Received: December 7, 2023. Revised: January 23, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Efficient Multi-signature and QR Code Integration for Document Authentication Using EdDSA-based Algorithm

Mike Yuliana[1]* **Wildan Dharma Walidaniy[1]**

[1]*Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya, 60111, Indonesia*
* Corresponding author's Email: mieke@pens.ac.id

Abstract: The previous global pandemic has shifted many of these proposed needs to remote interactions, including document authentication. The solution for remote document authentication is the use of digital signatures. However, in reality, most implementations are inefficient and even weaken already secure methods. This study presents a novel efficient digital signature method based on a multi-signature method that incorporates the edwards-curve digital signature algorithm (EdDSA) algorithm, a quick-response (QR) code, and the cryptographically secure pseudorandom number generator (CSPRNG) for key generation. This study assessed the performance efficiency of this method by running it on a large number of executions in terms of generation, signature generation, and signature generation time for three signers and a verifier. The results show that this proposed method outperforms all the existing methods that this study compared across all parameters. In key generation performance assessments, this proposed method stands out by employing CSPRNG that already been proven for its efficiency and security for cryptographic use. The usage of the multi-signature method in the EdDSA algorithm has made this proposed method superior in signature generation and signature verification performance assessment. On paper, EdDSA is the latest algorithm that has surpassed its predecessors regarding security and efficiency. By using the multi-signature method, this study further improves the signature generation and signature verification efficiency as signatures only need to be generated and verified once to represent all signers. With the performance assessment done on this proposed device, this proposed method has an average improvement of 76.27% across all parameters against the existing method. Additionally, using QR codes in the method facilitates real-life signature verification by simplifying the scanning process for verifiers. This work provides a secure and efficient solution to document file authentication.

Keywords: Document authentication, Digital signature, Multi-signature, CSPRNG, EdDSA, QR code.

1. Introduction

The global pandemic over the past three years has necessitated interactions without physical contact and remote engagements. One notable consequence of this phenomenon is the need for signing and verifying documents. A signed document signifies its approval by the relevant parties. In traditional practice, documents are signed in writing, often leading to the possibility of physical contact. In addition, the traditional signing process is inefficient as documents must be physically delivered to the signer. Facing new situations during the pandemic has prompted a shift to digital methods, such as the widespread adoption of digital signatures, especially in academic institutions [1].

Despite the widespread use of digital signatures, their implementation has not been precise in practice. There is a prevailing belief among some individuals that a digital signature is simply a scanned image or text and QR code representation of a handwritten signature [2]. Forgery is a significant risk associated with such models. The inconsistency and susceptibility to fabrication of these items stem from the fact that their creation and verification processes both rely on human visual perception [3]. Real digital signatures require cryptographic methods within them, moving beyond the representation of traditional handwritten signatures.

The essence of a digital signature lies in the cryptographic algorithms used. These algorithms safeguard the authenticity and integrity of the signed documents. To accomplish this, the document is encoded in a distinct and difficult-to-alter format (hash) using intricate mathematical operations [4].

To better understand how digital signatures work, an understanding of cryptographic techniques is necessary. Simply put, cryptography, which derives its name from the Greek for "secret writing," is a method for generating secret messages. This technique ensures that messages or information can be securely transmitted without being understood by third parties [5]. There are four main goals of cryptography [6]: 1) Confidentiality: ensuring data remains safe and private from unauthorized access; 2) Integrity: ensuring data is safe from modification by unauthorized parties; 3) Authentication: ensuring that the sender or origin of data information is from an official source; 4) Non-repudiation: ensuring that the sender of data information cannot deny their involvement in the transmission.

In a general sense, cryptography can be categorized into two distinct types: asymmetric cryptography and symmetric cryptography [7]. Symmetric cryptography uses the same key for both encryption and decryption processes. Therefore, the sender and receiver of secret data must have the same information about the key used. In its implementation, the channel used to share key information must be secure. Examples of symmetric cryptography include data encryption standard (DES), Blowfish, and advanced encryption standard (AES). On the other hand, asymmetric cryptography uses different keys for the encryption and decryption processes. These keys are the private key and the public key. The private key is kept secret, while the public key is openly shared. digital signature algorithm (DSA), Rivest-Shamir-Adleman (RSA), and elliptic curve cryptography (ECC) are examples of asymmetric cryptography [8].

Digital signatures themselves are part of asymmetric cryptography. As explained earlier, asymmetric cryptography employs two keys. Since the digital signature aims for authentication, the private key of the signer is used to create the digital signature [9]. The public key of the signer is then used for verification. The hash function also plays a crucial role in the digital signature. The hash function creates a digest of the document before signing to ensure the document's integrity [10]. The document's digest is checked during the verification process. The digest embedded in the digital signature is obtained by decrypting it using the public key of the signer. If the document's digest during verification has the same value as the digest in the digital signature, it means the document has not been modified and can be considered authentic. The public key used also allows verification of whether the signature on the document is indeed from an official entity because the public key is a unique code for each individual and is always paired with the privately held private key [11].

RSA, one of the first algorithms used in digital signature systems, was introduced in 1977 [12]. The RSA algorithm is based on the mathematical problem of logarithms, creating a one-way function that is easy to calculate the result of an input but difficult to reverse (compute the input value). DSA, which was introduced in 1993, also uses discrete logarithm problems in its method but uses a different mathematical approach. ECC, an algorithm renowned for its security and efficiency, was introduced more recently. ECC is based on the elliptic curve discrete logarithm (ECDLP), allowing the use of smaller key sizes while maintaining security equivalent to its predecessors.

