Adv Syst Sci Appl 2026; **1**; 50-65
Published online at https://ijassa.ipu.ru.

A Novel Identity-Based Digital Signature Scheme Compliant with GOST 34.10-2018 and Its Application in Digital Government Systems

Minh Tu Le, Van Nghi Nguyen*, Duc Tho Hoang, Quoc Hoang Pham

Academy of Cryptography Techniques, Hanoi, Vietnam

Abstract: GOST 34.10-2018 serves as the official standard for public key cryptography in the Russian Federation, constituting a vital pillar of the broader GOST regulatory framework. This standard specifies a digital signature algorithm characterized by high security and performance, predicated on Elliptic Curve Cryptography (ECC). Due to its robustness and alignment with national security requirements, it is officially mandated for use in governmental information systems, financial transactions, and diverse cryptographic applications. However, as digital ecosystems evolve toward complex architectures—such as digital transformation and Digital Government—the need for streamlined and efficient certificate management becomes increasingly paramount. This paper addresses that challenge by proposing a novel, secure, and efficient Identity-Based Signature (IBS) scheme based on the GOST 34.10-2018 standard. Unlike traditional Public Key Infrastructure (PKI), our identity-based approach allows a user's public key to be derived from unique identifiers, significantly reducing the administrative burden of certificate revocation and storage. The security of the proposed scheme is rigorously evaluated within the Random Oracle Model (ROM), demonstrating existential unforgeability against sophisticated forgery attacks. Furthermore, we perform a comprehensive performance analysis, assessing the computational cost in comparison to existing publications and experimental efficacy. Results indicate that the signature process maintains high throughput and low latency. Consequently, the proposed scheme is both theoretically sound and highly viable for practical integration into modern digital infrastructures, ensuring alignment with established digital signature standards and operational efficiency.

Keywords: Identity-based signature, GOST 34.10-2018, elliptic curve, ROM, digital government

1. INTRODUCTION

The IBS scheme was first proposed by Adi Shamir in 1984 [1], in which the user's public key is simply the user's identification information, such as email address, phone number, personal identification number... The user's corresponding secret key will be initialized and issued by a trusted key generation center (Key Generation Center - KGC). This secret key is initialized based on a master secret held by KGC, and only KGC knows this master secret. This method helps reduce dependence on PKI and digital certificates, thereby simplifying the deployment process as well as saving certain operating costs.

Identity-based signature schemes can be built on top of public key cryptography standards that are widely used in today's network systems, such as RSA [3-5], ElGamal [6-7], and elliptic curve-based cryptosystems [8-14]. Elliptic curve-based signature schemes provide security equivalent to RSA and ElGamal-based signature schemes [15]. However, compared to RSA and ElGamal-based signature schemes, elliptic curve signature schemes have the distinct advantage of significantly smaller key sizes, while RSA and ElGamal-based signature schemes require much larger key sizes [15] along with higher computational costs [16-17]. Therefore, the signature scheme on elliptic curves is a suitable solution for devices with limited resources such as IoT devices, smart watches, ... In particular, identity-based signature schemes fully inherit the characteristics of short key size, efficiency, and high security when developed using

* Corresponding author: nghinv@actvn.edu.vn

elliptic curve standards. Therefore, the research and development of identity-based signature schemes on elliptic curves is not only of scientific significance but also of high practical value.

Numerous publications exist regarding digital signature schemes that utilize bilinear pairing mechanisms on elliptic curves [8-14]. Furthermore, there exist publications on digital signature schemes that do not employ bilinear pairing [18-20], ... Issues remain in establishing the security of IBS schemes that utilize bilinear pairing. Currently, there is no published standard for digital signature schemes utilizing bilinear pairing. Furthermore, digital signature schemes that utilize identification through bilinear pairing exhibit significant computational costs, primarily due to the processing time required for bilinear pairing operations. The computational cost of a bilinear pairing is approximately twenty times greater than that of scalar multiplication on an elliptic curve [21]. An identity-based digital signature scheme utilizing an elliptic curve without bilinear pairing provides enhanced computational efficiency and synchronization, rendering it appropriate for contemporary network system conditions.

Over the past decade, digital signature systems utilizing elliptic curves include EdDSA as specified in RFC 2016 [22] and GOST 34.10-2018 [23]. GOST 2018 is the latest standard established by the Russian Federation, serving as the national standard for digital signatures utilizing elliptic curve cryptography, superseding GOST R 34.10-2012. This standard aims to address the growing necessity for security in data processing systems, particularly in relation to the swiftly evolving computing technology. GOST 34.10-2018 delineates a secure and efficient digital signature framework predicated on the elliptic curve discrete logarithm problem (ECDLP) and the cryptographic hash function GOST R 34.11-2018 (Streebog).

GOST 34.10-2018 plays an important role in information protection and is applied in many fields, from e-commerce, banking, and finance to e-government and the military. This national standard has become a reliable cryptographic solution, applied not only in Russia but also in a number of countries with close, strategic relations with Russia, such as Armenia, Kyrgyzstan, and Tajikistan.

