Article

Not peer-reviewed version

# LDSEGoV: An Efficient Lightweight Digital Signature Scheme for E-Governance Authentication

Seema Sirpal , Pardeep Singh , Om Pal *

Posted Date: 25 February 2026

doi: 10.20944/preprints202602.1346.v1

Keywords: lightweight cryptography; e-governance; authentication; digital signatures; discrete logarithm problem (DLP); random oracle model (ROM)

*Article*

# LDSEGoV: An Efficient Lightweight Digital Signature Scheme for E-Governance Authentication

**Seema Sirpal [1,2], Pardeep Singh[1,3] and Om Pal [1,3,*]**

[1] Department of Computer Science, University of Delhi, 110007, India
[2] Delhi University Computer Centre, University of Delhi, 110007, India
[3] SafeQbit Technologies Private Limited, Noida 201318, India
* Correspondence: opal@cs.du.ac.in

**Abstract**

Digital signatures serves as a crucial cryptographic primitive in an e-governance system for the authentication of citizen-government interactions. Traditional methods (DSA, ECDSA) pose computational overheads at resource-limited endpoints and centralized verification servers. While complex-number cryptography provides theoretical efficiency through the Complex Discrete Logarithm Problem (CDLP), prior works often fail to meet the requirements for real-world applications. This paper advances the knowledge in lightweight cryptography by introducing LDSEGoV, a lightweight digital signature scheme for e-governance infrastructure. The proposed method overcomes the shortcomings of previous methods by incorporating sound modular arithmetic for consistent verification, using NIST-approved hash functions. Furthermore, we provide a comprehensive security analysis that provides the formal security proofs of existential unforgeability (EUF-CMA) of the proposed scheme in the Random Oracle Model. Additionally, the experimental results show a 6.5× improvement in signing performance and a 24.76× improvement in verification performance over ECDSA, with a 61% reduction in signature size. These results demonstrate that LDSEGoV is suitable for large-scale digital governance systems for authentication scenarios.

**Keywords:** lightweight cryptography; e-governance; authentication; digital signatures; discrete logarithm problem (DLP); random oracle model (ROM)

## 1. Introduction

Digital governance systems are transforming the interface between citizens and the government through electronic service delivery, supporting scalable authentication of transactions such as tax returns, social security benefits, certificate generation, and regulatory compliance. As governments around the world establish Common Service Centers (CSCs), mobile portals, and verification infrastructure to support millions of citizens, it is necessary to ensure the authenticity, integrity, and non-repudiation of digitally signed documents to maintain public trust in the government. Digital signatures are essential for security, as they bind the transactions to trusted signers and allow verifications on a scale while preserving efficiency. However, the computational overhead of conventional signature schemes also becomes a bottleneck for both resource-constrained nodes (e.g., CSCs, smart cards or mobile devices), and verification servers enrolling thousands of concurrent authentication requests.

Well-recognized cryptographic primitives such as RSA, DSA and ECDSA offer high security assurance but they introduce computational overheads that prohibit scaling within large e-systems [1]. The signing process using the ECDSA algorithm on modern hardware takes around 0.699 ms, but this jumps to 150-200 ms when performed on secure smartcards. Verification, which takes around 3.368 ms/signature, limits the throughput on single-server systems to about 297 signatures per second, which is not sufficient for handling maximum government service traffic. In addition, the DSA and ECDSA signatures consume 832 bits, including encoding overheads, resulting in high bandwidth requirements in rural e-governance networks that support constrained 2G/3G communication channels.

 doi:10.20944/preprints202602.1346.v1



Therefore, efficient cryptographic algorithms are required to facilitate e-governance infrastructure, allowing for a trade-off between high security and low computational and communication overheads.

Complex number cryptography, which relies on the Complex Discrete Logarithm Problem (CDLP), has extended the use case into the Gaussian integers $\mathbb{Z}_p[i]$ with the potential for theoretical efficiency improvements due to the expanded problem space, potential reduction in signature size, and improved arithmetic. The Shortened Complex Digital Signature Algorithm (SCDSA), proposed by Mughal et al. [2], is the first attempt to leverage CDLP for lightweight digital signatures. Our analysis shows that SCDSA exhibit several shortcomings that make it impractical for use: it does not work for production-quality parameters ($p \geq 2048$ bits) because of inconsistent complex modular arithmetic in the exponentiation step; it is based on the deprecated SHA-1 hash function, which has been shown to be vulnerable to collision attacks and has been discouraged by NIST in favor of more secure alternatives [3,4] and lacks any formal security analysis in the context of EUF-CMA security games.

This work proposes LDSEGoV (Lightweight Digital Signature for E-Governance), a formally secure CDLP-based signature scheme that address the flaws in SCDSA while maintaining efficiency suitable for the e-governance infrastructure. The approach is designed to ensure consistency in verification by strictly performing modular reduction over $\mathbb{Z}_p[i]$, thus ensuring proper functionality for all sizes of parameters, including those in production scenarios where $p \geq 2048$ bits. LDSEGoV relies on hash functions proven secure by NIST, and SHA-256 is specified as the default hash function because of its well-established 128-bit security level. Moreover, SHAKE128 and SHAKE256 are also examined to meet the requirements of applications that demand variable-length hashing. The choice of SHA-256 is in line with current cryptographic traditions and is designed to be efficient within the standard framework of e-governance signature size of 320 bits.

Experimental validation demonstrate the verification rate of about 7,350 signatures per second in single-threaded mode (1,000 ms / 0.136 ms per signature). This corresponds to a 24.76 times speedup over ECDSA, which performs 297 signatures per second on the same hardware, thus proving the ability to handle multiple authentication requests from hundreds of CSCs without horizontal scaling. The smaller size of the signature also plays a part in solving the problem of bandwith bottleneck that is common in rural 2G/3G networks.

In addition, the paper provides a comprehensive security analysis both in formal and informal way. The formal security proof under the Random Oracle Model (ROM) for LDSEGoV shows existentially unforgeability against adaptive chosen-message attacks (EUF-CMA), which ensures its resistance to forgery and replay updates. The informal security analysis illustrates that LDSEGoV can withstand various attacks, thus demonstrating its suitability for e-governance authentication.

### *1.1. Key Contributions*

The primary contributions of this study are as follows:

1. **Lightweight Digital Signature Scheme for E-Governance:** In this work, we propose LDSEGoV, a CDLP signature scheme designed for large-scale digital governance applications, addressing computational barriers in resource-constrained endpoints and verification servers.
2. **Corrected and Validated CDLP Implementation:** Our assessment identifies critical flaws in Mughal et al.'s work [2] (SCDSA), such as verification failure for large parameters and the use of outdated SHA-1. We provide a corrected algorithm that ensures universal verification success for all parameter sizes, as validated by 1,000 iterations with 2048-bit primes.
3. **Formal and Informal Security Analysis:** We prove the existential unforgeability under adaptive chosen-message attacks (EUA-CMA) in the Random Oracle Model via a game-based reduction to the hardness of CDLP via the forking lemma, and show resistance to forgery, replay, man-in-the-middle, and nonce-replay attacks augmented by an informal security analysis.
4. **Comprehensive Performance Analysis:** Comparison with DSA, ECDSA, and Mughal et al. shows a speedup of 6.5 times in signing, a speedup of 24.76 times in verification, and a 61% reduction in signature size.

 doi:10.20944/preprints202602.1346.v1



5. **E-Governance Deployment Impact Analysis:** We analyze the scalability of LDSEGoV in the CSC infrastructure and show that the performance of verification allows for single-server deployment to handle 10-100 times traffic scaling, unlike ECDSA requiring distributed clusters.

### *1.2. Paper Organization*

The remainder of the paper is organized as follows. Section 2 introduces the cryptographic preliminaries and mathematical concepts related to complex modular arithmetic. In Section 3, we present the review of the existing literature highlighting the research gap. Section 4 describes the system and the formulation of the problem. The proposed LDSEGoV algorithm is described in Section 5. In Section 6, we detail security in the ROM and other security considerations. Performance comparison with other methods and implementation results are given in Section 7. Section 8 is Conclusion and future work.

## 2. Preliminaries

This section introduces the key concepts necessary for understanding the cryptographic basis of the proposed lightweight digital signature method.

### *2.1. Discrete Logarithm Problem (DLP) and Integer Factorization*

The Discrete Logarithm Problem (DLP) is a major field of study in cryptography, playing a pivotal role in the security designs underpinning many public-key cryptographic systems. Within the context of the DLP, an individual is provided with a generator $\alpha$ alongside an element $\beta$ [5]. The objective is to find an integer $x$ that satisfies the equation $alpha^x \equiv \beta \pmod{p}$, with $p$ being a prime number. The security of most cryptographic schemes, especially DSA and ECDSA, depends upon the problems resulting from DLP. The security of digital signature schemes is primarily based on the difficulties involved in breaking the DLP. This complexity significantly restricts an adversary's ability to derive the secret key from the public key. Another fundamental cryptographic problem is integer factorization. Given a large composite number $N$, the aim is to factor it into its prime components. The security of RSA and many other cryptosystems is founded on this issue. Though DLP and Integer Factorization are separate issues, they both depend on the challenge of solving number theory problems to offer security, hence resembling one another. The related CDLP in our proposed scheme LDSEGoV offers further security by extending DLP into the complex number realm.