In real-world conditions, a document often needs to be signed by several entities before it is declared valid. In digital signatures, such a method is referred to as a multi-party signature. Unlike common methods where one entity generates one signature, a multi-party signature creates one digital signature from several entities [13]. There have been many studies attempting to create secure and efficient multi-signatures. For example, the latest study by Shankar et al. [14] proposed a method where biometric credentials such as Identity (ID) numbers and fingerprints are hashed to generate a private key. Then, the private keys of each user are used to generate a master private key used to form a digital signature with the EdDSA, a family of ECC. However, this method poses security vulnerabilities. Because of the additional non-random information to generate the private key, third parties can more easily attack the system. This key generation method is also considered inefficient due to unnecessary pre-processing.

This proposed method aims to tackle this issue by using the dedicated random key generation process by using the CSPRNG algorithm so that the quality of randomness cannot be distinguished from pure random [15]. Because of this, the security issue can be fixed, so the attacker cannot predict how the private key is generated. Using CSPRNG as key generation also addresses the efficiency issue by eliminating unnecessary preprocessing with existing methods. Additionally, this study proposes to implement QR codes in the method to further

Algorithm 1: Digital signature	
1	**Key Generation:** - Alice generates private key K_{pr} using a random generator and its pair public key K_{pb} - Alice shares the K_{pb} publicly and keeps the K_{pr} privately
2	**Signature Generation:** - Alice creates a digest of message m using hash function $H(m)$ - Alice encrypts the digest $H(m)$ using cryptographic function C along with private key K_{pr}. The output is signature S_m that obtained by: $S_m = C(K_{pr}, H(m))$ - Signed message m_{signed} is formed by concatenating message m and signature S_m: $m_{signed} = m \parallel S_m$ - $m_{signed}$is ready to be sent to verifier
3	**Signature Verification:** - Bob receives the m_{signed} - Bob calculates his digest from the message using the same hash function $H(m)$ - Bob decrypts the received signature using decryption function D. The result is decrypted message D_{S_m} $D_{S_m} = D(K_{pb}, S_m)$ - Bob compares the decrypted message D_{S_m}that contains the digest with the digest from his own received message $H(m)$ - If D_{S_m} equal to $H(m)$, the message is valid

improve efficiency in the signature verification process. With QR codes, signatures can be easily read by verifiers.

This study's subsequent sections are as follows. The literature review of the current methodology is detailed in section 2. The proposed method is described in section 3. The study's evaluation and analysis are presented in section 4. Finally, the conclusions of this study are in section 5.

2. Literature study

This section discusses the fundamental aspects of existing digital signature method models. This section covers cryptographic concepts such as hash, digital signatures, ECC, multi-signature schemes, and a comparison with existing methods.

2.1 Hash

Hash functions are one of the most important aspects of authentication systems, especially digital signatures. Hash is used to produce a fixed-length unique string using a one-way function that ensures authenticity [16]. The string commonly referred to as "digest" is utilized to compare messages obtained from another source with those obtained during the decryption procedure [17]. Secure hash algorithm (SHA)-2, specifically SHA-256, was used in the existing method to generate the private key of the system [18]. SHA-2 is a family of hash algorithms developed by the National Institute of Standards and Technology (NIST) in 2002 to update SHA-1 [19]. The family includes common hash algorithms such as SHA-256, SHA-384, and SHA-512. SHA-256 produces a digest value that is 256 bits long for each input length, making it an appropriate choice for generating the private key of the EdDSA algorithm which requires a key that is 32 bytes (256 bits) long.

2.2 Digital signature

The main foundation of existing methods is digital signatures. Based on encryption methods in asymmetric cryptography, digital signatures ensure the authenticity of documents and provide proof that the signer is genuine [20]. In general, digital signature algorithms involve three main operations: key generation, signature generation, and signature verification. Algorithm 1 below provides an example of how digital signatures generally work, with Alice acting as the signer and Bob as the verifier.

2.3 ECC

In 1985, a new generation of public key cryptography was introduced, namely ECC developed by Victor Miller and Neal Koblitz [21]. ECC was a significant improvement over its predecessors, such as RSA and DSA, as it provided better security with smaller key sizes. The smaller key size makes the cryptographic process less time-consuming and more memory efficient [22]. An elliptic curve is a curve with the general form of the equation as Eq. (1):

$$y^2 = x^3 + ax + b \tag{1}$$

With the condition Eq. (2):

$$4a^3 + 27b^2 \neq 0 \tag{2}$$

This cryptographic process utilizes a mathematical property of elliptic curves on a finite field $GF(p)$,

which is difficult to invert. The general form of an elliptic curve over GF(p) is defined as follows Eq. (3):

$$y^2 \equiv x^3 + ax + b(mod\ p) \quad (3)$$

In this case, p is a prime number, and the elements in $GF(p)$ are $\{0, 1, 2, \dots, p-1\}$. This property is known as the elliptic curve discrete logarithm problem, creating a one-way function that is both difficult to solve and efficient to implement. Therefore, ECC is rapidly replacing previous public key algorithms in communication networks and systems, including in the context of digital signatures.

2.4 Multi-signature

Multi-signature or multi-sig is a protocol in digital signature cryptography that allows multiple signers to produce one master signature that represents their validation of a data or message [23] Verification is done by combining all signers' public keys that have been given publicly. This approach increases the security and efficiency of using digital signatures [24]. To crack a multi-sig, an attacker must know the private keys of all authorized parties who signed the data, a very difficult task. The efficiency of multi-sig is seen in the verification process. In traditional digital signatures, the verifier has to check all existing signatures with their respective owners. However, in the multi-sig method, the verifier only needs to check one signature using the master public key.