Therefore, in this paper, we propose an efficient and secure identity-based signature scheme based on the GOST 34.10-2018 standard. Currently, to our knowledge, there is no publication on the identity-based digital signature standard. The method employed for identifying the GOST 34.10-2018 scheme involves the bilinear unpairing technique, which ensures security and facilitates public verification through user identification information, including phone numbers and email addresses. The proposed scheme's security is assessed using the ROM security model and a heuristic method to address prevalent forgery attacks on identity-based digital signature schemes. This study evaluates the effectiveness of the proposed scheme by comparing the computational cost, measured in the number of mathematical operations, and the performance of experimental implementation on the same hardware platform against several other publications on IBS schemes utilizing elliptic curves. Our proposed scheme demonstrates significant security and efficiency, indicating substantial potential for application in network systems. This paper presents a specific application of the scheme within the domain of digital government.

The rest of this paper is organized as follows. Section 2 presents the relevant theoretical background. In Section 3, we propose a new identity-based digital signature scheme. Section 4 analyzes the security of this scheme. Next, Section 5 compares the computational cost and evaluates the performance of the proposed identity-based digital signature scheme. Section 6 presents the application of the proposed digital signature scheme in the digital government environment. Finally, Section 7 concludes.

2. MATHEMATICAL PRELIMINARY

In this section, we present digital signature schemes based on identification, elliptic curves, and the GOST 34.10-2018 digital signature standard. These are all theoretical foundations for the proposed scheme in this paper.

2.1. Identity-Based Signature

The identity-based digital signature scheme consists of 4 main processes (Setup, Key Generation, Signature Generation, and Signature Verification):

Setup: KGC executes this process to produce a master key pair: m_{pk} (master public key) and m_{sk} (master secret key). The m_{pk} is shared publicly, while the m_{sk} is kept confidentially by the KGC.

Key Generation: This process is executed by KGC, which takes as input parameters the user's ID and the public/private master key pair (m_{pk}, m_{sk}) generated during the setup phase. The output is the public/private key pair for the user: pk_{id} (user's public key) and sk_{id} (user's secret key). The ID key pair pk_{id} and sk_{id} are securely sent to the user.

Signature Generation: The signer executes this process by signing a message, utilizing the user's private key (sk_{id}), and the message (m) as inputs, and producing a digital signature (s) as output.

Signature Verification: The verification process is conducted by a verifier, which entails validating a digital signature using the message (m), the signer's public key (pk_{id}), and the digital signature (s) as inputs; the result is a determination of the digital signature's validity.

2.2. Elliptic Curve

$E_p(a, b)$ is the symbol for the elliptic curve over the finite field F_p. The equation of the elliptic curve in the Weierstrass form is as follows [24], [25]:

$$y^2 \equiv x^3 + ax + b \ (mod\ p) \tag{1}$$

where $a, b \in F_p$ and $4a^3 + 27b^2 \ mod\ p \neq 0$.

$E_p(a, b)$ is the set of points on the curve $y^2 \equiv x^3 + ax + b$. Point O is called the infinity point.

Point addition and scalar multiplication constitute the two fundamental operations of elliptic curve cryptography, including

Point addition: $R = P + Q$

Scalar multiplication: $Q = kP = P + P + \cdots + P$ (k times), where Q is the sum of k the point P, $k < p$.

Elliptic Curve Discrete Logarithm Problem (ECDLP): The security of Elliptic Curve Cryptography (ECC) is based on the difficulty of the discrete logarithm problem on elliptic curves. This problem is described as follows [24], [25]:

Given k and P, the calculation of Q is relatively straightforward. However, given P and Q, finding k is computationally infeasible. This is the complexity of the discrete logarithm function on an elliptic curve.

2.3. GOST 34.10-2018

The Russian standard GOST 34.10-2018 is a national cryptographic standard regulating the creation and verification of digital signatures in the field of information technology and cryptographic security [23]. This standard has been officially applied in Russia since June 1, 2019, and is an important part of the information security system of the Russian Federation.

The parameters of the GOST 34.10-2018 standard are shown in Table 1.

Table 1. Parameters in the GOST 34.10-2018 standard.

Notation	Meaning
H	Secure cryptographic hash function
p	Prime number $p > 3$
$E_p(a, b)$	An elliptic curve
M	Message
G	Base point of elliptic curve with degree n
d	Secret key (A random integer)

P	Public key (A point on the elliptic curve)
s	Digital signature of message M

GOST 34.10-2018 standard includes three main processes: key generation process, signature generation process and signature verification process.

a) Key generation process

This process uses domain parameters, including degree n and base point G, as input. The output of this process is the secret/public key pair d and P.

The process of the secret/public key pair generation is as follows:

1. Randomly generate the secret key d such that: $1 < d < n$.
2. Calculate the public key: $P = d \cdot G$.

b) Signature generation process

The signature generation process takes input parameters, including a message M and a private key d, and outputs the digital signature (r, s). The steps to create a signature for message M include:

1. Hash message M using a secure cryptographic hash function H: $h = H(M)\ mod\ n$. If $h = 0$, let $h = 1$.
2. Randomly generate a value k such that $0 < k < n$.
3. Calculate $Q = k \cdot G$ and let $r \equiv x_Q\ mod\ n$.

where x_Q is the x-coordinate of point Q. If $r = 0$, return to step 2.