### *2.2. Modular Arithmetic and Complex Numbers*

Modern cryptography depends on the fundamental function known as modular arithmetic; its use on complex numbers is crucial to our proposed LDSEGoV. LDSEGoV uses modular arithmetic in the complex domain, which adds more complexity and security even if conventional cryptography methods operate inside integer fields. LDSEGoV operates the modulus operation on both the real and imaginary components of complex numbers. The modulus of a complex number $z = a + bi$, where $a$ and $b$ are real numbers, is found by using the modulo operation on both $a$ and $b$ with respect to a large prime $n$. Adversaries have to negotiate two-dimensional number spaces, so this extension of complex numbers makes cryptanalysis more difficult in relation to traditional integer-based cryptosystems. Advancing the scope to complex numbers in cryptographic systems increases both security and computational complexity, hence strengthening the system against cryptographic attacks.

### *2.3. Complex Discrete Logarithm Problem (CDLP)*

The central cryptographic challenge in LDSEGoV is CDLP. It extends the conventional DLP over the complex numbers. Given a generator $\alpha$ and a complex number $\beta$, the goal of the CDLP is to find an integer $x$ such that $\alpha^x = \beta \mod p$, where $p$ is a large prime modulus, and $\alpha$ and $\beta$ are complex numbers [6]. Solving the CDLP is considered to be computationally difficult, so it provides a strong foundation for LDSEGoV protection. The security of the system is strengthened by the greater

 doi:10.20944/preprints202602.1346.v1



challenge of solving this issue in the complex number domain. Although conventional DLP-based systems are susceptible to sophisticated algorithms, the CDLP offers more resilience to such attacks.

### *2.4. Digital Signature Algorithm (DSA)*

Among the most often used cryptographic techniques for creating and validating digital signatures is DSA. Based on DLP, DSA runs using the undermentioned procedures:

1. Key Generation: Select random $x \in \{1, 2, \ldots, \omega - 1\}$, compute $Y = \alpha^x \bmod p$, return $(x, Y)$.
2. Signature Generation: Select random $\varepsilon \in \{1, 2, \ldots, \omega - 1\}$, $c = (\alpha^\varepsilon \bmod p) \bmod \omega$, $z = \varepsilon^{-1}(\mathcal{H}(M) + x \cdot c) \bmod \omega$, return $(c, z)$.
3. Signature Verification: Compute $w = z^{-1} \bmod \omega$, $u_1 = \mathcal{H}(M) \cdot w \bmod \omega$, $u_2 = c \cdot w \bmod \omega$, $v = (\alpha^{u_1} \cdot Y^{u_2} \bmod p) \bmod \omega$, verify if $v = c$ then accept, else reject.

Although DSA provides a strong security guarantee based on the DLP, it is still vulnerable to certain attacks, especially if the random value $\varepsilon$ is chosen insecurely. The LDSEGoV improves the verification step and uses more robust cryptographic primitives to overcome these weaknesses.

### *2.5. Elliptic Curve Digital Signature Algorithm (ECDSA)*

ECDSA provides improved efficiency in computation with the same level of security as DSA but with smaller key sizes. Although ECDSA uses elliptic curve arithmetic for key and signature pair generation, it follows the same basic process as DSA.

1. Key Generation: Select random $x \in \{1, 2, \ldots, \omega - 1\}$, compute $Y = x \cdot P$, return $(x, Y)$.
2. Signature Generation: Select random $\varepsilon \in \{1, 2, \ldots, \omega - 1\}$, $c = (\varepsilon \cdot P)_x \bmod \omega$, $z = \varepsilon^{-1}(\mathcal{H}(M) + x \cdot c) \bmod \omega$, return $(c, z)$.
3. Signature Verification: Compute $w = z^{-1} \bmod \omega$, $u_1 = \mathcal{H}(M) \cdot w \bmod \omega$, $u_2 = c \cdot w \bmod \omega$, $V = u_1 \cdot P + u_2 \cdot Y$, $v = V_x \bmod \omega$, verify if $v = c$ then accept, else reject.

ECDSA provides a more efficient alternative to DSA, especially in environments where smaller key sizes are advantageous. However, like DSA, it is vulnerable to attacks if the random value $\varepsilon$ is reused or poorly chosen. LDSEGoV improves on these schemes by introducing complex number-based cryptography to increase overall security.

CDLP Correctness: For a valid signature $(c, z)$:

$$\begin{aligned} V &= u_1 \cdot P + u_2 \cdot Y \\ &= \mathcal{H}(M) \cdot z^{-1} \cdot P + c \cdot z^{-1} \cdot (x \cdot P) \\ &= z^{-1}(\mathcal{H}(M) + x \cdot c) \cdot P \\ &= \varepsilon \cdot P \quad (\text{since } z = \varepsilon^{-1}(\mathcal{H}(M) + x \cdot c)) \end{aligned}$$

Therefore, $v = V_x = (\varepsilon \cdot P)_x = c$.

## 3. Related Work and Research Gap Analysis

For the as far status of e-governance infrastructure have evolved three main lines for digital signature schemes: traditional ones following on the integer factorization problem (RSA) and discrete logarithms (DSA, ECDSA), light ones when resources are constrained, and hybrid using both cryptographic primitives with document verification primitives. This section discusses representative works in these topic areas, some of the common security and performance limitations, and clearly establishes where our LDSEGoV solution stands in relation to contemporary research.

### *3.1. Classical Digital Signature Schemes for E-Governance*

Sharif et al. [7] proposed a web-based PDF authentication system that uses RSA signatures along with SHA-3 hashing. The analysis shows that there is negligible processing overhead with varying sizes of PDFs while maintaining integrity and non-repudiation. However, the proposed technique

 doi:10.20944/preprints202602.1346.v1



does not have content-aware verification capabilities beyond basic file hashing. In another research, Aufa et al. [8] analyzed the combination of RSA-1024 and DSA-512 for secure government communications and found that lower computation times are appropriate for e-governance applications. This research focused on the cryptographic efficiency aspect but did not consider the scalability issues that arise in high-speed verification scenarios; particularly, single-server ECDSA signing supports only 297 signatures per second, which is inadequate for a national-level authentication system handling thousands of simultaneous Common Service Centers.

### *3.2. Lightweight Digital Signature Scheme*

Taking into consideration the computational limitations that are necessarily present in classical schemes, there has been a need for lightweight solutions that are more suited for resource-limited settings. Meshram et al. [9] designed a discrete logarithm scheme that makes use of subtree topologies to minimize communication costs. Although they showed the scheme to be more efficient, it is not proven secure in the EUF-CMA model and is also vulnerable to side-channel attacks and fault injection, which are not desirable properties for a government authentication infrastructure that also needs to be legally defensible and physically secure. Ahmed and Barukab [10] offer a method that combines lightweight encryption with digital signatures for IoT, which yields smaller signature sizes. However, their method fails to resist nonce-reuse attacks. Yavuz et al. [11] focus on medical IoT authentication using scalable frameworks that emphasize device-level efficiency; however, their focus on healthcare telemetry applications fails to apply to e-governance document authentication requirements, which require legally binding citizen transactions.

### *3.3. Complex Number-Based Cryptography and CDLP*

The Complex number cryptography relies on the Complex Discrete Logarithm Problem (CDLP) with the objective of achieving efficiency through calculations in the Gaussian integer domain, which is represented as $\mathbb{Z}_p[i]$. Lalem et al. [12] revisited complex number based signatures for their efficiency in lightweight applications. Nevertheless, like other previous studies, they did not address the issues of ambiguities in verification, especially when using large sizes of parameters, nor did they provide a proof of security.

Cryptanalysis of Mughal et al. (SCDSA)

Mughal et al. [2] proposed the Shortened Complex Digital Signature Algorithm (SCDSA) for human-centric resource-constrained IoT networks, utilizing complex exponentiation to achieve smaller signatures. While the idea is very promising, our analysis of the implementation reveals some very important shortcomings, as follows:

1. *Verification Failures with Production Parameters:* The verification algorithm of SCDSA shows deterministic failures for the prime modulus $p$, which meets the condition $p \geq 2048$ bits (required for 128-bit security). The main reason is the lack of modular reduction in the complex exponentiation $(a + bi)^k \bmod p$. The original scheme reduces the result only after complex exponentiation, which causes integer overflow and incorrect signature rejection. The fixed code requires modular reduction after each multiplication in the square-and-multiply step, ensuring consistent verification results for any parameter size.
2. *Deprecated cryptographic primitives:* SCDSA employs SHA-1 hashes, which were deprecated by NIST in 2011 due to the collision attacks [13], [3]. Modern e-governance systems require NIST-compliant alternatives such as SHA-256, SHA-3, or SHAKE256, which provide approximately $2^{128}$ collision resistance.
3. *Lack of Formal Security Validation:* SCDSA lacks a proof of existential unforgeability under adaptive chosen-message attacks (EUF-CMA), which is the usual security notion for digital signatures. Without formal game-based proofs for security reduction to CDLP hardness, security assertions are left unvalidated, which causes concerns for legally binding government authentication that demands cryptographic rigor.

 doi:10.20944/preprints202602.1346.v1



4. *Ambiguous Modular Arithmetic Semantics:* The division operation in the scheme, expressed as $s = k/(r + U_a) \bmod q$, conflates the concepts of division and modular inversion, leading to ambiguities in implementation. The proper expression requires the evaluation of a modular multiplicative inverse: $s = k \cdot (r + U_a)^{-1} \bmod q$ where $(r + U_a)^{-1}$ can be calculated through the Extended Euclidean Algorithm.
5. *Vulnerabilities in Floating-Point Arithmetic:* The existing implementation of SCDSA uses floating-point arithmetic for complex exponentiation, causing non-determinism due to rounding errors. Cryptographic algorithms should be based on exact integer arithmetic to ensure reproducibility across different platforms, which is a requirement for government document verification, where the correctness of a signature can be disputed in court years after its creation.