2.5 Comparison to existing method

Based on the fundamentals above, several papers have implemented the digital signature method using various attributes. Wellem et al. [25], utilized the standard elliptic curve digital signature algorithm (ECDSA). single signature method and QR code to address the vulnerability of paper-based academic documents by proposing a document authentication system. The QR code embedded in printed documents contains a digital signature, allowing for subsequent authentication through either document upload or QR code scanning. However, this research has limited features that lead to security and efficiency issues. The ECDSA usage is already known to be vulnerable to side-channel analysis attacks. The research does not specify which key generation technique they use in the method. The inferior key generator method will lead to vulnerability. Furthermore, the lack of support for multi-signature features will impact the method's overall security and efficiency. Bisheh-Niasar et al. [26] proposed an optimized implementation of EdDSA. The researcher showcased significant improvements in execution time and performance while maintaining security levels comparable to advanced encryption standard (AES)-128. The usage of EdDSA covers the security issue in ECDSA. The method employed a hash function for the key generation which is the standard method. However, it lacks multi-signature support which has the same limitations as the previous methods. The method also does not specify the implementation of QR codes within it system. G. Shankar et al. [14] propose an improved multi-signature scheme for ensuring the authenticity of digital documents. The method employed EdDSA for the cryptography algorithm and includes support for multi-signature features. For the key generation, the method used the user's biometric credentials, derived from a hash of the XOR operation on an Aadhaar number (ID number), fingerprint, and a random number to generate the private key defined as follows Eq. (4)

$$k_{priv} = Hash(F_{print} \oplus ID \oplus RNG) \quad (4)$$

However, this method of key generation is not usual which leads to efficiency and security issues. XOR-ing a biometric number with a random number may not be an optimal approach for key generation, as directly utilizing the random number would suffice. Moreover, incorporating non-random information into key generation might introduce vulnerabilities, potentially aiding attackers in key cracking. Additionally, the paper does not specify support for QR code implementation, which could be a valuable aspect of document verification and user-friendly interactions.

Our proposed method introduces several improvements over the existing methods. Firstly, we proposed to utilize the robust EdDSA digital signature algorithm, known for its superiority over ECDSA [27], addressing security concerns and improving efficiency. Secondly, our method eliminates the insignificant pre-process in key generation by employing CSPRNG for the key generation process that is dedicated to the cryptographic field, mitigating security issues present in previous methods and optimizing efficiency. Thirdly, we introduce multi-signature support, adding a layer to security and also the efficiency. Lastly, we propose the integration of QR code support to simplify the user experience during signature verification. Table 1 provides a brief overview of previous research and its comparison with our method.

Table 1. Attributes comparison of the previous method

Method	Cryptography Algorithm	Key Generator	Multi-Signature Support	QR Code support
Wellem et al. [25]	ECDSA	Not specified	–	✓
Bisheh-Niasar et al. [26]	ECDSA	Hash function	–	–
G. Shankar et al. [14]	EdDSA	The hash of XOR on the Aadhaar number, fingerprint, and a random number	✓	–
Proposed Method	**EdDSA**	**CSPRNG**	✓	✓

Algorithm 2: EdDSA algorithm	
1	**Key Generation:** **Input:** None - Create random number for seed (k) using hash - Split the hash of k into two part, a and b half left of the seed as a, half right as b - Private key $(d) = a$ - Set base point (G) - Public key $(Q) = dG$ **Output:** private key (d), public key (Q)
2	**Signature Generation:** **Input:** Data (M), private key (d), public key (Q), half right of seed (b) - Calculate the hash of b and M $r = Hash(d, M)$ - Find point in R $R = rG$ - Calculate the hash of Data (M), public key (Q), and point R $h = Hash(M, Q, R)$ - Find s as component of the digital signature $s = (r + h * d) \bmod n$ **Output:** Digital signature (R, s)
3	**Signature Verification:** **Input:** Public key (Q), Data (M), and Digital signature (R, s) - Get the hash value from the received data (M), public key (Q), and point R $h = Hash(M, Q, R)$ - Find point $(p1) = sG$ - Find point $(p2)$ using elliptic curve calculation $p2 = R + hQ$ - Compare point $(p1)$ and point $(p2)$, if equal the signature is valid. Otherwise, the signature is invalid **Output:** Valid or Invalid

3. Proposed method

This section describes the main components used in the efficient multi-signature method as an improvement over existing methods and its implementation. This method utilizes EdDSA, which is considered the most efficient and secure digital signature algorithm. To provide the key, this study utilizes the CSPRNG which provides an efficient and secure unpredictable random number. This study also integrates QR codes as a signature format to improve the efficiency of the verification process.

3.1 EdDSA

EdDSA is one of the latest digital signature algorithms that still belongs to the ECC family. EdDSA was developed by Bernstein in 2012 to overcome the weaknesses of its predecessor, ECDSA [28]. EdDSA utilized Edwards curve that is defined as follows Eq. (5).

$$ax^2 + y^2 = 1 + dx^2y^2 \tag{5}$$

Where d is a non-square constant. Edwards curves excel in terms of their superior efficiency in addition, doubling, and tripling operations when compared to the Weierstrass form of elliptic curves that are used in ECDSA. EdDSA offers a high increase in efficiency without compromising the security level [29]. ECDSA, its predecessor, was notoriously vulnerable to side-channel analysis attacks because it relied heavily on the quality of the random number generator (RNG) algorithm for the cryptographic process. One of the advantages of EdDSA is that it overcomes this weakness by implementing a hash function in the algorithm to replace the RNG [26]. Based on the advantages offered, this research proposes using EdDSA as the cryptographic algorithm for the proposed digital signature method. Algorithm 2 below explains how the EdDSA cryptographic calculation process works.