4. Calculate $s \equiv (r \cdot d + k \cdot h)\ mod\ n$. If $s = 0$, return to step 2.
5. The signature for message M is (s, r).

c) Signature verification process

The signature verification process takes input parameters including a message M, a public key P, and a signature (r, s), and outputs a notification indicating whether the digital signature is valid or invalid. The steps to verify a digital signature include:

1. Check s and r. If $0 < r < n,\ 0 < s < n$, then proceed to the next step. Otherwise, the digital signature is invalid.
2. Hash message M using the secure cryptographic hash function H: $h = H(M)\ mod\ n$. If $h = 0$, let $h = 1$.
3. Calculate $v \equiv h^{-1}\ mod\ n$.
4. Calculate $z_1 \equiv s \cdot v\ mod\ n$ and $z_2 \equiv -r \cdot v\ mod\ n$.
5. Calculate $T = z_1 \cdot G + z_2 \cdot P$ and set $r' \equiv x_T\ mod\ n$ where x_T is the x-coordinate of point T.
6. Compare r' and r. If $r' = r$, the signature is valid; otherwise, the signature is invalid.

3. PROPOSED ID-BASED SIGNATURE SCHEME

In this section, we propose an identity-based signature scheme based on the Russian GOST 34.10-2018 digital signature standard.

3.1. Operation of Proposed Signature Scheme

The proposed signature scheme comprises four main algorithms: Setup, Key Generation, Signature Generation, and Signature Verification.

3.1.1. Setup

This process uses domain parameters similar to those of the GOST 34.10-2018 signature scheme (in Table 1, Section 2.3). In addition, there are other input parameters such as

- Master secret key: m_{sk}.
- Master public key: $m_{pk} = m_{sk} \cdot G$.

This master key pair (m_{sk}, m_{pk}) is issued by a trusted third-party Certificate Authority (CA) and granted to the organization via a certificate. The certificate is forwarded through a

secure channel, and the organization's network administrator is responsible for securely storing and managing this master key pair.

- Identity parameter id_A.

where $id_A = \{A@gmail.com, PhoneNumber\_A, Citizen\ ID\ Number\_A\}$. Hence, the identification parameters consist of 3 components:

+ User's email address ($A@gmail.com$).
+ User's phone number ($PhoneNumber\_A$).
+ Personal identification number ($Citizen\ ID\ Number\_A$).

3.1.2. Key Generation

This process takes m_{sk} and id_A as input parameters and outputs the user identification key pair: sk_A and pk_A. Generating a key pair based on user A's identity involves the following process:

1. Randomly generate $Nonce$ such that $0 < Nonce < n$.
2. Calculate $k_A = H(m_{sk} \parallel Nonce)\ mod\ n$.
3. Calculate $Q_A = k_A \cdot G$ and let $r_A \equiv x_{Q_A}\ mod\ n$.

where x_{Q_A} is x-coordinate of point Q_A. If $r_A = 0$, return to step 1.

4. Randomly generate value c such that $0 < c < n$.
5. Calculate $h_A = H(m_{sk} \parallel id_A \parallel c)\ mod\ n$. If $h_A = 0$ then let $h_A = 1$.
6. Calculate $sk_A \equiv (r_A \cdot m_{sk} + k_A \cdot h_A)\ mod\ n$. If $sk_A = 0$, return to step 1.
7. Calculate the public key of A: $pk_A = sk_A \cdot G$.

The private key of A is sk_A, and the public key of A is pk_A. This key pair is sent to user A via a physical transmission or through a secure channel. The remaining parameters $\{id_A, h_A, Q_A, pk_A\}$ are public.

3.1.3. Signature Generation

The signature generation process takes the input parameters M and sk_A and outputs the signature (s, r). The signature generation process is performed sequentially as follows:

1. Randomly generate $Nonce$ such that $0 < Nonce < n$.
2. Calculate $k = H(sk_A \parallel Nonce)\ mod\ n$.
3. Calculate $Q = k \cdot G$ and let $r \equiv x_Q\ mod\ n$.

where x_Q is the x-coordinate of point Q. If $r = 0$, return to step 1.

4. Calculate $h = H(M \parallel r)\ mod\ n$. If $h = 0$ then let $h = 1$.
5. Calculate $s \equiv (r \cdot sk_A + k \cdot h)\ mod\ n$. If $s = 0$, return to step 1.

The obtained signature for message M is (s, r).

3.1.4. Signature Verification

The signature verification process requires the input parameters message M, public key pk_A, and signature (s, r), and produces an output indicating the validity of the signature. The procedure for signature verification is as follows:

1. Verify that s and r belong to $(0, n)$. If satisfied, proceed to the next step. Otherwise, the signature is invalid and the verification process ends.
2. Calculate $h = H(M \parallel r)\ mod\ n$. If $h = 0$ then let $h = 1$.
3. Calculate $v \equiv h^{-1}\ mod\ n$.
4. Calculate $z_1 \equiv s \cdot v\ mod\ n$ and $z_2 \equiv -r \cdot v\ mod\ n$.
5. Calculate $T = z_1 \cdot G + z_2 \cdot pk_A$ and let $r' \equiv x_T\ mod\ n$.

where x_T is the x-coordinate of point T.

6. Compare r' and r. If $r' = r$, the signature is valid; otherwise, the signature is invalid.

3.2. The correctness of IBS-GOST 34.10-2018

The signature verification process with the public key pk_A yield a valid result using the signature (s, r) obtained from the IBS-GOST 34.10-2018 signing algorithm.