### *3.4. Cryptanalytic Advances Against Classical Schemes*

Recent cryptanalytic work has revealed that there are certain weaknesses in widely used signature schemes. The Minerva attack [14] showed the electromagnetic side-channel recovery of ECDSA private keys through partial nonce leakage, thus compromising confidence in the standard discrete logarithm-based schemes. Macchetti [15] proposed related-nonce attacks that allow full ECDSA key recovery with poor randomness, requiring only a few observations of signatures.

Fault injection attacks remain a threat to hardened implementations. Cao et al. [16] showed the ability of lattice-based fault attacks to obtain ECDSA private keys by exploiting weak curves via single fault injections. Hou et al. [17] offer a thorough review of the vulnerabilities of RSA, DSA, and ECDSA to fault injection attacks in commercial implementations, emphasizing the inadequacy of theoretical security proofs in the face of resistance to physical attacks.

Implementation-level vulnerabilities continue to exist despite many years of research. Blessing et al. [18] showed empirical evidence of the Bleichenbacher-style RSA signature forgery attack being feasible on modern cryptographic libraries. Geimer [19] showed timing side-channel attacks on Infineon's hardware implementation of ECDSA. Taken together, these results make it clear that cryptographic protocols require not only formal security proofs but also constant-time and fault-tolerant implementations to secure the government authentication infrastructure.

## 4. Identification of Problem and System Modelling

### *4.1. Problem Identification*

The commonly used signature schemes, such as DSA, are based on the DLP. The three basic public parameters are the large prime $p$, the small prime $\omega$ such that $\omega$ divides $p - 1$, and the generator parameter $\alpha$, which is defined as $\alpha = h^{(p-1)/\omega} \bmod p$. This model significantly imposes high computational complexity, which is particularly evident in resource-limited e-governance devices such as CSC terminals, smart cards, and mobile devices in rural areas with limited power and processing capacity. While solving this, researchers opt for smaller instances of $p$ and $\omega$. Reducing the parameter sizes (in other words, smaller (smaller $p$ and $\omega$) may ease the computational burden to some extent; yet, this approach provides an unacceptably poor security and performance trade-off. For instance, reducing p from 2048 bits to 1024 bits cuts signing time by around 75%, simultaneously reducing the level of security from 112-bit to 80-bit, which is not strong enough for government authentication schemes requiring long-term validity of signatures (7-10 years). On the other hand, keeping appropriate security levels with regular parameters($p \geq 2048$ bits, $\omega \geq 256$ bits) raises a dual bottleneck: The signing speed of ECDSA on smartcards worsens to around 150-200 ms, while the verification process at 3.368 ms per signature limits the throughput of single-server systems to 297 signatures per second, which is not sufficient for maximum e-governance traffic rates of over thousands of simultaneous submissions. Additionally, the size of DSA/ECDSA signatures is 832 bits, which is equivalent to 416 bytes for a four-document citizen application when communicated through bandwidth-limited rural networks (64-384 kbps). These challenges, further exacerbated by the increased use of digital services in the context of digital governance, require the development of cryptographic algorithms

 doi:10.20944/preprints202602.1346.v1



tailored to the specific constraint profile of the e-governance domain, characterized by high transaction volumes, diverse device capabilities, and intermittent connectivity.

While the complex-number cryptography offers computational advantage, the previous implementations (SCDSA[2]) have been found to have several important shortcomings that make them unsuitable for use in legally binding e-governance applications, where the validity of signatures needs to resist cryptanalytic attacks over an archival period of seven to ten years: (1) irregular modular arithmetic operations that result in verification errors for parameters p with 2048 bits or larger; (2) use of the SHA-1 hash function, which is no longer recommended due to the risk of collision attacks, and (3) lack of rigorous security proofs for resistance to adaptive chosen-message attacks.

This analysis identifies a significant gap: current signature schemes compromise on security and also on the computational requirement in the context of e-governance and prior attempts in this direction have implementation and validation issues [2]. Thus, a scalable government authentication infrastructure requires an efficient and secure CDLP-based signature method.

### *4.2. System Model for LDSEGoV*

The proposed system LDSEGoV is targeted at secure and lightweight e-governance deployments suitable even for resource limited end points like Common Service Center (CSC) terminals, smart cards, and mobile applications. Each party (citizen, CSC operator and governmental department) has its own pair of keys: a private key $x$ is kept securely inside the device (a HSM in case of servers or a secure enclave for smart cards), whereas his public key $Y$ is published on government PKI directory. As shown in Figure 1, when a citizen applies for a service through a CSC, the terminal signs the application with the private key $x$, and the government verification server verifies the signature with the public key $Y$ to ensure that the application comes from an authorized CSC and has not been altered during transmission.

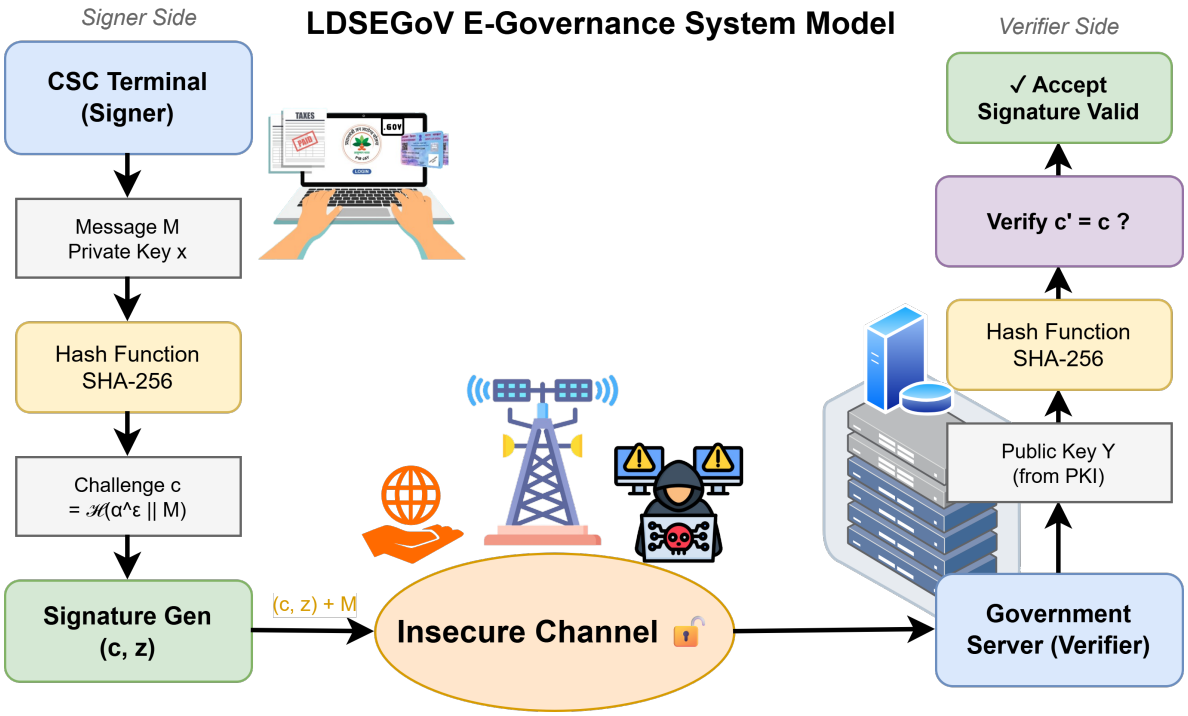


**Figure 1.** LDSEGoV System Model: Secure Signature Generation & Verification.

The process of generating the signature involves the signer (CSC terminal) calculating the pair $(c, z)$ using the ephemeral nonce $\varepsilon$ and the private key $x$, followed by the transmission of the signature through an insecure channel. The verifier (government server) then checks $c' = c$ using the public key $Y$, with a new nonce $\varepsilon$ for every signature to prevent replay attacks.

 doi:10.20944/preprints202602.1346.v1



*4.3. Threat Model for LDSEGoV*

We analyze an adversary $\mathcal{A}$ with network control (capable of intercepting and modifying messages), polynomial-time computational resources (in particular, unable to solve the Computational Diffie–Hellman Problem for modulus $p \geq 2048$ bits), and adaptive oracle access (capable of observing legitimate CSC signatures). The LDSEGoV defenses against the most important attack vectors are the following.

