing it as a trusted option for government and official applications. DSA can also be more efficient than RSA in certain signing operations, making it suitable for scenarios requiring rapid transaction processing. Additionally, it often requires smaller key sizes than RSA for a similar level of security, which can be beneficial for systems needing to optimize resources. Despite these strengths, DSA has limitations. Its verification process tends to be slower than RSA, which may be a disadvantage in time-sensitive applications. DSA also relies on high-quality random numbers for each signature; inadequate randomness can lead to vulnerabilities. Furthermore, DSA lacks the broad platform support that RSA enjoys, potentially limiting its integration in certain environments.

2.1.2 RSA

Rivest Shamir Adleman, cipher algorithm that came out before DSA but also widely used since it was the first published algorithm that's about digital signatures [5]. It was commonly used in the circumstances where security is concerned, for example, it was used when users need to login into their banking accounts, the login sessions. Also, it can be used by email company to ensure privacy and maintain security. It is highly compatible with digital certificate systems like X.509 and Public Key Infrastructure (PKI), which enhances its interoperability in secure communications. RSA also offers flexibility with various key sizes, such as 2048-bit and 3072-bit, allowing it to meet different security requirements. However, RSA's performance can be a drawback, as it is slower in signing and verification operations compared to some newer algorithms, particularly with larger key sizes. Additionally, RSA requires substantial key sizes to achieve high security, which can strain storage and computational resources, making it less efficient for devices with limited capacity. Looking to the future, RSA may also be vulnerable to quantum computing advancements, which could compromise its security unless much larger key sizes are adopted.

2.1.3 EdDSA

A variation of Schnorr's signature method using (potentially twisted) Edwards curves is called the Edwards-curve Digital Signature Algorithm (EdDSA) [6]. EdDSA has excellent performance across a range of platforms; each signature does not need to utilize a distinct random number; It can withstand side-channel attacks better; For Ed25519 and Ed448 EdDSA employs short public keys (32 or 57 bytes) and signatures (64 or 114 bytes), respectively; the formulas are "complete" meaning they work for any point on the curve. This eliminates EdDSA's requirement to carry out costly point validation on unreliable public values. Because EdDSA offers collision robustness, hash-function clashes do not disrupt this system (this is only true for PureEdDSA).

2.1.4 ECDSA

The Elliptic Curve Digital Signature Algorithm (ECDSA) provides high security with smaller key sizes, reducing storage and bandwidth requirements. This efficiency, combined with smaller key sizes, makes ECDSA especially appealing for mobile and IoT applications where resources are constrained [7]. While not fully quantum-resistant, ECDSA is more adaptable than RSA to post-quantum cryptography developments. Nonetheless, implementing ECDSA can be complex due to the mathematical intricacies of elliptic curve cryptography, which increases the risk of implementation errors. Patent restrictions on certain elliptic curve methods can also create licensing issues for some users. Additionally, ECDSA's performance can vary widely depending on the selected curve and available hardware support, making performance unpredictable in some cases.

2.2 Optimized algorithms

2.2.1 Schnorr signature

It was taken into consideration as an ECDSA substitute. An effective technique that may provide brief signatures without compromising security is Schnorr's digital signature algorithm [8]. It is among the first signatures whose security is predicated on the discrete logarithm problem's complexity.

2.2.2 ElGamal encryption

Another available public key encryption scheme to use is ElGamal. This encryption is slightly different than other ones because it was mixed with DSA, variant Schnorr signatures and Pointcheval-stern signature algorithm. The ElGamal based digital signature scheme is good at securing the document signing process and providing persuasive verification results since the randomness of some numbers' values are introduced and it makes the signing process hard to attack.