We have $s \equiv (r \cdot sk_A + k \cdot h)\ mod\ n$. With $k \in (0, n), r \equiv x_Q\ mod\ n$ and $h = H(M \parallel r)\ mod\ n$. Therefore, $k \equiv h^{-1} \cdot (s - r \cdot sk_A)\ mod\ n$.

When applying the signature verification algorithm, we have

$$T = z_1 \cdot G + z_2 \cdot pk_A = s \cdot v \cdot G - r \cdot v \cdot pk_A = v \cdot (s \cdot G - r \cdot pk_A)$$
$$= h^{-1} \cdot (s \cdot G - r \cdot sk_A \cdot G) = h^{-1} \cdot (s - r \cdot sk_A) \cdot G = k \cdot G$$

Thus, we have $r' \equiv x_T\ mod\ n \equiv x_{k \cdot G}\ mod\ n \equiv x_Q\ mod\ n = r$, so the signature (s, r) is valid, and the correctness of the proposed scheme IBS-GOST 34.10-2018 has been proven.

Furthermore, if we want to verify A's public key, we don't need to deploy a PKI infrastructure. Instead, the signature verifier can verify A's public key by checking the equation $pk_A = r_A \cdot m_{pk} + h_A \cdot Q_A$. If it is satisfied, A's public key is valid. Otherwise, A's public key is invalid.

4. SECURITY ANALYSIS

We evaluate the security of identity-based signature schemes in accordance with GOST 34.10-2018 utilizing the Random Oracle Model (ROM) predicated on the discrete logarithm problem on elliptic curves, employing heuristic methods to counteract various prevalent cyberattacks, specifically identity spoofing and signature spoofing attacks.

4.1. Security Analysis of the Proposed Scheme Based on the ROM Model

4.1.1. ECTEGTSS Scheme

In [26], E. F. Brickell et al. introduced two variants of the TEGTSS scheme (TEGTSS-I and TEGTSS-II) that attain "EUF-CMA security" within the ROM model. The authors employ a method to demonstrate EUF-NMA security and the presence of a simulator for this type of scheme to infer EUF-CMA security.

TEGTSS serves as a general framework for many "secure" digital signatures; however, its explanation has predominantly concentrated on the schema class derived from the DLP problem in FFC [26]. A variation of TEGTSS in ECC, designated ECTEGTSS, has been proposed by J. Malone-Lee and N. P. Smart [27]. Consequently, a schema assumes the structure ECTEGTSS if it satisfies

1. The basic group is an elliptic curve $E(F_p)$ with $\#E(F_p) = c \cdot n$ (where n is a large prime number and c is a small factor). A basic point $P \in E(F_p)$ has order n.
2. The scheme uses two functions, H and G, with corresponding feasible sets $\mathcal{H}$ and $\mathcal{G}$. In this scheme, security is considered with H assumed to be a random oracle and G satisfying the ℓ-collision-resistant or ℓ-non-collision property, for some integer $\ell \geq 2$.
3. There exist three functions:
$$F_1(\mathbb{Z}_n, \mathbb{Z}_n, \mathcal{G}, \mathcal{H}) \to \mathbb{Z}_n$$
$$F_2(\mathbb{Z}_n, \mathcal{G}, \mathcal{H}) \to \mathbb{Z}_n$$
$$F_3(\mathbb{Z}_n, \mathcal{G}, \mathcal{H}) \to \mathbb{Z}_n$$
satisfied with $\forall (k, d, r, h) \in (\mathbb{Z}_n, \mathbb{Z}_n, \mathcal{G}, \mathcal{H})$,
$$F_2(F_1(k, d, r, h), r, h) + a \cdot F_3(F_1(k, d, r, h), r, h) = k\ mod\ n.$$
4. Each user has a private key d and a public key $Q = dP$.
5. To sign a message m, first select $k \in_R \mathbb{Z}_n^*$, calculate $R = kP$ and $r = G(R)$. Subsequently, compute $h = H(m||r)$ and $s = F_1(k, d, r, h)$. The signature of m is (s, r, h), although in practice, (s, r) suffices, as h may be derived from m and r.
6. To verify whether the signature (s, r, h) is valid for message m, first calculate $e_P = F_2(s, r, h)$ and $e_Q = F_3(s, r, h)$, then calculate $W = e_P P + e_Q Q$. Finally, check $r \stackrel{?}{=} G(W)$ and $h \stackrel{?}{=} H(m||r)$.

7. The functions F_2 and F_3 must satisfy the following condition 1–1: given r, e_P, and e_Q, there exists a unique pair (h, s) such that $e_P = F_2(s, r, h)$, and $e_Q = F_3(s, r, h)$. Furthermore, this pair must be easy to find.

In fact, ECTEGTSS is the ECC version of TEGTSS-II, and this has also been confirmed by J. Malone-Lee and N. P. Smart [27]. Furthermore, these authors also give two variants of ECDSA that are in the form of ECTEGTSS, namely ECDSA-II and ECDSA-III, and indicate EUF-NMA security in the ROM model for them [27]. The basis for this result is the improved Forking Lemma [26], which is restated for ECTEGTSS as follows.