*Adversary Capabilities:*

1. *Network Control:* Complete access to communication channels enables the interception, modification, and insertion of messages exchanged between citizen service centers (CSCs) and government servers on the assumption of an insecure channel.
2. *Computational Power:* A polynomial-time bounded adversary is assumed capable of performing up to $2^{80}$ operations but is unable to solve discrete logarithm problem (CDLP) instances with modulus $p \geq 2048$ bits within feasible time.
3. *Oracle Access:* Adaptive chosen-message queries to signing oracles are considered, enabling observation of legitimate CSC transactions and reflecting real-world scenarios in which attackers observe citizen application submissions.
4. *Insider Threats:* The potential compromise of CSC operator credentials or temporary access to unattended terminals is acknowledged, while direct access to private keys stored within hardware security modules or secure enclaves is explicitly ruled out.

*Threat Scenarios and Defense:*

- *Man-in-the-Middle:* The modification of the signed message $M$ to $M'$ is prevented since $c = \mathcal{H}(\alpha^{\varepsilon} \parallel M)$ binds signature to message hash through the property of $2^{128}$ preimage resistance of SHA-256.
- *Replay Attacks:* Using a new nonce value $\varepsilon$ for each signature prevents the same messages from being signed with the same pair $(c, z)$. In e-governance systems, including timestamps in the signed data offers a further safeguard against repeated claims for welfare benefits.
  Using a fresh nonce $\varepsilon$ for every signature avoids signatures on the same message with same pair $(c, z)$. For e-governance applications adding timestamps to the signed data would help avoiding multiple identity claims for welfare benefits.
- *Forgery:* The generation of valid $(c^*, z^*)$ without knowledge of the private key x is equivalent to the CDLP problem in the cyclic group, which is shown to be EUF-CMA secure in Theorem 1.
- *Eavesdropping:* The private key $x$ is never transmitted. The response $z = \varepsilon \cdot (c + x)^{-1} \bmod \omega$ hides both $x$ and $\varepsilon$ using one-way modular arithmetic.
- *Nonce Reuse:* The ability to see two signatures with the same value of $\varepsilon$ allows key extraction using the forking lemma. This is addressed by requiring fresh randomness (Algorithm 2) and the use of hardware random number generators (e.g., TPM 2.0, secure enclaves).
- *CSC Compromise:* Hardware-protected storage solutions (e.g., JavaCard, cloud-HSM) protect against private key extraction even when terminal access is available.

## 5. Proposed Lightweight Digital Signature Scheme

This section presents our proposed scheme, LDSEGoV: Lightweight Digital Signature for E-Governance, a corrected and formally validated complex-number signature scheme. The scheme works within the proposed four algorithmic framework. The full algorithm specification consists of four algorithms: system parameter setup (Algorithm 1), key pair generation (Algorithm 2), signing (Algorithm 3), and verification (Algorithm 4). The notations and parameters used in LDSEGoV are enumerated in Table 1.

 doi:10.20944/preprints202602.1346.v1



**Table 1.** Notation and parameters used in LDSEGoV.

| Symbol | Description |
|---|---|
| $p$ | Large prime modulus ($\|p\| \geq 2048$ bits) |
| $\omega$ | Subgroup order, where $\omega = p - 1$ |
| $\alpha$ | Complex generator in $\mathbb{Z}_p[i]$ with order $\omega$ |
| $\varepsilon$ | Ephemeral random nonce, $1 < \varepsilon < \omega$ |
| $\sigma = (c, z)$ | Digital signature pair |
| $\mathcal{H}(\cdot)$ | Cryptographic hash function (SHA-256/SHAKE256) |
| $Y$ | Public verification key (complex number mod $p$) |
| $x$ | Private signing key, $1 < x < \omega$ |
| $M$ | Message to be authenticated (End User's data (i.e., a PDF document, file, application payload, email, or any digital information needing authenticity verification) |
| $c$ | Challenge: $c = \mathcal{H}(\alpha^{\varepsilon} \parallel M) \bmod \omega$ |
| $z$ | Response: $z = \varepsilon \cdot (c + x)^{-1} \bmod \omega$ |
| $c'$ | Recomputed challenge during verification |

**Algorithm 1** LDSEGoV System Setup

**Require:** Security parameter $\lambda$
**Ensure:** System parameters $(p, \omega, \alpha)$
Choose a large prime $p > 2^{\lambda}$
Set group order $\omega \leftarrow p - 1$
Select generator $\alpha = (\alpha_r, \alpha_i)$ where $\alpha_r, \alpha_i \in \mathbb{Z}_p^*$
**return** $(p, \omega, \alpha)$

**Algorithm 2** Key Generation

**Require:** System parameters $(p, \omega, \alpha)$
**Ensure:** Key pair $(x, Y)$
**repeat**
  $x \xleftarrow{\$} \{1, 2, \ldots, \omega - 1\}$
**until** $\gcd(x, \omega) = 1$
$Y \leftarrow \alpha^x \bmod p$ {Complex exponentiation}
**return** Private key $x$, Public key $Y$

**Algorithm 3** Signature Generation

**Require:** Message $M$, private key $x$, parameters $(p, \omega, \alpha)$
**Ensure:** Signature $\sigma = (c, z)$
*attempts* $\leftarrow 0$
**while** *attempts* $<$ *MAX_ATTEMPTS* **do**
  **repeat**
    $\varepsilon \xleftarrow{\$} \{1, 2, \ldots, \omega - 1\}$
  **until** $\gcd(\varepsilon, \omega) = 1$
  $(u, v) \leftarrow \alpha^{\varepsilon} \bmod p$ {Extract real, imaginary parts}
  $c \leftarrow \mathcal{H}(u \parallel v \parallel M) \bmod \omega$
  **if** $\gcd(c + x, \omega) \neq 1$ **then**
    *attempts* $\leftarrow$ *attempts* $+ 1$
    **continue**
  **end if**
  $z \leftarrow \varepsilon \cdot (c + x)^{-1} \bmod \omega$
  **return** $\sigma = (c, z)$
**end while**
**abort** with error "Signature generation failed"

 doi:10.20944/preprints202602.1346.v1



**Algorithm 4** Signature Verification

**Require:** Message $M$, signature $\sigma = (c, z)$, public key $Y$, parameters $(p, \omega, \alpha)$
**Ensure:** Decision {accept, reject}
**if** $c \notin [1, \omega - 1]$ **or** $z \notin [1, \omega - 1]$ **then**
  **return** reject
**end if**
$\alpha^c \leftarrow (\alpha_r + i\alpha_i)^c \bmod p$ {Complex exponentiation}
$A \leftarrow Y \cdot \alpha^c \bmod p$ {Complex multiplication}
$(u', v') \leftarrow A^z \bmod p$ {Extract real, imaginary parts}
$c' \leftarrow \mathcal{H}(u' \parallel v' \parallel M) \bmod \omega$
**if** $c' = c$ **then**
  **return** accept
**else**
  **return** reject
**end if**

*5.1. Pseudocode for LDSEGoV:*

1. **System Setup:** Choose a large prime $p$, $\omega \leftarrow p - 1$, $\alpha \leftarrow$ generator (complex number), return $(p, \omega, \alpha)$.
2. **Key Generation:** Pick a random $x \in \{1, 2, \ldots, \omega - 1\}$, calculate $Y = \alpha^x \bmod p$, return $(x, Y)$.
3. **Signature Generation:** Select a random number $\varepsilon \in \{1, 2, \ldots, \omega - 1\}$, $c = \mathcal{H}(\alpha^\varepsilon \parallel M) \bmod \omega$, $z = \varepsilon \cdot (c + x)^{-1} \bmod \omega$, return $(c, z)$.
4. **Signature Verification:** Calculate $\alpha^c$, find $A = Y \cdot \alpha^c \bmod p$, $c' = \mathcal{H}(A^z \parallel M) \bmod \omega$, if $c' = c$ then accept, otherwise reject.

*5.2. Overview of the proposed method: LDSEGoV*

The LDSEGoV scheme operates in the Gaussian integer ring $\mathbb{Z}_p[i]$, leveraging complex modular arithmetic to achieve computational efficiency and maintain cryptographic security. Unlike conventional signature schemes that operate over prime fields or elliptic curve groups, the LDSEGoV scheme extends the DLP into the two-dimensional complex plane, where both the real and imaginary parts are used to define the difficulty of key recovery.

The message $M$ (end user's data i.e, file , PDF document, application payload, etc.) is signed by the signature generation mechanism based on the operation of complex exponentiation $\alpha^\varepsilon \bmod p$, where the base $\alpha = \alpha_r + i\alpha_i$ is a complex generator, and the exponent $\varepsilon$ is a per-message random value. The result of the exponentiation is a commitment point $(u, v) \in \mathbb{Z}_p \times \mathbb{Z}_p$, which is used to link the signature to the message $M$ and the ephemeral randomness. The challenge part $c$ is derived from the hash function by hashing the commitment point together with the message: $c = \mathcal{H}(u \parallel v \parallel M) \bmod \omega$. The response part $z$ is then calculated as $z = \varepsilon \cdot (c + x)^{-1} \bmod \omega$.

One of the basic security features of the LDSEGoV protocol is the mandatory use of new randomness $\varepsilon$ in each signature. This ensures that even when the same message is signed on different occasions, the signatures $(c, z)$ are different because of the unpredictability of $\alpha^\varepsilon$. If the same randomness $\varepsilon$ is reused, then an attacker can easily extract the private key $x$ by algebraic manipulation of two different signatures with the same randomness $\varepsilon$ but different challenges $c_1, c_2$.