3 Discussion

3.1 Limitation and challenges

Digital signature algorithms, such as RSA, can demand substantial processing power, especially at larger key sizes, making them inefficient for devices with limited resources, such as IoT or mobile devices. Additionally, algorithms with large key sizes tend to generate large signatures, consuming more memory and bandwidth, which can hinder performance on constrained networks and devices. There is another challenge when the implementation come in. Implementing digital signature algorithms correctly can be complex, and mistakes in implementation—such as insufficient side-channel resistance or improper key handling—can introduce vulnerabilities, even if the algorithm itself is secure. Testing cryptographic implementations thoroughly to ensure security can also be difficult, particularly across large and diverse software ecosystems. Managing keys securely is essential in digital signature systems. Secure storage of private keys is crucial, as a compromised private key renders the digital signature unreliable. Managing key pairs for large numbers of users is challenging, especially in large organizations, where consistent key security can be difficult to maintain. Additionally, when a key is compromised or needs to be replaced, timely revocation and reissuing of keys can be complex in real-world applications, potentially delaying access to updated information.

3.2 Future prospect

Efforts to develop more efficient algorithms, such as elliptic curve-based methods and future post-quantum algorithms, are likely to ease the computational load on devices. Algorithms like ECDSA and EdDSA are already popular due to their efficiency on resource-constrained devices. Continued advancements in cryptographic hardware, including specialized coprocessors, will further support efficient processing on IoT and mobile platforms [8]. Additionally, the National Institute of Standards and Technology (NIST) is working to standardize lightweight cryptographic algorithms for resource-constrained devices, which aim to provide robust security with minimal processing power, benefiting IoT and other low-power applications [8].

To reduce the complexity of cryptographic algorithms and minimise the risk of implementation errors, some efforts are focusing on simpler curve types in elliptic curve cryptography. Automated cryptographic tools, including formally verified libraries and fuzzing tools [9, 10], are also becoming more prevalent, helping developers identify vulnerabilities early in the process. New algorithms designed with built-in side-channel resistance aim to reduce susceptibility to attacks like timing and power analysis, and standardisation organisations are increasingly emphasising side-channel-resistant implementations.

4 Conclusion

This paper provides a comprehensive analysis of digital signature algorithms, including DSA, RSA, EdDSA, and ECDSA, highlighting their strengths, limitations, and specific applications. Furthermore, the paper discussed key challenges in digital signature implementation, such as computational demands, memory constraints, and security vulnerabilities in real-world applications. Proposed solutions include optimizing algorithms for resource-constrained devices and incorporating side-channel resistance features. Additionally, the paper explores future directions, such as the development of post-quantum cryptographic algorithms and improvements in implementation tools, which aim to enhance the security and efficiency of digital signature systems. Through these contributions, the paper provides valuable insights into advancing digital signature technology for secure and efficient authentication across diverse platforms.

References

1. R. Kaur, A. Kaur, Digital Signature, in Proceedings of 2012 International Conference on Computing Sciences, IEEE (2012), 295-301
2. C. Nist, The digital signature standard. Communications of the ACM, **35**(7) (1992), 36-40
3. M. Pooja, M. Yadav, Digital signature. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), **3**(6) (2018), 71-75
4. D. Naccache, D. M'RaÏhi, S. Vaudenay & D. Raphaeli, Can D.S.A. be improved? — Complexity trade-offs with the digital signature standard —. In: A. De Santis (Ed.), Advances in Cryptology — EUROCRYPT'94, Lecture Notes in Computer Science, **950**, Springer, Berlin, Heidelberg (1995)
5. D. Boneh, Twenty Years of Attacks on the RSA Cryptosystem. Notices of the AMS, **46** (1999), 203-213
6. L. Lawrence, R. Shreelekshmi, Edwards curve digital signature algorithm for video integrity verification on blockchain framework. Science & Justice, **64**(4) (2024), 367-376
7. National Institute of Standards and Technology, Digital Signature (DSS). Department of Commerce, Washington, D.C., Federal Information Processing Standards Publications (FIPS) NIST FIPS (2023), 186-5
8. M. Battagliola, A. Galli, R. Longo, A. Meneghetti, A Provably-Unforgeable Threshold Schnorr Signature With an Offline Recovery Party, In Ceur Workshop Proceedings, **3166** (2022), 60-76
9. J. K. Zinzindohoué, K. Bhargavan, J. Protzenko, B. Beurdouche, HACL*: A verified modern cryptographic library, in Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (2017), 1789-1806
10. M. Ammann, L. Hirschi, S. Kremer, Dy fuzzing: formal Dolev-Yao models meet cryptographic protocol fuzz testing, in Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), IEEE (2024), 1481-1499