Lemma 1:

Let us consider a probabilistic polynomial-time Turing machine $\mathcal{A}$, called the attacker, and a probabilistic polynomial time simulator $\mathcal{B}$. If $\mathcal{A}$ can find with probability $\varepsilon > \frac{4}{p}$ a verifiable tuple (M, R, S, T, U) with less than $q_$ queries to the hash function, for a new message M and for a U directly defined by H, then with a constant probability $1/96$, with $(1 + 24q_\ell \log(2\ell))/\varepsilon$ replays of $\mathcal{A}$ and $\mathcal{B}$ with different random oracles, $\mathcal{A}$ will output $\ell + 1$ (where $\ell \leq \frac{\sqrt{p}}{4}$) verifiable tuples $(M_i, R_i, S_i, T_i, U_i)_{\{i=1,\dots,\ell+1\}}$ such that all the (M_i, T_i) equal for ECTEGTSS scheme.

Proof. Similar to the proof of the improved Forking Lemma for TEGTSS-II in [26].

Theorem 1:

With three functions

$$F_1: (\mathbb{Z}_n, \mathbb{Z}_n, \mathcal{G}, \mathcal{H}) \to \mathbb{Z}_n,$$
$$F_2: (\mathbb{Z}_n, \mathcal{G}, \mathcal{H}) \to \mathbb{Z}_n$$
$$F_3: (\mathbb{Z}_n, \mathcal{G}, \mathcal{H}) \to \mathbb{Z}_n$$

satisfying the requirements for the ECTEGTSS scheme, then function $\boldsymbol{F_1}$ satisfies the following relation: given $(\boldsymbol{b}, \boldsymbol{r}, \boldsymbol{h}) \in (\mathbb{Z}_{\boldsymbol{q}}, \boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{H}})$, with $\boldsymbol{a_1}, \boldsymbol{a_2} \in \mathbb{Z}_{\boldsymbol{q}}$, then $F_1(a_1, b, r, h) = F_1(a_2, b, r, h)$ if and only if $a_1 = a_2$.

Proof. Consider two cases:

$*$) If $a_1 = a_2$, it is easy to obtain $F_1(a_1, b, r, h) = F_1(a_2, b, r, h)$. Therefore, we only need to prove the reverse.

$*$) If $F_1(a_1, b, r, h) = F_1(a_2, b, r, h)$, combining the fact that F_1, F_2, F_3 satisfy the requirements of the TEGTSS scheme, then we have:

$$a_1 \bmod n = F_2(F_1(a_1, b, r, h), r, h) + b \cdot F_3(F_1(a_1, b, r, h), r, h)$$
$$= F_2(F_1(a_2, b, r, h), r, h) + b \cdot F_3(F_1(a_2, b, r, h), r, h) = a_2 \bmod n.$$

From the two cases above, the proof of the theorem is complete.

4.1.2. The Existence of Simulators for ECTEGTSS

Lemma 2:

Let H be an ideal random function with outputs in the range $\{0, \dots, |\mathcal{H}| - 1\}$. There exists a simulator $\mathcal{B}$ that generates valid sets such that after b steps, the probability of distinguishing $\mathcal{B}$ from an ECTEGTSS set is less than $b^2/(2n)$.

Proof. Given a message M to be signed, the simulator $\mathcal{B}$ operates as follows: Select $\alpha, \beta \in_R \mathbb{Z}_n$, calculate $W = \alpha P + \beta Q \bmod n$, and $r = G(W)$. According to the properties of F_2 and F_3, we have s and $h \bmod n$ determined as unique values from the system of two equations $F_2(s, r, h) = \alpha$ and $F_3(s, r, h) = \beta$, except for negligible probability. Finally, $H(M, r)$ is defined by h.

In this way, the simulator generate the sets (W, s, r, h) corresponding to the requested message M. Let's consider the following two distributions:

$$A = \left\{ (W,s,r,h) \middle| \begin{array}{c} k \in_R \mathbb{Z}_n^* \\ h \in_R \{0,\dots,|\mathcal{H}|-1\} \\ W = kP \\ r = G(W) \\ s = F_1(k,X,r,h) \end{array} \right\}$$

and

$$B = \left\{ (W,s,r,h) \middle| \begin{array}{c} \alpha \in_R \mathbb{Z}_n \\ \beta \in_R \mathbb{Z}_n \\ W = \alpha P + \beta Q, \text{sao cho } W \neq \mathcal{O} \\ h,s \text{ are the solutions of the system } \begin{cases} F_2(s,r,h) = \alpha \\ F_3(s,r,h) = \beta \end{cases} \end{array} \right\}$$

It is important to note that distributions A and B are indistinguishable from one another.