Verification calculates the original commitment by computing $A = Y \cdot \alpha^c \bmod p$ and then raising it to the power of the response, resulting in $A^z \bmod p$. The validity of this process is justified by the following algebraic identity:

$$(Y \cdot \alpha^c)^z = (\alpha^x \cdot \alpha^c)^{\varepsilon \cdot (c+x)^{-1}} = \alpha^{(x+c) \cdot \varepsilon \cdot (c+x)^{-1}} = \alpha^\varepsilon$$

Therefore, a valid signature will result in the verifier reconstructing the same commitment $(u, v)$ that the signer computed originally, which further leads to $c' = c$ and acceptance of the signature.

 doi:10.20944/preprints202602.1346.v1



### 5.3. Implementation Considerations

The hash function $\mathcal{H}$ is specified as SHA-256 to guarantee compliance with NIST FIPS 180-4 and to provide collision resistance. As alternatives for variable-length outputs, SHAKE-128 and SHAKE-256 can be used without any modifications. All complex arithmetic operations are required to use uniform modular reduction after each operation to guarantee deterministic computation and to prevent verification failures. Key storage requires hardware security: USB tokens (PKCS#11) for CSC personnel, HSMs (FIPS 140-2 Level 3) for government agencies, and platform-specific secure enclaves (Android Keystore, iOS Secure Enclave) for mobile apps. Public keys $Y$ are embedded in X.509 certificates (RFC 5280) with OID mappings tailored to our scheme.

### 5.4. Multi-Option Parameter Selection

In addition, the security of LDSEGoV is enhanced by adding the Multi-option Parameter Selection (MPS) technique to the signature generation and verification process. To improve resistance against cryptanalytic attacks that rely on fixed algebraic structures, LDSEGoV proposes three different response formulas. Each of these formulas offers the same level of security in CDLP but employs different algebraic expressions involving the challenge c and the secret key x. As a result, it becomes much more challenging for an attacker to forge the signature as they need to account for all the alternatives. We have provided these three distinct formulations for both the signature and verification process along with their respective proofs of correctness and consistency in Table 2. The technique offers immunity against any possible cryptographic attacks, such as forgery and signature impersonation.

*Deployment Recommendation:* In the context of e-governance systems, MPS-1 is recommended as the default. MPS-2 and MPS-3 can be used in scenarios that demand higher flexibility of algorithms or better resistance to unknown structural attacks. The chosen variant needs to be included in the signature metadata to ensure that the correct verification formula is used by the verifier.

**Table 2.** Correctness Proofs for Multi-Option Signature and Verification.

| ID | Response $z$ | Verification | Correctness |
|---|---|---|---|
| MPS-1 | $\varepsilon/(c+x) \bmod \omega$ | $A = Y \cdot \alpha^{c}$ | $A^{z} = (\alpha^{x+c})^{\varepsilon/(c+x)} = \alpha^{\varepsilon}$ |
| MPS-2 | $\varepsilon/(x+c^2) \bmod \omega$ | $A = Y \cdot \alpha^{c^2}$ | $A^{z} = (\alpha^{x+c^2})^{\varepsilon/(x+c^2)} = \alpha^{\varepsilon}$ |
| MPS-3 | $\varepsilon/(xc+1) \bmod \omega$ | $A = Y^{c} \cdot \alpha$ | $A^{z} = (\alpha^{xc+1})^{\varepsilon/(xc+1)} = \alpha^{\varepsilon}$ |

### 5.5. Example Illustrating the working of LDSEGoV

I) Complex Modular Arithmetic

Operations are performed in the ring $\mathbb{Z}_p[i]$, where $i$ is the imaginary unit. For an instance, `complex_mod(w, p)` reduces the real and imaginary parts of $w$ modulo $p$ and exponentiation $\alpha^{\varepsilon}$ mod $p$ uses a square-and-multiply approach adapted for complex numbers. The use of complex numbers expands the problem space for the DLP and Computational Diffie-Hellman (CDH) problems. Solving DLP in $\mathbb{Z}_p[i]$ is conjectured to be harder than in integer fields due to its higher dimensionality (real and imaginary components) and non-commutative structures when combined with modular arithmetic.

II) System Setup for LDSEGoV

The LDSEGoV algorithm is engineered with best parameters to guarantee secure and efficiency. The prime number 5915587277 is used as modulus $p$, specifying the arithmetic between complex numbers to be in $\mathbb{Z}_p[i]$. The order of a subgroup is selected $\omega = 5915587276$, and it holds $\omega|(p-1)$, which leads to the existence of the cyclic subgroup in the Gaussian integer field. We chose $\alpha$ to be the complex number $11 + 12i$ which, when taken modulo $p$, becomes $4 + 5i$. All of the in-subgroup public key and ephemeral computations are based on this generator. For the secret key we took $x = 5$ which is a uniformly random integer in $[1, \omega - 1]$, and thus invertible modulo $\omega$. Similarly, the ephemeral key used during signature generation was set as $\varepsilon = 4$, also satisfying $\gcd(\varepsilon, \omega) = 1$ to prevent any

 doi:10.20944/preprints202602.1346.v1



leakage of secret key material. Key generation involves computing the public key $Y = \alpha^x \bmod p$. Signature generation involves hashing the ephemeral exponentiated value along with the message, and computing a signature $(c, z)$. Verification recalculates the expected hash and compares it with $c$.

III) Key Generation

Computing $\alpha^5 \bmod 7$ using square-and-multiply:

1. Binary of 5: `101`
2. $\alpha = 4 + 5i$
3. Step 1: $\alpha^1 = (4, 5)$
4. Step 2: $\alpha^2 = (4 + 5i)^2 = (4^2 - 5^2, 2 \cdot 4 \cdot 5) = (-9, 40) \Rightarrow (5, 5)$
5. Step 3: $\alpha^4 = (5 + 5i)^2 = (25 - 25, 50) = (0, 50) \Rightarrow (0, 1)$
6. Step 4: $\alpha^5 = g^4 \cdot g = (0 + i)(4 + 5i) = (-5 + 4i) \Rightarrow (2, 4)$

Alice's public key: $Y = (2, 4)$.

IV) Generation of Signature

1. Compute ephemeral value: $\alpha^\varepsilon = (4 + 5i)^4 \Rightarrow \alpha^4 = (0, 1)$.
2. Hash the tuple with the message "hello":
   $c = \mathcal{H}(\alpha^\varepsilon \parallel \texttt{hello}) = \mathcal{H}(0 \parallel 1 \parallel \texttt{hello})_{\text{SHA256}} = 11 \Rightarrow c = 11 \bmod 3 = 2$.
3. Check invertibility: $\gcd(c + x, \omega) = \gcd(2 + 5, 3) = \gcd(7, 3) = 1$.
4. Compute inverse and $z$:
   $(c + x)^{-1} \bmod \omega = 7^{-1} \bmod 3 = 1$.
   $z = \varepsilon \cdot (c + x)^{-1} \bmod \omega = 4 \cdot 1 \bmod 3 = 1$.
5. Final Signature: $\sigma = (c, z) = (2, 1)$.

V) Verification of Signature

1. Compute $\alpha^c$: $\alpha^2 = (4 + 5i)^2 = (5, 5)$.
2. Multiply with public key:

$$\begin{aligned} Y \cdot \alpha^c &= (2 + 4i)(5 + 5i) = (2 \cdot 5 - 4 \cdot 5,\ 2 \cdot 5 + 4 \cdot 5) \\ &= (10 - 20,\ 10 + 20) = (-10,\ 30) \\ &\equiv (4,\ 2) \bmod 7. \end{aligned}$$

3. Raise to $z$: $\alpha'^\varepsilon = (4 + 2i)^1 = (4, 2)$.
4. Recompute hash: $c' = \mathcal{H}(4 \parallel 2 \parallel \texttt{hello}) = 2$.
5. Compare: $c' = c \Rightarrow$ Signature is VALID.

## 6. Security Analysis of LDSEGoV Scheme

### *6.1. Formal Security Analysis in Random Oracle Model (ROM)*

We prove the security of LDSEGoV through a formal game proof in the Random Oracle Model (ROM). In this model, the hash function $\mathcal{H}$ is modeled as an ideal "black-box" that returns uniformly random values for every different input. This simplification helps in making formal security proofs and captures the main characteristics of cryptographic hash functions, such as SHA-256.

