A Comparative Performance Analysis of Digital Signature Schemes in Passkey-Based Authentication Systems

Noel Christoffel Simbolon
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
hi@noelsimbolon.com

Rinaldi Munir
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia
rinaldi@informatika.org

***Abstract*— Passkey-based authentication systems, designed to replace traditional password-based methods, fundamentally rely on digital signature schemes. The choice of scheme, however, impacts overall system performance. This research presents a comparative analysis of three widely used digital signature schemes: RSASSA-PKCS1-v1_5, ECDSA, and EdDSA at an equivalent security level of approximately 128 bits. Performance is evaluated based on space cost (public key, private key, and signature sizes) and time cost (key generation, signing, and verification times). Experiments were conducted across three distinct environments: a cloud server, a laptop, and an Android mobile device. Additionally, this study examines the influence of challenge sizes ranging from 16 to 256 bytes on scheme performance. The results indicate that both ECDSA and EdDSA are consistently more efficient than RSASSA-PKCS1-v1_5. Specifically, EdDSA provides the fastest signing operations in all environments, while ECDSA excels in key generation. For verification, EdDSA leads on cloud and laptop platforms, whereas ECDSA is superior on the Android device. It was also determined that variations in challenge size do not have a significant effect on the signing or verification times of any scheme. Consequently, the recommended order of preference for implementing these schemes is EdDSA, followed by ECDSA, and lastly RSASSA-PKCS1-v1_5.**

Keywords—performance, digital signature schemes, passkey-based authentication systems, RSASSA-PKCS1-v1_5, ECDSA, EdDSA

I. Introduction

The proliferation of online services has made robust user authentication a critical security requirement. While password-based systems remain prevalent, they are fraught with risks, including phishing, credential stuffing, and poor user practices [1], [2], [3], which motivate the development of more secure alternatives. Passkey-based authentication, built upon the FIDO2 standards (WebAuthn and CTAP) [4], [5], has emerged as a leading replacement. This paradigm leverages public-key cryptography, specifically digital signature schemes [6], [7], [8], to provide strong, phishing-resistant authentication without relying on shared secrets.

In passkey systems, the choice of a digital signature scheme directly impacts performance, scalability, and user experience, especially on resource-constrained devices like smartphones and hardware security keys. The WebAuthn specification recommends several algorithms, with RSASSA-PKCS1-v1_5, the Elliptic Curve Digital Signature Algorithm (ECDSA), and the Edwards-curve Digital Signature Algorithm (EdDSA) being the most commonly adopted [4].

Previous research has compared the performance of these schemes in different contexts. Studies have analyzed the benefits of Elliptic Curve Cryptography (ECC) over RSA in SSL/TLS handshakes [9] and within Named Data Networking (NDN) architectures [10]. Another analysis compared EdDSA and ECDSA for DNSSEC applications [11]. However, a direct, comprehensive performance comparison of all three major schemes (RSASSA-PKCS1-v1_5, ECDSA, and EdDSA) within the specific context of passkey-based authentication systems remains a research gap.

This paper bridges that gap by presenting a comparative performance analysis of these three digital signature schemes at an equivalent security level of approximately 128 bits. We evaluate their efficiency based on space cost (public key, private key, and signature sizes) and time cost (key generation, signing, and verification times). Furthermore, we investigate the impact of varying cryptographic challenge sizes (16 to 256 bytes) on performance. The experiments are conducted across three distinct environments representing the typical actors in a passkey ceremony: a cloud server (relying party), a laptop (client), and an Android smartphone (authenticator). Based on our findings, we provide recommendations for selecting the optimal scheme in passkey implementations.

II. Methodology

A. Scheme Configuration

All three signature schemes were configured to provide a comparable security strength of approximately 128 bits, a standard level for modern applications . This ensures that performance differences are attributable to the algorithms themselves rather than varying levels of security. The specific configurations were:

1. RSASSA-PKCS1-v1_5: Utilized a 3072-bit RSA modulus with the SHA-256 hash function.
2. ECDSA: Implemented using the NIST P-256 elliptic curve with the SHA-256 hash function.
3. EdDSA: Implemented as Ed25519, which uses the Edwards25519 curve and the SHA-512 hash function.

All cryptographic operations were performed using the standard crypto package in the Go programming language (version 1.24.4), which is a trusted and widely used library in production systems.

B. Performance Metrics

Performance was evaluated based on two categories of metrics:

1. Space Cost: The theoretical storage size required for cryptographic artifacts, measured in bytes. This includes public key size, private key size, and signature size. These metrics are crucial for evaluating storage overhead on authenticators and data transmission loads.
2. Time Cost: The computational time required for core cryptographic operations, measured in milliseconds (ms). This includes key generation time, signing time, and verification time. Each operation was executed 1 to 1000 times per batch, depending on the duration per one operation execution. The experiment was repeated for 100 batches. The median time from these 100 batches was used for analysis to minimize the impact of outliers.