Indeed, with a tuple (W_0, s_0, r_0, h_0) chosen in such a way that $\alpha_0, \beta_0 \in_R \mathbb{Z}_q$, $W_0 = \alpha_0 P + \beta_0 Q$ (such that $W_0 \neq \mathcal{O}$), $r_0 = G(W_0)$, and s_0, h_0 are determined as unique values from the system of two equations $F_2(s_0, r_0, h_0) = \alpha_0$ and $F_3(s_0, r_0, h_0) = \beta_0$. Then we consider the possibility that a tuple (W,s,r,h) chosen according to distribution A will coincide with (W_0, s_0, r_0, h_0), that is, $(W,s,r,h) = (W_0, s_0, r_0, h_0)$. This is equivalent to the possibility $(\alpha, \beta) = (\alpha_0, \beta_0)$, where (α, β) is the pair that generates the tuple (R,S,T,U) according to the distribution B (a way of simulating $\mathcal{B}$'s signature). Therefore,

$$\Pr_B[(W,s,r,h) = (W_0, s_0, r_0, h_0)] = \Pr_{\alpha,\beta}[\alpha = \alpha_0, \beta = \beta_0, W = W_0 \neq \mathcal{O}] = \frac{1}{n(n-1)}.$$

According to Theorem 1, we obtain a tuple (W,s,r,h) chosen according to distribution A (a method for simulating $\mathcal{A}$'s signature) that aligns with (W_0, s_0, r_0, h_0), which is equivalent to $(k,h) = (\alpha_0 + X\beta_0, h_0)$, where k,h represents the pair that generates the tuple (W,s,r,h) in accordance with distribution A. Let $|\mathcal{H}| = an + z$, where $a, z \in \mathbb{N}$ and $0 < z < n$. Assume $z < n/2$ (the case $z > n/2$ is handled similarly). If $0 \leq (h_0 \bmod n) \leq z - 1$, then:

$$\Pr_A[(W,s,r,h) = (W_0, s_0, r_0, h_0)] = \Pr_{k \neq 0, U}[k = \alpha_0 + X\beta_0, h = h_0 \bmod n] = \frac{a+1}{(n-1)|\mathcal{H}|}.$$

If $z \leq (h_0 \bmod n) \leq n - 1$, then

$$\Pr_A[(W,s,r,h) = (W_0, s_0, r_0, h_0)] = \Pr_{k \neq 0, U}[k = \alpha_0 + X\beta_0, h = h_0 \bmod n] = \frac{a}{(n-1)|\mathcal{H}|}.$$

Let $C = \{(W,s,r,h) \text{ be taken according to the distribution } A | 0 \leq h \bmod n \leq z-1\}$, we have

$$\begin{aligned} \varepsilon &= \sum_{(R_0,T_0,S_0,U_0)} \left| \Pr_A[(W,s,r,h) = (W_0,s_0,r_0,h_0)] - \Pr_B[(W,s,r,h) = (W_0,s_0,r_0,h_0)] \right| \\ &= \sum_{(R_0,T_0,S_0,U_0) \in C} \left| \Pr_A[(W,s,r,h) = (W_0,s_0,r_0,h_0)] - \Pr_B[(W,s,r,h) = (W_0,s_0,r_0,h_0)] \right| \\ &+ \sum_{(R_0,T_0,S_0,U_0) \notin C} \left| \Pr_A[(W,s,r,h) = (W_0,s_0,r_0,h_0)] - \Pr_B[(W,s,r,h) = (W_0,s_0,r_0,h_0)] \right| \\ &= z(n-1) \left| \frac{1}{n(n-1)} - \frac{a+1}{(n-1)|\mathcal{H}|} \right| + (n-z)(n-1) \left| \frac{1}{n(n-1)} - \frac{a}{(n-1)|\mathcal{H}|} \right| \\ &= z \cdot \left| \frac{|\mathcal{H}| - (a+1)n}{n|\mathcal{H}|} \right| + (n-z) \left| \frac{|\mathcal{H}| - an}{n|\mathcal{H}|} \right| = \frac{2z(n-z)}{n|\mathcal{H}|} < \frac{2z}{|\mathcal{H}|}. \end{aligned}$$

This indicates that the two distributions, A and B, are statistically indistinguishable. Since H is a random oracle, the tuple (W,s,r,h) generated according to distribution A is similar to the actual generation of the ECTEGTSS signature, except that $H(M,r)$ is predefined. Therefore, the adversary can only distinguish the distribution of the tuples (W,s,r,h) generated by the simulator $\mathcal{B}$ and the ECTEGTSS signature generator if $\mathcal{B}$ calculates (M,r) but $H(M,r)$ is predefined or (M,r) has already been calculated by the signature generator.

With b being the number of queries made to the oracle H (including both direct and indirect cases). The probability that one of the above events occurs is less than $1 - e^{-\frac{b(b-1)}{2n}} < \frac{b^2}{2n}$.

Consequently, based on [26], [28], [29], if a "simulator" $\mathcal{B}$ (utilizing solely public information) of the signing process in the digital signature scheme exists, then the EUF-CMA security of the signature scheme can be diminished to EUF-NMA security. This reduction is understandable because the signatures that an attacker would obtain from the signing process in the digital signature scheme under the CMA scenario could be replaced by signatures returned from $\mathcal{B}$, which would be difficult to detect. In such a case, what the attacker obtains is based entirely on public information and is equivalent to an NMA attack.

Therefore, the existence of a simulator for the ECTEGTSS scheme allows us to conclude that if an ECTEGTSS scheme is EUF-NMA security, then it is also EUF-CMA security.

4.1.3. Provable Security of the Proposed Scheme in the ROM Model

Theorem 2:

The IBS-GOST 34.10-2018 signature scheme is in the form of ECTEGTSS.