**Theorem 1.** *If the Complex Discrete Logarithm Problem (CDLP) is computationally intractable, then LDSEGoV is existentially unforgeable under adaptive chosen-message attacks (EUF-CMA) in the Random Oracle Model.*

**Proof of Theorem 1.** This theorem is proved by a series of four games, starting from the standard definition of EUF-CMA security and gradually changing the environment until a solution to a CDLP problem can be obtained. Each game differs slightly from the previous one, allowing a limit to be placed on the advantage of the adversary.

 doi:10.20944/preprints202602.1346.v1



**Game $G_0$: Standard EUF-CMA Challenge:**

This game captures the standard notion of signature unforgeability. The challenger $\mathcal{C}$ performs the following:

1. *Setup:* Generate a random private key $x \xleftarrow{\$} [1, \omega - 1]$ with $\gcd(x, \omega) = 1$, compute public key $Y = \alpha^x \bmod p$, and give $Y$ to the adversary $\mathcal{A}$.
2. *Oracle Access:* Provide $\mathcal{A}$ with two oracles:
   **Signing Oracle** $\mathcal{O}_{\text{Sign}}(M_i)$: On an input message $M_i$, the oracle generates a fresh random nonce $\varepsilon_i$, computes $c_i = \mathcal{H}(\alpha^{\varepsilon_i} \parallel M_i) \bmod \omega$, calculates $z_i = \varepsilon_i \cdot (c_i + x)^{-1} \bmod \omega$, and returns signature $(c_i, z_i)$.
   **Hash Oracle** $\mathcal{H}(\cdot)$: On input $(u \parallel v \parallel M)$, returns a hash value. The oracle works consistently: repeated queries on the same input produce the same output.
3. *Forgery:* After polynomial-time running the queries to both Oracles, $\mathcal{A}$ outputs a forgery attempt $(M^*, c^*, z^*)$ for a message $M^*$ it has never queried to$\mathcal{O}_{\text{Sign}}$.
4. *Winning Condition:* $\mathcal{A}$ wins if $(c^*, z^*)$ is a legitimate signature on $M^*$ accepted by the regular verification algorithm.

Let $\epsilon_0$ denote the probability that $\mathcal{A}$ wins in Game $G_0$.

**Game $G_1$: Explicit Random Oracle Simulation** This construction changes the internal workings of the hash oracle without changing anything for the adversary. The challenger keeps a record, represented by the set $\mathcal{L}_H$ (which is empty initially), of all hash queries:

On query $(u \parallel v \parallel M)$:

1. If $(u, v, M)$ exists in $\mathcal{L}_H$ with associated value $h$, return $h$.
2. Otherwise, sample $h \xleftarrow{\$} \mathbb{Z}_\omega$ uniformly, store $((u, v, M), h)$ in $\mathcal{L}_H$, and return $h$.

*Indistinguishability:* For $\mathcal{A}$, this change is purely syntactic. Both game scenarios respond identical to all queries. Therefore:

$$\Pr[\mathcal{A} \text{ wins in } G_1] = \Pr[\mathcal{A} \text{ wins in } G_0] = \epsilon_0$$

**Game $G_2$: Forking Lemma Application**

This game contains the forking method, which is the technical core of the security reduction. When $\mathcal{A}$ produces a valid forgery $(M^*, c^*, z^*)$, the challenger rewinds $\mathcal{A}$ to an appropriate point and replays it with a different hash oracle response.

*Forking Procedure:*

1. Suppose $\mathcal{A}$ makes $q_H$ hash queries $Q_1, Q_2, \ldots, Q_{q_H}$ before outputting forgery $(M^*, c^*, z^*)$.
2. The challenger identifies the critical query $Q_j = (u^* \parallel v^* \parallel M^*)$ where $u^*, v^*$ are derived from $\alpha^{\varepsilon^*}$ for some unknown $\varepsilon^*$ used in the forgery.
3. Rewind $\mathcal{A}$ to just before query $Q_j$, and respond with a fresh random value $c'^* \neq c^*$ instead of the original $c^*$.
4. If $\mathcal{A}$ outputs a second valid forgery $(M^*, c'^*, z'^*)$, we obtain two equations:

$$z^* = \varepsilon^* \cdot (c^* + x)^{-1} \bmod \omega \tag{1}$$
$$z'^* = \varepsilon^* \cdot (c'^* + x)^{-1} \bmod \omega \tag{2}$$

*Key Extraction:* Assuming $c^* \neq c'^*$ (which holds with probability $1 - 1/\omega \approx 1$), we can solve for the private key $x$:

From Equations (1) and (2), isolate $\varepsilon^*$:

$$\varepsilon^* = z^* \cdot (c^* + x) \bmod \omega \tag{3}$$
$$\varepsilon^* = z'^* \cdot (c'^* + x) \bmod \omega \tag{4}$$

 doi:10.20944/preprints202602.1346.v1



Setting these equal and solving for $x$:

$$z^* \cdot (c^* + x) \equiv z'^* \cdot (c'^* + x) \pmod{\omega} \tag{5}$$
$$z^* c^* + z^* x \equiv z'^* c'^* + z'^* x \pmod{\omega} \tag{6}$$
$$z^* c^* - z'^* c'^* \equiv x(z'^* - z^*) \pmod{\omega} \tag{7}$$
$$x \equiv \frac{z^* c^* - z'^* c'^*}{z'^* - z^*} \pmod{\omega} \tag{8}$$

provided $z'^* \neq z^*$, which follows from $c'^* \neq c^*$.

*Success Probability:* Using the General Forking Lemma [20], if $\mathcal{A}$ succeeds with probability $\epsilon_0$ after making $q_H$ hash queries, then the rewinding algorithm produces two valid forgeries with probability:

$$\epsilon_2 \geq \epsilon_0 \left( \frac{\epsilon_0}{q_H} - \frac{1}{\omega} \right)$$

For any non-negligible $\epsilon_0$ and polynomial $q_H$, this probability is non-negligible.

**Game $G_3$: CDLP Reduction**

We now construct a simulator $\mathcal{S}$ that solves CDLP using $\mathcal{A}$ as a subroutine. Given a CDLP challenge $(\alpha, Y)$ where $Y = \alpha^{\xi} \bmod p$ for unknown $\xi \in [1, \omega - 1]$, the simulator aims to compute $\xi$.

*Simulation Strategy:*

1. **Embed Challenge:** Set the public key $Y = Y$ and provide it to $\mathcal{A}$. The corresponding private key is $x = \xi$ (unknown to $\mathcal{S}$).
2. **Answer Hash Queries:** Maintain list $\mathcal{L}_H$ and respond to hash queries as in Game $G_1$ (uniformly random values).
3. **Answer Signing Queries:** For each signing query on message $M_i$:
   (a) Choose random $c_i, z_i \xleftarrow{\$} \mathbb{Z}_\omega$.
   (b) Compute $A_i = \alpha^{z_i^{-1}} \cdot Y^{-c_i \cdot z_i^{-1}} \bmod p$.
   (c) Extract $(u_i, v_i)$ from $A_i$ (real and imaginary parts).
   (d) Program the hash oracle: set $\mathcal{H}(u_i \parallel v_i \parallel M_i) = c_i$ by adding entry to $\mathcal{L}_H$.
   (e) Return signature $(c_i, z_i)$.

   This is a valid signature because:

$$(Y \cdot \alpha^{c_i})^{z_i} = (\alpha^{\xi} \cdot \alpha^{c_i})^{z_i} \bmod p \tag{9}$$
$$= \alpha^{(\xi + c_i) z_i} \bmod p \tag{10}$$
$$= A_i \quad \text{(by construction)} \tag{11}$$

4. **Extract CDLP Solution:** When $\mathcal{A}$ produces two forgeries $(M^*, c^*, z^*)$ and $(M^*, c'^*, z'^*)$ via the forking procedure in Game $G_2$, compute:

$$\xi = x = \frac{z^* c^* - z'^* c'^*}{z'^* - z^*} \bmod \omega$$

Verify: $Y \stackrel{?}{=} \alpha^{\xi} \bmod p$. Output $\xi$ as the CDLP solution.

*Indistinguishability:* The simulated signatures $(c_i, z_i)$ are distributed identically to real signatures because:

- In real scheme: $c_i = \mathcal{H}(\alpha^{\varepsilon_i} \parallel M_i)$ where $\varepsilon_i$ is uniform $\Rightarrow c_i$ is uniform (by ROM).
- In simulation: $c_i$ is directly sampled uniform.
- Both $z_i$ values satisfy the verification equation.

Thus, $\mathcal{A}$ cannot distinguish the simulation from the real game, and succeeds with probability $\epsilon_2$ (from Game $G_2$).

 doi:10.20944/preprints202602.1346.v1



**Final Reduction Analysis:** We have shown that if an adversary $\mathcal{A}$ breaks LDSEGoV with advantage $\epsilon_0$ after $q_H$ hash queries, then there exists a CDLP solver $\mathcal{S}$ with success probability:

$$\epsilon_{\text{CDLP}} \geq \epsilon_0 \left( \frac{\epsilon_0}{q_H} - \frac{1}{\omega} \right)$$

Since CDLP is assumed computationally hard (no polynomial-time algorithm solves it with non-negligible probability), $\epsilon_{\text{CDLP}}$ must be negligible. This implies $\epsilon_0$ is also negligible.

*Concrete Security:* For parameters $\omega \approx 2^{160}$ and adversary making $q_H \leq 2^{60}$ hash queries, breaking LDSEGoV with advantage $\epsilon_0 = 2^{-30}$ would yield a CDLP solver with advantage $\epsilon_{\text{CDLP}} \approx 2^{-90}$, contradicting the CDLP hardness assumption for 128-bit security.