C. Experimental Environments

Tests were conducted on three different hardware platforms representing the relying party, client, and authenticator.

1. Cloud Server (Google Compute Engine): A c4-highcpu-2 instance with an Intel Xeon (5th gen) CPU and 4GB RAM, running Debian 12. This represents the backend infrastructure of a relying party.
2. Laptop: A device with an AMD Ryzen 5 4600HS CPU and 16GB RAM, running Windows 11. This represents a typical end-user client platform.
3. Android Device: A Samsung device with an Exynos 7885 octa-core CPU and 6GB RAM, running Android 9. This represents a resource-constrained platform authenticator.

D. Challenge Size Analysis

In a passkey authentication ceremony, the relying party sends a cryptographic challenge to be signed by the authenticator. To analyze the impact of the data payload on performance, the signing and verification operations were tested with challenge sizes varying from 16 bytes (the minimum recommended by WebAuthn) up to 256 bytes.

III. Results and Evaluation

The experimental results highlight significant performance disparities between the RSA-based scheme and the two elliptic-curve-based schemes.

A. Space Cost Analysis

As shown in Table I and Fig. 1, the space costs showed that ECC-based schemes are substantially more efficient.

1. Key Sizes: RSASSA-PKCS1-v1_5 required the largest storage, with a private key size of 768 bytes and a public key size of 387 bytes. In contrast, both ECDSA and EdDSA required only 32 bytes for their private keys. For public keys, EdDSA was the most compact at 32 bytes, followed by ECDSA at 64 bytes.
2. Signature Size: The signature produced by RSASSA-PKCS1-v1_5 was 384 bytes, whereas both ECDSA and EdDSA produced much smaller signatures of 64 bytes.

These results demonstrate that EdDSA and ECDSA offer a significant advantage in minimizing data storage on authenticators and reducing bandwidth consumption during authentication ceremonies.

TABLE I. Breakdown of Space Cost (bytes)

Space Cost Metric	Digital Signature Scheme		
	RSASSA-PKCS-v1_5	ECDSA	EdDSA
Public key size	387	64	32
Private key size	768	32	32
Signature size	384	64	64

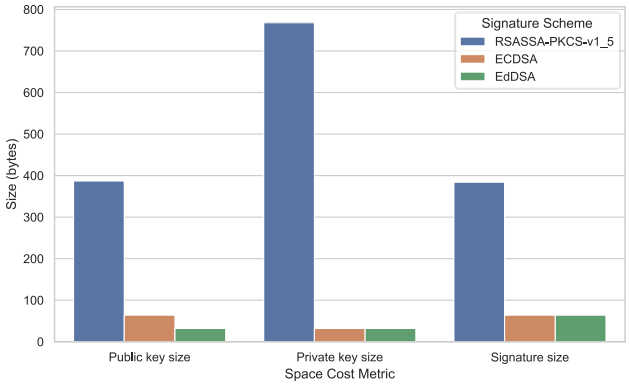

Fig. 1. Comparison of space cost

B. Time Cost Analysis

The time cost measurements reveal the computational efficiency of each scheme for the primary operations in the passkey lifecycle. The median results from 100 batches are summarized below.

1) Key Generation Time: Key generation occurs once during the initial registration. As shown in Table II, ECDSA was the fastest in all environments, followed closely by EdDSA. RSASSA-PKCS1-v1_5 was orders of magnitude slower, taking over 880 ms on the laptop compared to just 0.018 ms for ECDSA.

TABLE II. Median Key Generation Time (ms)

Environment	Digital Signature Scheme		
	RSASSA-PKCS-v1_5	ECDSA	EdDSA
Cloud Server	596.186	0.016	0.018
Laptop	882.691	0.018	0.025
Android	2925.312	0.082	0.122

2) Signing Time: The signing operation is performed by the authenticator during every login. As seen in Table III, EdDSA was the fastest signing algorithm across all three platforms. Its performance advantage was particularly notable on the resource-constrained Android device, where it was nearly three times faster than ECDSA and over 135 times faster than RSASSA-PKCS1-v1_5. A fast signing time is critical for a responsive user experience.

TABLE III. Median Signing Time (ms)

Environment	Digital Signature Scheme		
	RSASSA-PKCS-v1_5	ECDSA	EdDSA

Cloud Server	2.712	0.038	0.022
Laptop	3.915	0.042	0.031
Android	19.001	0.399	0.140

3) Verification Time: Verification is executed by the relying party's server to validate a login attempt. Table IV shows that on the high-performance cloud and laptop environments, EdDSA offered the fastest verification. However, an interesting reversal occurred on the Android device, where ECDSA was the fastest. RSASSA-PKCS1-v1_5, while slower than the ECC schemes, had a verification time that was significantly faster than its signing time, making it more viable on the server side than on the client side.