Proof. We show that IBS-GOST 34.10-2018 satisfies requirements (i) to (vii) of an ECTEGTSS signature scheme. Indeed,

i. The basic group of IBS-GOST 34.10-2018 is based on an elliptic curve $E(F_p)$ such that $\#E(F_p) = c \cdot n$, where n represents a high prime number and c denotes a minor factor, with the basis point $G \in E(F_p)$ having degree n. By assumption, the criteria for prime numbers p and n in IBS-GOST 34.10-2018 are as follows: case 1, p is 256 bits and $2^{254} < n < 2^{256}$; case 2, p is 512 bits and $2^{508} < n < 2^{512}$. Conversely, according to Hasse's Theorem, $p - 2\sqrt{p} + 1 < \#E(F_p) < p + 2\sqrt{p} + 1$. Therefore, in IBS-GOST 34.10-2018, with $\#E(F_p) = c \cdot n$, the value of c does not surpass 16. Thus, IBS-GOST 34.10-2018 fulfills the requirement (i).

ii. IBS-GOST 34.10-2018 employs the hash function H and the function $G(Q) = x_Q \bmod n$ (where $Q \in E(F_p)$). In the ROM model, H_{GOST} is conceptualized as a random oracle. It needs to show that G fulfills the ℓ-collision-resistant or ℓ-non-collision feature. As stated in [27], if integers $k_1, k_2, \dots, k_{c+1}$ exist such that $G(k_1 P) = G(k_2 P) = \cdots = G(k_{c+1} P)$, then there must be two distinct indices $i, j \in \{1, 2, \dots, c+1\}$ where $i \neq j$ such that $k_i = \pm k_j$. This indicates that G is a $2c + 1$ non-collision graph. In accordance with the section (i), the value of c in IBS-GOST 34.10-2018 does not exceed 16; hence, G is classified as 33-non-collision.

iii. IBS-GOST 34.10-2018 satisfies requirement (iii). Indeed, consider $F_1(k, sk_A, r, h) = hk + sk_A r$, $F_2(s, r, h) = sh^{-1}$, $F_3(s, r, h) = -rh^{-1}$.We have:

$$F_2(F_1(k, sk_A, r, h), r, h) + sk_A \cdot F_3(F_1(k, sk_A, r, h), r, h)$$
$$= F_1(k, sk_A, r, h)h^{-1} - sk_A r h^{-1} = (hk + sk_A r)h^{-1} - sk_A r h^{-1} = k$$

Therefore, IBS-GOST 34.10-2018 satisfies requirement (iii).

IBS-GOST 34.10-2018 satisfies requirements (iv) to (vii) thanks to the description of this scheme and the definition of functions F_1, F_2, and F_3.

Thus, it can be concluded that the IBS-GOST 34.10-2018 signature scheme has the form ECTEGTSS.

Theorem 3:

Assume there exists an adversary $\mathcal{A}$ capable of forging the signature of the IBS-GOST 34.10-2018 scheme with a success probability of $\varepsilon > \frac{4}{p}$ after q_h queries to the random oracle H; hence, the ECDLP in $E(\mathbb{F}_p)$ can be resolved using

$$\frac{1 + 768 q_h \log_2 64}{\varepsilon} = \frac{1 + 4608 q_h}{\varepsilon}$$

of $\mathcal{A}$'s iterations with a probability exceeding $1/100$.

Proof. According to Theorem 2, IBS-GOST 34.10-2018 is equivalent to the ECTEGTSS type. Using Lemma 1 and applying it with parameters including: message M, $S = s$, $T = r$, $U = H_{GOST}(M, r) = h$, and $\ell = 32$. Then, with the number of iterations of $\mathcal{A}$ being

$$\frac{1 + 768q_h \log_2 64}{\varepsilon} = \frac{1 + 4608q_h}{\varepsilon}$$

$\mathcal{A}$ generates 33 distinct valid signatures $(M, R_i, S_i, T_i, U_i)_{i=1,\dots,33}$ for a specific message M, where $T = r$ and each U_i represents different hash values ($U_i = H^{(i)}(M, T) = h_i$, employing various hash functions), with a probability of $1/96$, which exceeds $1/100$. These signatures correspond to 33 points, $R_1 = k_1 P, \dots, R_{33} = k_{33} P$; hence, we obtain $G(R_1) = \cdots = G(R_{33}) = r$. Nonetheless, given that G is 33-collision-resistant (indicating a maximum of 32 different points yielding identical values), it follows that at least two of those 33 points must be similar. Assuming without loss of generality that $Q_1 \equiv Q_2$ implies that the random variables k_1 and k_2 associated with these two points are also equivalent.

$$k_1 = k_2 \bmod n \Leftrightarrow s_1(h_1 + rsk_A) = s_2(h_2 + rsk_A) \bmod n$$
$$\Leftrightarrow sk_A = r^{-1}(s_1 - s_2)^{-1}(h_2 s_2 - h_1 s_1) \bmod n.$$

Consequently, we acquire the signer's private key, thereby resolving the ECDLP. Since $h_1 \neq h_2 \bmod n$, we have $s_1 \neq s_2 \bmod n$.

Given that EUF-NMA security can be deduced from EUF-CMA security for signature schemes of the ECTEGTSS type, we conclude that IBS-GOST 34.10-2018 attains EUF-CMA security.