Therefore, LDSEGoV is existentially unforgeable under adaptive chosen-message attacks in the Random Oracle Model, assuming CDLP hardness. □

**Interpretation for Practitioners:** Theorem 1 guarantees that even if the adversary views a polynomial number of valid signatures and can choose adaptively which messages to sign (as in the case of chosen-plaintext attacks), it is impossible to sign a new message without breaking the underlying CDLP. The assumption made in the random oracle model (ROM) is common practice when using cryptographic hash functions such as SHA-256, which currently have no known structural weaknesses relevant to this construction.

### *6.2. Informal Security Analysis*

In addition to the formal proof of security in the ROM, we analyze the resistance of LDSEGoV to real-world cryptographic attacks that might arise in the context of e-governance.

1. *Resistance to Key Recovery Attacks:* The private key $x$ is randomly chosen from $\{1, 2, \ldots, \omega - 1\}$ ensuring $\gcd(x, \omega) = 1$, which creates a large key space and makes brute-force attacks infeasible. The public key $Y = \alpha^x \mod p$ is derived using complex exponentiation, making the reverse computation equivalent to solving CDLP, which is considered computationally hard.
2. *Resistance to Forgery Attacks:* Each signature uses a fresh random nonce $\varepsilon$, making repeated signature correlation attacks ineffective. The hash function $\mathcal{H}(u \parallel v \parallel M)$ tightly couples the message to the signature, providing collision resistance and preventing message substitution.
3. *Resistance to Replay Attacks:* The uniqueness of the nonce $\varepsilon$ in every valid signature $(c, z)$ ensures that even if an attacker reuses a signature, the context mismatch can be detected by the verifier.
4. *Resistance to Chosen-Message Attacks:* The value $z = \varepsilon \cdot (c + x)^{-1} \mod \omega$ binds the signature to the private key. Without knowledge of $x$, it is computationally infeasible to create valid signatures for attacker-chosen messages.
5. *Collision Resistance:* The use of a secure cryptographic hash function ensures that it is infeasible to find two different message inputs producing the same $c = \mathcal{H}(u \parallel v \parallel M \bmod \omega$, thereby ensuring message integrity.
6. *Preimage Resistance:* The private key $x$ is never exposed and cannot be derived from the public signature elements, as the equation $z = \varepsilon \cdot (c + x)^{-1} \mod \omega$ hides both $\varepsilon$ and $x$, making reverse engineering of $x$ infeasible.
7. *Second Preimage Resistance:* The hash function $\mathcal{H}$ provides strong second preimage resistance, which prevents an adversary from finding a different message $M'$ that produces the same hash value as $M$ in the computation of $c$.
8. *Resistance to Side-Channel Attacks:* A fresh random $\varepsilon$ for each signature ensures that power or timing analysis attacks are ineffective due to the lack of repeatable computation patterns.

**Summary:** LDSEGoV is resistant to both theoretical attacks (as shown by formal verification) and practical attacks (achieved through careful nonce handling and constant-time arithmetic). The combination of CDLP security, cryptographic hash binding, and randomized signing provides a defense-in-depth strategy for the e-government authentication infrastructure.

 doi:10.20944/preprints202602.1346.v1



## 7. Implementation and Result Analysis

The performance and robustness of the proposed complex number-based digital signature algorithm, LDSEGoV, are evaluated in this section. The algorithm works on complex integers modulo a large prime number, taking advantage of the cryptographic robustness and algebraic complexity. In order to determine the efficiency of the LDSEGoV algorithm, a comparison test was conducted among SCDSA, DSA, and ECDSA algorithms. Experimental verification carried out on commodity hardware (Intel i7-10700, 16 GB RAM, Ubuntu 22.04) shows a verification throughput of $\approx 7,350$ signatures per second in single-threaded mode (1,000 ms / 0.136 ms per signature), which is a 24.76× improvement over ECDSA, which reaches 297 signatures per second on the same hardware, sufficient to handle concurrent authentication requests from hundreds of CSCs without horizontal scaling. The performance of each algorithm was tested for 1,000 iterations, and the results are presented in Table 3. The performance analysis was carried out based on three important functions: key generation, signature generation, and verification time.

**Table 3.** Performance Comparison of Digital Signature Schemes (Average over 1,000 Iterations).

| Algorithm | KeyGen (ms) | Sign (ms) | Verify (ms) |
|---|---|---|---|
| LDSEGoV (Proposed) | 0.0351 | 0.1080 | 0.1360 |
| SCDSA | 0.0359 | 0.0920 | 0.2980 |
| DSA | 602.8861 | 0.9266 | 0.5561 |
| ECDSA | 0.7814 | 0.6999 | 3.3681 |

The key results show that LDSEGoV has consistent success rate in verification for all hash functions, thus establishing the effectiveness of LDSEGoV. In contrast, SCDSA showed unsteady verification results, thereby being overcome by LDSEGoV. LDSEGoV outperforms DSA and ECDSA in key pair generation, signature generation, and verification with execution times of less than 0.13 ms. Although SCDSA showed faster signature generation with 0.092 ms, its high verification time of 0.298 ms and failure rates make it unsuitable for practical e-governance applications.

### *7.1. Performance Analysis Across Varying Message Sizes*

The performance of LDSEGoV, SCDSA, ECDSA, and DSA in generating signatures was evaluated for message sizes ranging from 1 MB to 10 MB. As shown in Figure 2(a), LDSEGoV had the lowest signature generation time (0.07–0.14 ms), performing better than ECDSA (0.88–0.99 ms) and DSA (0.62–0.77 ms) for all message sizes. SCDSA showed marginally faster signature generation (0.07–0.13 ms), but its unreliable verification negates this advantage. ECDSA had the highest fluctuation ($\pm$12%), so may be sensitive to message size, however LDSEGoV maintained stability ($\pm$8%). The verification latency was also tested and results were statistically evaluated. As shown in Figure 2(b), LDSEGoV achieved sub-millisecond verification (0.08–0.2 ms), 2.5 times faster than SCDSA (0.25–0.39 ms) and 15–20 times faster than ECDSA (3.09–4 ms). DSA (0.49–0.61 ms) performed competitively but was still 3 times slower than LDSEGoV. SCDSA's verification delays (0.25–0.39 ms) highlight its impracticality for real-world deployment. LDSEGoV and SCDSA performed similarly regarding communication overhead, as shown in Figure 2(c). Moreover, LDSEGoV's low and stable latency makes it suitable for high-throughput applications. Up to 92% reduction in communication cost translates to lower energy consumption. Moreover, LDSEGoV's compatibility with SHA3 and SHAKE128/256 hash functions ensures alignment with NIST standards.

 doi:10.20944/preprints202602.1346.v1



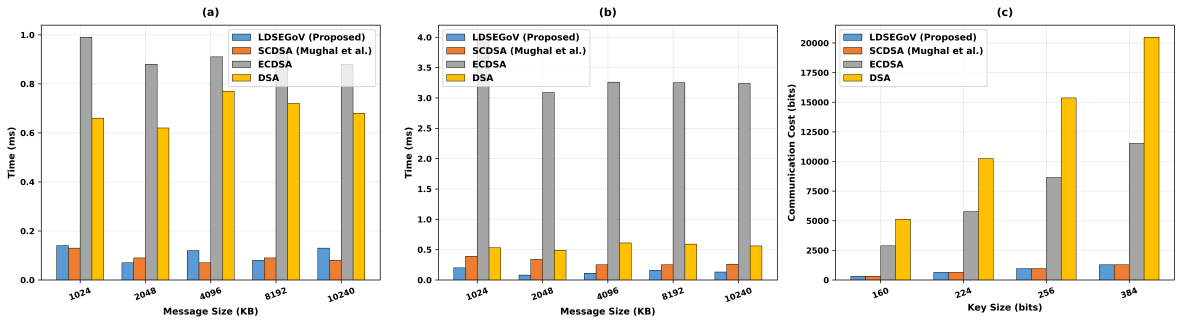


**Figure 2.** Comparative Performance of the proposed algorithm (a) Signature Generation Time, (b) Verification Time, (c) Communication Overhead.

### *7.2. Communication Overhead*

We have investigated the impact of communication overhead by varying the number of messages containing digital signatures. Table 4 summarizes the calculated communication overhead for various signature schemes. We have chosen $\alpha$ within the finite field spectrum at 512 bits, $\omega$ at 160 bits, and $p$ between 512 and 1024 bits for DSA. The lengths, as determined by the signature algorithm, are $|\omega| = |c| = 160$ bits and $|z| = |\mathcal{H}(M)| + |p| = 160$ bits + 512 bits [2]. Thus, the total length of each message's signature is $|p| + |\omega| + 160 = 512 + 160 + 160 = 832$ bits. Similarly, the ECDSA signature has a size of 832 bits. In our proposed LDSEGoV technique, the dimensions of $c$ and $z$ are each 160 bits. The signature size is $|\omega| + 160 = 160 + 160 = 320$ bits. The communication overhead was measured for varying key sizes (160–384 bits). LDSEGoV and SCDSA generated compact signatures (320–1280 bits), reducing bandwidth usage by 60–92% compared to ECDSA and DSA. At 256-bit security, LDSEGoV's signature size was 9 times smaller than ECDSA and 16 times smaller than DSA. This grants LDSEGoV an equivalent 60% decrease in communication overhead per message as SCDSA, in comparison to DSA and ECDSA.