TABLE IV. MEDIAN VERIFICATION TIME (MS)

Environment	Digital Signature Scheme		
	RSASSA-PKCS-v1_5	ECDSA	EdDSA
Cloud Server	0.151	0.077	0.050
Laptop	0.207	0.091	0.075
Android	1.455	0.310	0.354

C. Impact of Challenge Size

Our analysis showed that varying the challenge size from 16 to 256 bytes had no significant impact on the signing or verification times for any of the three schemes. As shown in Fig. 2 and 3, the performance remained stable across the tested range in all environments.

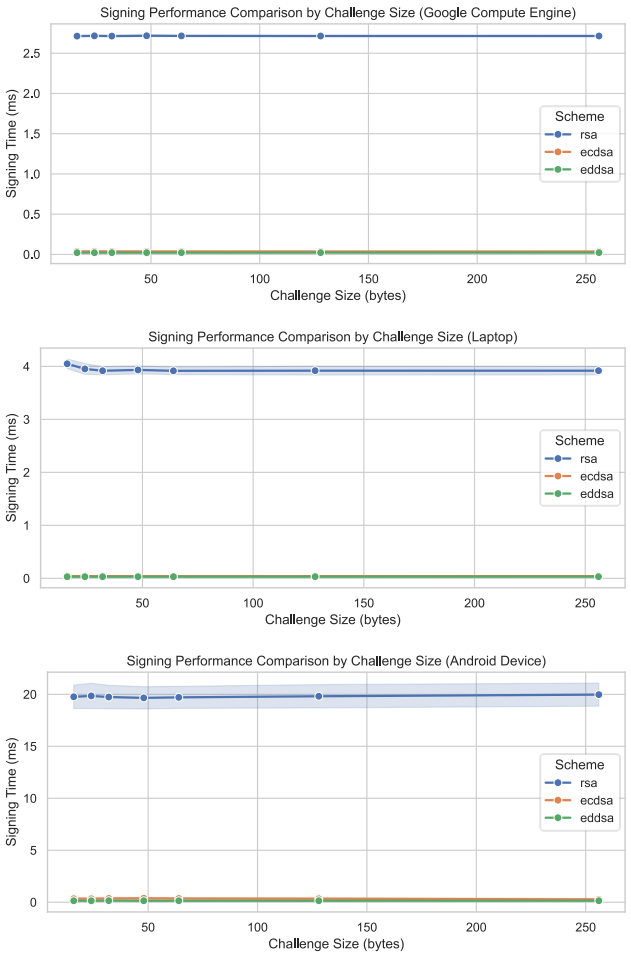

Fig. 2. The effect of challenge size on signing time

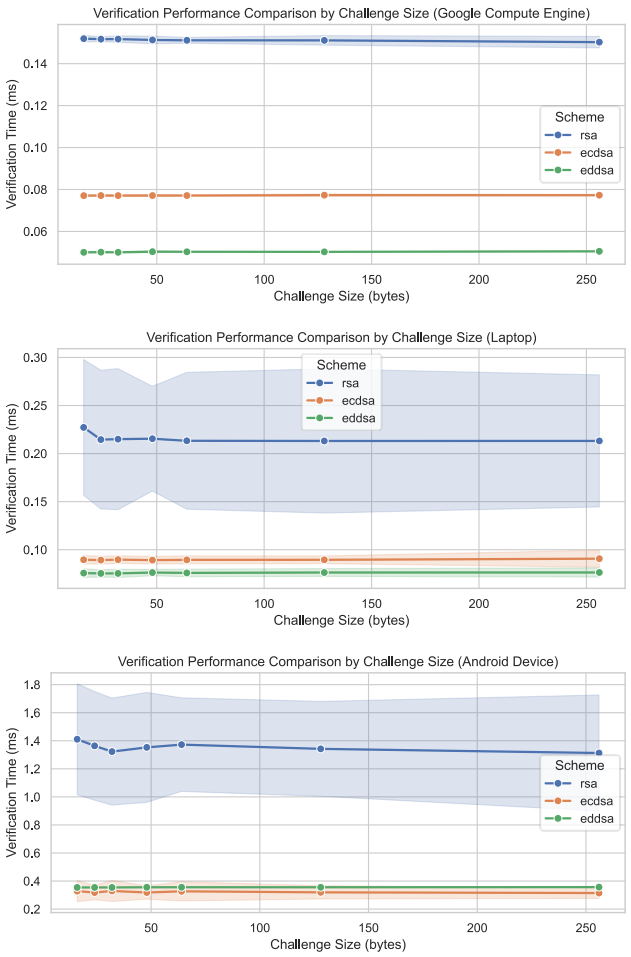

Fig. 3. The effect of challenge size on verification time

This is because the primary computational cost in digital signature schemes comes from the cryptographic calculations (e.g., modular exponentiation or elliptic curve point multiplication), while the initial hashing of the input message is extremely fast and its duration is negligible in comparison. This finding implies that developers can select a challenge size based on security requirements (e.g., ensuring sufficient entropy) without worrying about performance degradation.