3.2 CSPRNG

CSPRNG is a function specifically designed to generate unpredictable random numbers, used for security purposes in cryptographic processes. The numbers generated by CSPRNG have a high degree of unpredictability and are indistinguishable from true random numbers. An attack involving the known output of the CSPRNG generator has more than a 50% chance of being predicted, making it mathematically impossible to get an accurate prediction of the numbers to come [30]. Furthermore, even if an attacker knows the full CSPRNG algorithm used, they cannot use backtracking to analyze previously generated numbers due to seed uncertainty. This makes it resistant to external and internal attacks. This study proposes the use of CSPRNG for EdDSA key generation, taking into account the required security qualities.

3.3 QR code

QR Code is a 2D matrix image used as a data representation in the form of a pattern. QR Code was first developed by an automotive company in Japan in 1994 to facilitate tracking of the production process [31]. Data is read by scanning the QR Code pattern, which is very efficient as it allows the data input process by simply scanning the image.

3.4 Implementation

This study has implemented the proposed method on a device with the specifications described in Table 2. An illustration of the proposed method can be seen in Fig. 1. In this study, the overall process is classified into three main phases: key generation, signature generation, and signature verification. This method uses three signers as an example of a multi-signature methodKey generation is the first step in executing this method. This stage generates a key pair (private key and public key) for each signer CSPRNG is used to perform number generation. EdDSA uses 32-byte-long keys, so this study customized CSPRNG to generate values of that size. The random number generated is used as the private key. The private key is provided to the EdDSA public key generator to

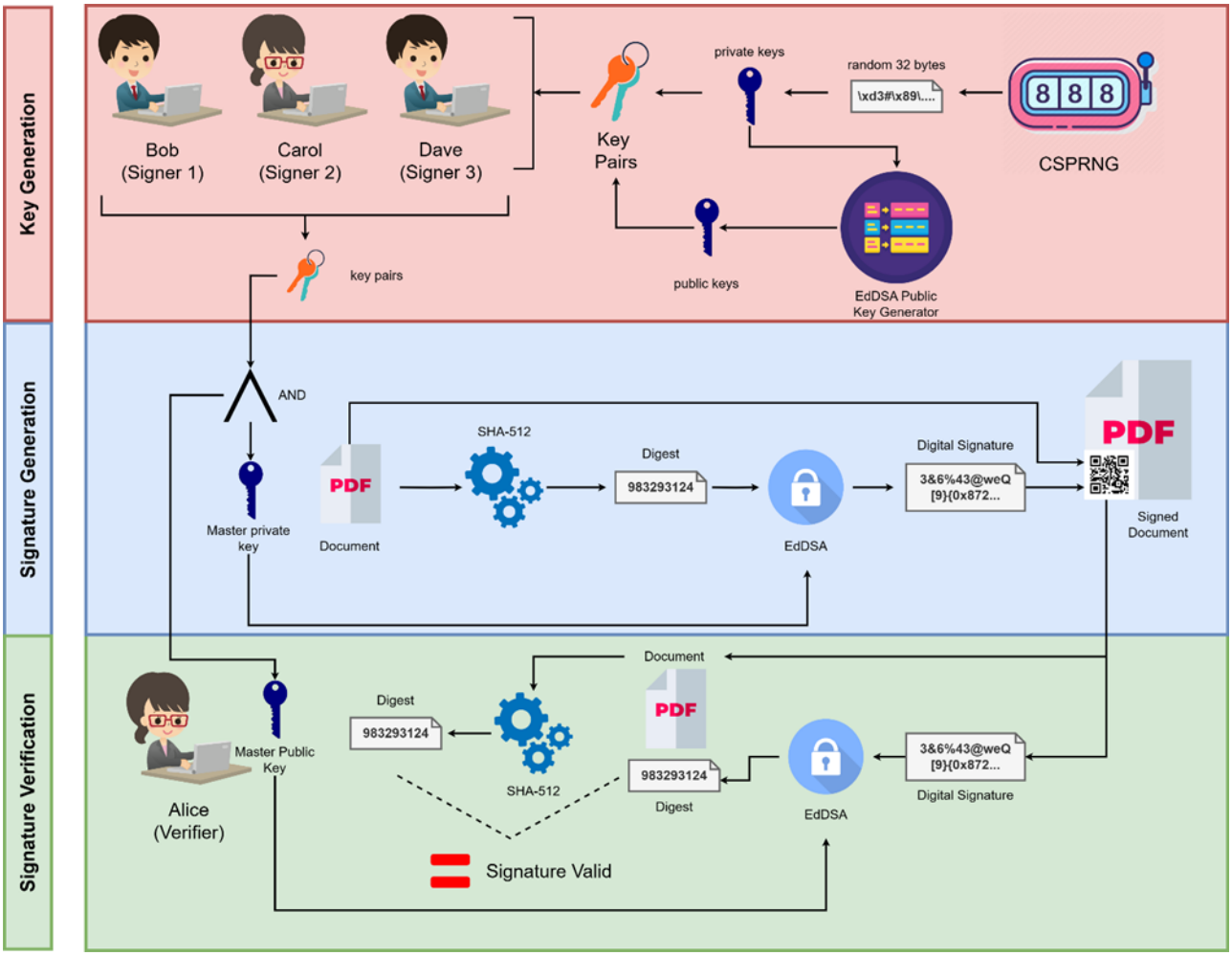


Figure. 1 Proposed method

obtain the public key. The key pair is then stored by each signer. These key pairs correspond to each other and operations performed with one key can only be efficiently reversed by the other key. The example of 32 bytes private key generated from CSPRNG and its public key pair from EdDSA in base64 format are given as follows Eq. (6).