**Table 4.** Communication Overhead for LDSEGoV Signature Generation and Verification. (In this Table, $T_h$, $T_e$, $T_g$, $T_{sm}$, and $T_{key}$ denote the overhead of the Hash Function, Exponentiation, Generator Function (g), ECC Scalar Multiplication and Key Generation, respectively).

| Scheme | Key Gen. | Signature Gen. | Verification | Total Overhead | Total Cost (bits) |
|---|---|---|---|---|---|
| DSA | $T_{key}$ | $T_h + T_e + T_g + T_{sm}$ | $T_h + T_e + T_g + T_{sm}$ | $T_{key} + T_h + T_e + T_g + T_{sm}$ | 832 |
| ECDSA | $T_{key}$ | $T_h + T_e + T_g + T_{sm}$ | $T_h + T_e + T_g + T_{sm}$ | $T_{key} + T_h + T_e + T_g + T_{sm}$ | 832 |
| Mughal et al. (SCDSA [2]) | $T_{key}$ | $T_h + T_e + T_g$ | $T_h + T_e + T_g$ | $T_{key} + T_h + T_e + T_g$ | 320 |
| LDSEGoV (Proposed) | $T_{key}$ | $T_h + T_e + T_g$ | $T_h + T_e + T_g$ | $T_{key} + T_h + T_e + T_g$ | 320 |

## 8. Conclusions

This work examined the computational barriers of conventional digital signature schemes (DSA, ECDSA) when implemented at a large scale in e-governance networks, where resource-constrained endpoints and centralized verification servers struggle to maintain an acceptable level of throughput. We propose LDSEGoV, a corrected and formally proven complex discrete logarithm-based digital signature scheme that address essential flaws identified in previous CDLP implementations through meticulous modular arithmetic, NIST-safe cryptographic primitives, and formal security proofs. Experimental results show that LDSEGoV provides a 6.5-fold improvement in signing performance and a 24.76-fold speedup in verification over ECDSA, with a 61% reduction in signature size. Server-side implementations maintain throughput at a level sufficient to handle hundreds of concurrent endpoints without horizontal scaling. The formal security proof verifies the existential unforgeability under adaptive chosen-message attacks (EUA-CMA) in the Random Oracle Model, with security reducible to the hardness of CDLP. The results justify the implementation of cost-efficient authentication in-

 doi:10.20944/preprints202602.1346.v1



frastructure for large-scale digital governance applications, especially in bandwidth-scarce networks. However, LDSEGoV is best suited for operational authentication scenarios where signature validity is expected to last from years to decades. Long-term archival applications with a lifetime of a century (e.g., land records, constitutional texts) or post-quantum security requirements should use well-established RSA/ECDSA or NIST-standardized post-quantum cryptographic schemes (ML-DSA, SLH-DSA). Applying CDLP — which is so far less thoroughly investigated in cryptanalytic literature than classical DLPs —would involve pessimistic parameter choices ($p \geq 2048$ bits) and maintenance of the security assessments over time to keep pace with the state of the art. Future research may explore the hardware design and implementation on resource-constrained platforms such as smart cards and IoT devices, hybrid cryptographic constructions of CDLP, along with post-quantum primitives for transitory security, including formal verification of implementation on a constant-time basis to prevent side-channel attacks.

**Author Contributions:** Conceptualization, S.S. and P.S.; methodology, P.S.; software, O.P.; validation, S.S, P.S. and O.P.; formal analysis, P.S. and S.S.; investigation, S.S. and O.P.; resources, O.P.; data curation, S.S. and O.P.; writing—original draft preparation, P.S. and S.S.; writing—review and editing, S.S., P.S. and O.P.; visualization, P.S.; supervision, O.P.; project administration, O.P.; All authors have read and agreed to the submitted version of the manuscript.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References


1. Fernández-Caramés, T.M. From Pre-Quantum to Post-Quantum IoT Security: A Survey on Quantum-Resistant Cryptosystems for the Internet of Things. *IEEE Internet of Things Journal* **2020**, *7*, 6457–6480. https://doi.org/10.1109/JIOT.2019.2958788.
2. Mughal, M.A.; Luo, X.; Ullah, A.; Ullah, S.; Mahmood, Z. A Lightweight Digital Signature Based Security Scheme for Human-Centered Internet of Things. *IEEE Access* **2018**, *6*, 31630–31643. https://doi.org/10.1109/ACCESS.2018.2844406.
3. Stevens, M.; Bursztein, E.; Karpman, P.; Albertini, A.; Markov, Y. The First Collision for Full SHA-1. In Proceedings of the Advances in Cryptology – CRYPTO 2017; Katz, J.; Shacham, H., Eds., Cham, 2017; pp. 570–596.
4. Leurent, G.; Peyrin, T. SHA-1 is a Shambles: First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20). USENIX Association, 2020, pp. 1839–1856.
5. Katz, J.; Lindell, Y. *Introduction to modern cryptography: Principles and protocols*; Chapman and hall/CRC, 2007. https://doi.org/10.1201/b17668.
6. Peng, S.; Zhao, L.; Al-Dubai, A.Y.; Zomaya, A.Y.; Hu, J.; Min, G.; Wang, Q. Secure Lightweight Stream Data Outsourcing for Internet of Things. *IEEE Internet of Things Journal* **2021**, *8*, 10815–10829. https://doi.org/10.1109/JIOT.2021.3050732.
7. Sharif, A.; Ginting, D.S.; Dias, A.D. Securing the Integrity of PDF Files using RSA Digital Signature and SHA-3 Hash Function. In Proceedings of the 2021 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA), 2021, pp. 154–159. https://doi.org/10.1109/DATABIA53375.2021.9650121.
8. Aufa, F.J.; Endroyono.; Affandi, A. Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm. In Proceedings of the 2018 4th International Conference on Science and Technology (ICST), 2018, pp. 1–5. https://doi.org/10.1109/ICSTC.2018.8528584.
9. Meshram, C.; Ibrahim, R.W.; Obaid, A.J.; Meshram, S.G.; Meshram, A.; El-Latif, A.M.A. Fractional chaotic maps based short signature scheme under human-centered IoT environments. *Journal of Advanced Research* **2021**, *32*, 139–148. Fractional Calculus Models for the Dynamics of Complex System, https://doi.org/https://doi.org/10.1016/j.jare.2020.08.015.
10. Ahmed, A.A.; Barukab, O.M. Unforgeable Digital Signature Integrated into Lightweight Encryption Based on Effective ECDH for Cybersecurity Mechanism in Internet of Things. *Processes* **2022**, *10*. https://doi.org/10.3390/pr10122631.

 doi:10.20944/preprints202602.1346.v1




11. Yavuz, A.A.; Darzi, S.; Nouma, S.E. Lightweight and Scalable Post-Quantum Authentication for Medical Internet of Things. *arXiv preprint arXiv:2311.18674* **2023**.
12. Lalem, F.; Laouid, A.; Kara, M.; Al-Khalidi, M.; Eleyan, A. A Novel Digital Signature Scheme for Advanced Asymmetric Encryption Techniques. *Applied Sciences* **2023**, *13*. https://doi.org/10.3390/app13085172.
13. Wang, X.; Yin, Y.L.; Yu, H. Finding collisions in the full SHA-1. In Proceedings of the Annual International Cryptology Conference. Springer, 2005, pp. 17–36.
14. Jancar, J.; Sedlacek, V.; Svenda, P.; Sys, M. Minerva: The curse of ECDSA nonces. Cryptology ePrint Archive, Paper 2020/728, 2020.
15. Macchetti, M. A Novel Related Nonce Attack for ECDSA. Cryptology ePrint Archive, Paper 2023/305, 2023.
16. Cao, W.; Shi, H.; Chen, H.; Wei, W.; Chen, J. Lattice-Based Weak Curve Fault Attack on ECDSA. In Proceedings of the IFIP Advances in Information and Communication Technology; Jøsang, A.; Futcher, L.; Hagen, J., Eds., Oslo, Norway, June 2021; Vol. AICT-625, *ICT Systems Security and Privacy Protection*, pp. 146–161. Part 3: Covert Channels and Cryptography, https://doi.org/10.1007/978-3-030-78120-0_10.
17. Hou, X.; Breier, J.; Jap, D.; Ma, L.; Bhasin, S.; Liu, Y. Physical security of deep learning on edge devices: Comprehensive evaluation of fault injection attack vectors. *Microelectronics Reliability* **2021**, *120*, 114116. https://doi.org/https://doi.org/10.1016/j.microrel.2021.114116.
18. Blessing, J.; Specter, M.A.; Weitzner, D.J. Cryptography in the Wild: An Empirical Analysis of Vulnerabilities in Cryptographic Libraries. In Proceedings of the Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, New York, NY, USA, 2024; ASIA CCS '24, p. 605–620. https://doi.org/10.1145/3634737.3657012.
19. Geimer, A.; Vergnolle, M.; Recoules, F.; Daniel, L.A.; Bardin, S.; Maurice, C. A Systematic Evaluation of Automated Tools for Side-Channel Vulnerabilities Detection in Cryptographic Libraries. In Proceedings of the Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 2023; CCS '23, p. 1690–1704. https://doi.org/10.1145/3576915.3623112.
20. Pointcheval, D.; Stern, J. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology* **2000**, *13*, 361–396. https://doi.org/10.1007/s001450010003.