IV. CONCLUSION AND FUTURE WORK

This study provides a comprehensive performance comparison of RSASSA-PKCS1-v1_5, ECDSA, and EdDSA for passkey-based authentication systems. Our findings lead to the following conclusions:

1. Overall Performance: The elliptic-curve-based schemes, ECDSA and EdDSA, are decisively superior to RSASSA-PKCS1-v1_5 in terms of both space and time costs. They produce smaller keys and signatures and perform cryptographic operations significantly faster.
2. Scheme Recommendations: Based on the results, we recommend the following order of preference for implementation in passkey systems:
    - 1st - EdDSA: It offers the fastest signing performance across all platforms, which is critical for a responsive user login experience. It also provides excellent

verification speed and compact key/signature sizes.

- 2nd - ECDSA: A very strong alternative, offering the fastest key generation and competitive signing/verification times. Its superior verification speed on Android devices is a notable advantage for certain peer-to-peer or decentralized use cases.
- 3rd - RSASSA-PKCS1-v1_5: While a well-established algorithm, its poor performance in signing and key generation, combined with its large space requirements, makes it the least suitable choice for modern, resource-aware passkey systems.

3. Challenge Size: The size of the cryptographic challenge (in the 16–256 byte range) does not materially affect performance. Therefore, a 16-byte challenge, which provides 128 bits of entropy, is sufficient and efficient for security against replay attacks.
4. For future work, this analysis could be extended to include other performance metrics like energy consumption, which is highly relevant for mobile authenticators. Additionally, as the threat of quantum computing grows, a similar comparative analysis of post-quantum digital signature schemes standardized by NIST would be a valuable contribution to ensuring the long-term security of passkey authentication.

REFERENCES

[1] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Montréal Québec Canada: ACM, Apr. 2006, pp. 581–590. doi: 10.1145/1124772.1124861.

[2] M. H. Nguyen Ba, J. Bennett, M. Gallagher, and S. Bhunia, "A Case Study of Credential Stuffing Attack: Canva Data Breach," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA: IEEE, Dec. 2021, pp. 735–740. doi: 10.1109/CSCI54926.2021.00187.

[3] D. Poza, "What Is Password Spraying? How to Stop Password Spraying Attacks." [Online]. Available: https://auth0.com/blog/what-is-password-spraying-how-to-stop-password-spraying-attacks/

[4] T. Cappalli, M. B. Jones, A. Kumar, E. Lundberg, and M. Miller, Eds., "Web Authentication: An API for accessing Public Key Credentials Level 3." Web Authentication Working Group, Jan. 27, 2025. [Online]. Available: https://www.w3.org/TR/2025/WD-webauthn-3-20250127/

[5] J. Bradley, M. B. Jones, A. Kumar, R. Lindemann, J. Verrept, and D. Waite, Eds., "Client to Authenticator Protocol (CTAP)." FIDO Alliance, Feb. 28, 2025. [Online]. Available: https://fidoalliance.org/specs/fido-v2.2-ps-20250228/fido-client-to-authenticator-protocol-v2.2-ps-20250228.html

[6] S. Goldwasser, S. Micali, and R. L. Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, Apr. 1988, doi: 10.1137/0217017.

[7] J. Katz and Y. Lindell, *Introduction to modern cryptography*, Third edition. in Chapman & Hall/CRC cryptography and network security series. Boca Raton, FL: CRC Press, 2021.

[8] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. in CRC Press series on discrete mathematics and its applications. Boca Raton: CRC Press, 1997.

[9] V. Gupta, S. Gupta, S. Chang, and D. Stebila, "Performance analysis of elliptic curve cryptography for SSL," in *Proceedings of the 1st ACM workshop on Wireless security*, Atlanta GA USA: ACM, Sept. 2002, pp. 87–94. doi: 10.1145/570681.570691.

[10] A. I. Ali, "Comparison and Evaluation of Digital Signature Schemes Employed in NDN Network," *IJESA*, vol. 5, no. 2, pp. 15–29, June 2015, doi: 10.5121/ijesa.2015.5202.

[11] J. J. Yu, "Is there a case to prefer Ed25519 over ECDSA P-256 for DNSSEC?" July 2017. [Online]. Available: http://essay.utwente.nl/75354/