$$k_{priv} = "vjSGfNHY9nQ/FnxtgaXvo/wy9 + KrawS00CbIldp/hz4 = "$$
$$k_{pub} = "5A1DO4RotYkX8fca0k55jUF fwHwedF4uRLgdbD54P7w = " \quad (6)$$

In the signature generation phase, the objective is to generate a digital signature of the document. The private keys of all signers are combined using the AND operator to get a master private key that represents all signers. This process is defined as follows Eq. (7).

$$master\ k_{priv} = k1_{priv} \wedge k2_{priv} \wedge k3_{priv} \quad (7)$$

The digest is calculated by hashing the document. This digest value is passed to the EdDSA algorithm along with the master private key to generate a digital signature. To ease the process of future verification, the digital signature is encoded into a QR code format and embedded into the signed document.

Finally, at the signature verification stage, the verifier will check the validity of the signature. The resulting digest of the hashed document will be compared with the digest of the digital signature. The digest of the digital signature can be obtained by decrypting the digital signature using the master public key from the combined public keys of all signers. Master public key is obtained with Eq. (8).

$$master\ k_{pub} = k1_{pub} \wedge k2_{pub} \wedge k3_{pub} \quad (8)$$

Verification is considered successful if both digests have the same value. Otherwise, the signature is considered invalid, indicating that the signature or document may have been forged.

Table 2. Device specifications

Specifications	Details
Model	Acer Aspire E5-476G
Processor	Intel® Core™ i5-8250U
RAM	8 GB DDR4 2400 MHz
Graphics	NVIDIA® GeForce® MX150
Storage	128 GB M.2 SSD
OS	Windows 11 Home 64-Bit
Programming language	Python 3.9.7

4. Evaluation and results

In this section, this study evaluates the performance and security of the proposed method. This evaluation aims to measure the efficiency and security of the proposed method in comparison with existing methods.

4.1 Device

The performance test was run on a hardware device with the specification Intel® Core™ i5-8250U Processor, 8 GB RAM, NVIDIA® GeForce® MX150 graphics card, and Windows 11 Home 64-Bit operating system (OS). The detailed specifications can be found in Table 2 below.

4.2 Performance assessment

This subsection aims to evaluate how well the proposed method performs on a real-time application by running the method on multiple execution counts for 3 signers and a verifier. The findings provide an understanding of the performance of the method on a larger scale in a practical implementation. The parameters observed in this study include key generation, signature generation, and signature generation time.

We conducted a performance assessment on our device with specifications outlined in Table 2. Various methods were implemented to evaluate their efficiency based on distinct cryptographic attributes, such as the encryption algorithm used, key generation techniques, and the support for multi-signatures. The detailed differences between the various method can be seen in Table 1.

For comparison, this study compared the proposed method with the methods proposed by Wellem et al. [25], which uses the standard ECDSA single signature method, Bisheh-Niasar et al. [26], with the EdDSA single signature method, and G. Shankar et al. [14], with the EdDSA double signature method but using Aadhar numbers, fingerprints, and random numbers for the key generation process.

In the assessment of key generation performance, this study analyzes how efficient each method is in generating key pairs, which is one of the fundamental aspects in the cryptographic process. As can be seen in Table 3 and Fig. 2 (a), the proposed method, which uses CSPRNG for key generation, has the fastest execution time compared to the previous methods. Overall, for all execution counts, the

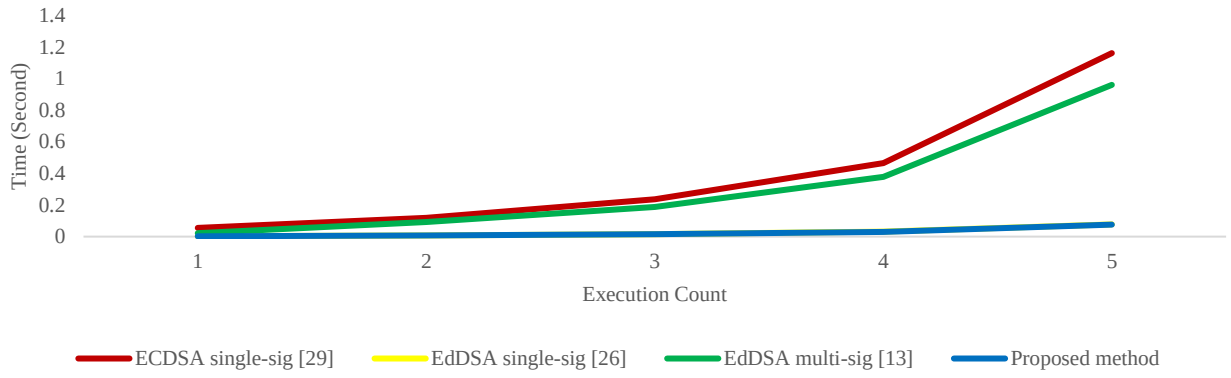
Table 3. Performance comparison

Method	Parameter	Execution times				
		10	**50**	**100**	**200**	**500**
Wellem et al. [25]	Key Generation	0.055086	0.118986	0.236023	0.464312	1.16126
	Signature Generation	0.0246723	0.127484	0.249966	0.506532	1.21894
	Signature Verification	0.0920871	0.474262	0.967783	2.06469	4.66371
	Overall	0.1718454	0.720732	1.453772	3.035534	7.04391
Bisheh-Niasar et al. [26]	Key Generation	0.0023415	0.0076932	0.0154026	0.0306648	0.077323
	Signature Generation	0.0016377	0.0076458	0.0155037	0.0310347	0.076934
	Signature Verification	0.0038067	0.0182367	0.0365517	0.0768912	0.183661
	Overall	0.0077859	0.0335757	0.067458	0.1385907	0.337918
G. Shankar et al. [14]	Key Generation	0.0219081	0.0923676	0.188202	0.377695	0.960287
	Signature Generation	0.001122	0.0052923	0.0106201	0.0211665	0.053014
	Signature Verification	0.0012437	0.006081	0.0121651	0.0245325	0.060815
	Overall	0.0242738	0.1037409	0.2109872	0.423394	1.074116
Proposed method	Key Generation	**0.0019248**	**0.0075903**	**0.0151734**	**0.0304335**	**0.075952**
	Signature Generation	**0.001122**	**0.0053251**	**0.0105689**	**0.0211677**	**0.052826**
	Signature Verification	**0.0012335**	**0.0060961**	**0.0121345**	**0.0245531**	**0.061249**
	Overall	**0.0042803**	**0.0190115**	**0.0378768**	**0.0761543**	**0.190026**

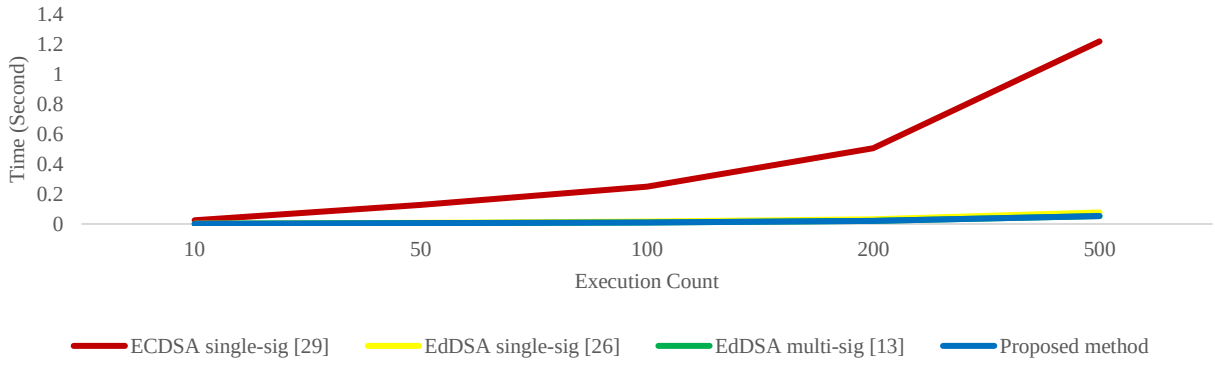
proposed method has an average of 91.79% better than the latest method [14] for key generation time. On paper, EdDSA is already more efficient than ECDSA. The method [14] has a longer execution time because there is preprocessing that is not conventional for key generation, even though they use the EdDSA algorithm. The proposed method uses CSPRNG which is proven to be very efficient and secure for cryptographic processes.

For signature generation performance assessment, this study analyzes how efficiently each method generates digital signatures. The proposed method, which uses multi-signature EdDSA, has the fastest execution time compared to the previous methods, as shown in Table 3 and Fig. 2 (b). Again, EdDSA outperforms ECDSA in both single-signature and multi-signature methods. All multi-signature methods outperform single-signature methods as they only need to generate one master signature representing all three signers, while single-signature methods have to generate three signatures for each signer. For all execution counts, the proposed methods have an average of 95.71% and 92.25% better than [25] and [26], respectively. Both the proposed method and [14] have similar performance as they both use the same multi-signature EdDSA method.

The final stage of the digital signature method process is testing the performance of signature verification. This test analyzes how efficient each method is in verifying the pre-generated signatures. Table 3 and Fig. 2 (c) show similar results to the signature generation assessment, where the multiple signature method surpasses the single signature method in terms of efficiency. The multi-signature

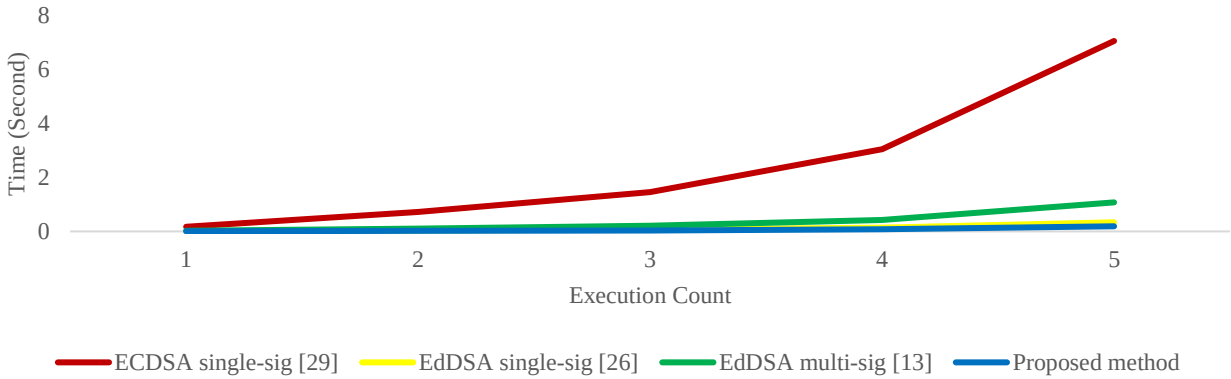


(a)



(b)

(c)



(d)

Figure. 2 Performance comparison: (a) key generation, (b) signature generation, (c) signature verification, and (d) overall performance

method only needs to verify one primary signature that represents the signatures of all signers, while the single-signature method needs to verify all three signatures. The proposed method overcomes all previous methods with an average improvement of 98.72% and 67.16% compared to [25] and [26], respectively. [14] achieved similar results to this study as it used the same multi-signature EdDSA method used in this study.

In all three basic processes of digital signature, the proposed method that implements CSPRNG key generation in EdDSA multi-signature signature method, surpasses all existing methods in terms of performance. This clearly shows that the use of CSPRNG is the best choice for key generation as this algorithm is already efficient on paper and has been established for cryptographic use, in this case used in digital signature methods. Multi-signature methods outperform single signature methods in signature generation and signature verification because they only need one master signature to represent all signers. Therefore, they only need to create and verify signatures once. It should be noted that this study previously conducted a performance assessment against the DSA method and a commonly used algorithm, namely RSA. However, the results were astronomically inefficient compared to the proposed method. Therefore, this study decided not to include them in this proposed analysis, given that in Figure 2 it is difficult to distinguish between the two methods. For the overall performance, the proposed method achieved an average of 97.42% and 44.26% against [25] and [26], respectively. Meanwhile, against the latest method [14], the proposed method shows an improvement of 76.27%. This indicates that the proposed method offers better efficiency compared to the existing methods, especially in terms of lower execution time on all steps in the digital signature method.

5. Conclusion

This study proposes an efficient double signature method with QR code integration for document authentication using an EdDSA-based algorithm. This proposed method uses CSPRNG for unpredictable, robust, and efficient key generation. EdDSA was chosen as the digital signature algorithm as it is one of the latest algorithms that offers a high-security level and superior efficiency compared to previous algorithms. The application of the multi-signature method in this study improves the overall efficiency, as it integrates all the signer's private and public keys. This allows signature generation and signature verification to be done only once. Finally, this study recommends the implementation of QR codes in its method to facilitate the signature verification process in real-world situations. With the QR code, the signature can be easily read by the verifier through scanning the QR code. This study evaluates the proposed method by performing an efficiency performance assessment through various executions for 3 signers and a verifier. The observed parameters include key generation, signature generation, and signature generation time. The proposed method can outperform all existing methods. Against the latest method [14], the proposed method shows an average improvement of 76.27% for all parameters in the performance assessment. This study makes a significant contribution to the field of document file authentication by presenting an efficient and secure method.

Nomenclature

- K_{pr}: private key
- K_{pb}: public key
- m: the message that will be signed
- $H(.)$: hash function to generate digest
- digest: unique fixed-size representation of data
- $C(.)$: encryption function
- S_m: digital signature of the message
- m_{signed} : signed message, combination of message and digital signature
- $D(.)$: decryption function
- D_{S_m}: decrypted digital signature
- k: random number used as a seed
- $seed$: initial value
- a: the left half of the seed
- b: the right half of the seed
- d: private key
- G: base point on the elliptic curve
- Q: public key,
- M: the message that will be signed
- $Hash(.)$: a hash function
- r: the result of hashing the concatenation of private key and message
- R: a point on the elliptic curve
- h: the result of hashing the concatenation of message, public key, and point R
- s: digital signature component
- n: the order of the elliptic curve
- $p1$: point 1 on elliptic curve
- $p2$: point 2 on elliptic curve
- $GF(.)$: Galois Field

Conflicts of interest

The authors declare no conflict of interest

Author contributions

Conceptualization, M.Y. and W.D.W.; methodology, M.Y. and W.D.W.; software, W.D.W; validation, M.Y.; formal analysis, M.Y. and W.D.W; supervision, M.Y.; funding acquisition, M.Y.

References

[1] A. A. Santosa, Y. T. Prasetyo, F. Alamsjah, A. A. N. P. Redi, and I. Gunawan, "How the COVID-19 Pandemic Affected the Sustainable Adoption of Digital Signature: An Integrated Factors Analysis Model", *Sustainability*, Vol. 14, No. 7, p. 4281, 2022.

[2] S. J. Basha, V. S. Veesam, T. Ammannamma, S. Navudu, and M. V. V. S. Subrahmanyam, "Security enhancement of digital signatures for blockchain using EdDSA algorithm", In: *Proc. of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks*, pp. 274–278, 2021.

[3] G. Maulani, E. A. Nabila, and W. Y. Sari, "Digital Certificate Authority with Blockchain Cybersecurity in Education", *International Journal of Cyber and IT Service Management (IJCITSM)*, Vol. 1, pp. 136-10, 2021.

[4] S. Pramanik and S. K. Bandyopadhyay, "Signature Image Hiding in Color Image using Steganography and Cryptography based on Digital Signature Concepts", In: *Proc. of 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pp. 665-669, 2020.

[5] A. M. Qadir and N. Varol, "A Review Paper on Cryptography", In: *Proc. of 7th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1–6, 2019.

[6] S. A. Busafi and B. Kumar, "Review and Analysis of Cryptography Techniques", In: *Proc. of 9th International Conference System Modeling and Advancement in Research Trends (SMART)*, pp. 323–327, 2020.

[7] M. T. Gençoğlu and M. T. Gençoğlu, "Importance of Cryptography in Information Security", *IOSR Journal of Computer Engineering (IOSR-JCE)*, Vol. 21, No. 1, pp. 65–68, 2019.

[8] M. A. A. Shabi, "A Survey on Symmetric and Asymmetric Cryptography Algorithms in information Security", *International Journal of Scientific and Research Publications (IJSRP)*, Vol. 9, No. 3, pp. 576-589, 2019.

[9] Z. A. Saputri, A. Sudarsono, and M. Yuliana, "E-voting security system for the election of EEPIS BEM president", In: *Proc. of 2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, pp. 147–152, 2017.

[10] R. Kasodhan and N. Gupta, "A New Approach of Digital Signature Verification based on BioGamal Algorithm", In: *Proc. of 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 10–15, 2019.

[11] W. Lin, "Trends in Data Protection and Encryption Technologies", *Springer Nature Switzerland*, 2023.

[12] R. Imam, Q. M. Areeb, A. Alturki, and F. Anwer, "Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status", *IEEE Access*, Vol. 9, pp. 155949–155976, 2021.

[13] Q. Feng, K. Yang, M. Ma, and D. He, "Efficient Multi-Party EdDSA Signature With Identifiable Aborts and its Applications to Blockchain", *IEEE Transactions on Information Forensics and Security*, Vol. 18, pp. 1937–1950, 2023.

[14] G. Shankar, L. H. Farhani, P. A. C. Angelin, P. Singh, A. Alqahtani, A. Singh, G. Kaur, and I. A. Samori, "Improved Multisignature Scheme for Authenticity of Digital Document in Digital Forensics Using Edward-Curve Digital Signature Algorithm", *Security and Communication Networks*, Vol. 2023, pp. 1-18, 2023.

[15] J. P. Arockiasamy, L. E. Benjamin, and R. U. Vaidyanathan, "Beyond Statistical Analysis in Chaos-Based CSPRNG Design", *Security and Communication Networks*, Vol. 2021, pp. 1-14, 2021.

[16] P. P. Pittalia, "A Comparative Study of Hash Algorithms in Cryptography", *International Journal of Computer Science and Mobile Computing (IJCSMC)*, Vol. 8, pp. 147-152, 2019.

[17] W. D. Walidaniy, M. Yuliana, and H. Briantoro, "Improvement of PSNR by Using Shannon-Fano Compression Technique in AES-LSB StegoCrypto", In: *Proc. of 2022 International Electronics Symposium (IES)*, 2022.

[18] H. Bensalem, Y. Blaquière, and Y. Savaria, "Acceleration of the Secure Hash Algorithm-256 (SHA-256) on an FPGA-CPU Cluster Using OpenCL", In: *Proc. of 2021 IEEE International*

Symposium on Circuits and Systems (ISCAS), pp. 1–5, 2021.

[19] M. R. Anwar, D. Apriani, and I. R. Adianita, "Hash Algorithm In Verification Of Certificate Data Integrity And Security", *Aptisi Transactions on Technopreneurship (ATT)*, Vol. 3, No. 2, pp. 65–72, 2021.

[20] N. J. G. Saho, E. C. Ezin, B. Watson, E. Badouel, and O. Niang, "Comparative Study on the Performance of Elliptic Curve Cryptography Algorithms with Cryptography through RSA Algorithm", *Colloque Africain sur la Recherche en Informatique et en Mathématiques Apliquées*, 2020.

[21] A. H. Darrel and Menezes, "Elliptic Curve Cryptography", *Encyclopedia of Cryptography, Security and Privacy*, pp. 1–2, 2019.

[22] L. Nayak and V. Jayalakshmi, "A Study of Securing Healthcare Big Data using DNA Encoding based ECC", In: *Proc. of 6th International Conference on Inventive Computation Technologies (ICICT)*, pp. 348–352, 2021.

[23] G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille, "Simple Schnorr multi-signatures with applications to Bitcoin", *Des Codes Cryptogr*, Vol. 87, No. 9, pp. 2139–2164, 2019.

[24] Y. Xiao, P. Zhang, and Y. Liu, "Secure and Efficient Multi-Signature Schemes for Fabric: An Enterprise Blockchain Platform", *IEEE Transactions on Information Forensics and Security*, Vol. 16, pp. 1782–1794, 2021.

[25] T. Wellem, Y. Nataliani, and A. Iriani, "Academic Document Authentication using Elliptic Curve Digital Signature Algorithm and QR Code", *JOIV : International Journal on Informatics Visualization*, Vol. 6, No. 3, p. 667, 2022.

[26] M. B. Niasar, R. Azarderakhsh, and M. M. Kermani, "Cryptographic Accelerators for Digital Signature Based on Ed25519", *IEEE Trans. Very Large Scale Integr VLSI Syst*, Vol. 29, No. 7, pp. 1297–1305, 2021.

[27] G. J and S. Koppu, "An empirical study to demonstrate that EdDSA can be used as a performance improvement alternative to ECDSA in Blockchain and IoT", *Informatica*, Vol. 46, No. 2, 2022.

[28] A. H. Darrel and Menezes, "Elliptic Curve Signature Schemes", *Encyclopedia of Cryptography, Security and Privacy*, pp. 1–3, 2019.

[29] W. D. Walidaniy, M. Yuliana, and H. A. Darwito, "Enhancing Document Authenticity with QR Codes and ECC-Based Digital Signatures", In: *Proc. of 2023 International Electronics Symposium (IES)*, pp. 238–243, 2023.

[30] P. Kietzmann, T. C. Schmidt, and M. Wählisch, "A Guideline on Pseudorandom Number Generation (PRNG) in the IoT", *ACM Comput Surv*, Vol. 54, No. 6, pp. 1–38, 2022.

[31] E. Uçak, "Teaching Materials Developed Using QR Code Technology in Science Classes", *International Journal of Progressive Education*, Vol. 15, No. 4, pp. 215–228, 2019.