Theorem 4:

Consider an attacker $\mathcal{A}$ against IBS-GOST 34.10-2018, and assume that $\mathcal{A}$ is capable of generating a forgery that exists under a CMA attack with probability $\varepsilon > 4/p$ after q_h queries to function H; then the ECDLP in $E(\mathbb{F}_p)$ can be solved by using

$$(1 + 4608q_h)/\varepsilon$$

of $\mathcal{A}$'s iterations with probability greater than $1/100$.

Proof. This is a direct result of combining Lemma 2 and Theorem 3.

Therefore, the IBS-GOST 34.10-2018 scheme achieves EUF-CMA security in the ROM model.

4.2. Security Analysis of the Proposed Scheme Using Heuristic Methods

4.2.1. Identity Forgery Attack

Suppose attacker C attempts to perform an identity spoofing attack against legitimate user A. Once C obtains A's identity id_A, a spoofing attack is possible if the Key Generation Center (KGC) lacks robust authentication mechanisms before issuing private keys to users. If the spoofing attack is successful, attacker C can sign arbitrary documents in the name of user A, leading to serious consequences for the authenticity and integrity of the system.

To mitigate this identity spoofing attack, the proposed solution is to verify user identity through the email address/phone number previously registered with KGC before issuing the identity-based private key. This solution includes the following steps:

1. User A requests KGC to generate a private key pair corresponding to the identity id_A.
2. KGC compares id_A with $id_{A'}$. If $id_A = id_A{}'$, it proceeds to step 3; otherwise, it stops and the private key generation process ends.
   Where $id_A{}'$ is the identity of A stored in KGC's database.
3. KGC sends a verification code via A's email/phone number.
4. User A enters the verification code accurately.
5. KGC then generates and issues the identifier key pair to user A.

In this case, C knows A's identity id_A. C then sends a request to KGC for the allocation of a private key pair based on id_A. However, because KGC uses multi-factor authentication, the verification code is only sent to A's registered email address or phone number, so the attacker, C, cannot know this verification code. Therefore, C's request for the allocation of a private key pair based on id_A is rejected. Consequently, C's attempt to impersonate A's identity is unsuccessful.

Based on the above arguments, our proposed digital signature scheme is secure against identity spoofing attacks.

4.2.2. Signature Forgery Attack

Assume that there are two entities involved in the signature generation and electronic document exchange process: user A and user B. A is the signer, and B is the verifier. The signature scheme used is an identity-based signature scheme according to the GOST 34.10-2018 standard.

Suppose attacker C forges the signature of legitimate user A. The signature forgery process unfolds as follows:

1. C randomly generates a forged key pair consisting of sk_C and pk_C. The private key is sk_C and the public key is $pk_C = sk_C \cdot G$.
2. Calculate the values k', h', Q', r'.
3. Calculate the signature (s', r') based on the parameters h', k', Q', r' and sk_C.
4. C sends the signature (s', r') to B in the name of A.
5. B verifies the signature (s', r').

B uses the public key pk_A to verify the signature (s', r'). This process includes the following steps:

1. Check $0 < s' < n$ and $0 < r' < n$.
2. Calculate $h = H(M)\ mod\ n$. If $h = 0$, let $h = 1$.
3. Calculate $z_1 \equiv s' \cdot h^{-1}\ mod\ n$ and $z_2 \equiv -r' \cdot h^{-1}\ mod\ n$.
4. Calculate $T = z_1 \cdot G + z_2 \cdot pk_A$ and let $R' \equiv x_T\ mod\ n$.
5. Compare R' and r'.

The forged key pair (pk_C, sk_C) by C cannot correspond with A's identification key pair (pk_A, sk_A), as (pk_A, sk_A) is derived from the organization's secret master key (m_{sk}), which C cannot access. Consequently, $pk_C \neq pk_A$, resulting in the condition $x_T \neq r'$ or $R \neq r'$. The value of R' consistently differs from r', leading B to determine that the signature (s', r') is a forgery, not A's signature. Consequently, C's attempt to forge A's signature is unsuccessful.

To compute the secret master key m_{sk}, C rely on the equation $m_{pk} = m_{sk} \cdot G$. Deriving m_{sk} from m_{pk} is equivalent to solving the discrete logarithm problem on an elliptic curve over a finite field F_p. This is a difficult problem, impossible to solve in polynomial time when the length of p is sufficiently large.

Therefore, the proposed signature scheme is secure against signature forgery attack.

5. PERFORMANCE ANALYSIS

In this section we compare the computational cost and performance evaluation of the IBS-GOST 34.10-2018 signature scheme with several other published identity-based digital signature works [10], [11], [18], [30], [31].

The table below shows the notations for several arithmetic operations utilized in digital signature schemes:

Table 2. Running time of operations

Notation	Operation	Running time (ms)
H	Hash Operation	1.105
PM	Point Multiplication	5.266
PA	Point Addition	1.343

MM	Modular Multiplication	0.716
MA	Modular Addition	0.314
MI	Modular Inversion	2.601
Ex	Exponentiation	10.804
e	Bilinear Pairing Operation	19.297

The comparison of computational costs between these digital signature schemes is essentially a comparison of the number of operations performed in each scheme. We evaluated the experimental performance of these operations on the same hardware platform: a laptop with an Intel Core i7-8750H 2.20 GHz processor, 16 GB of RAM, and Windows 11. The hash function used was SHA-256. The results of the comparison of computational costs and experimental performance of several identity-based signature schemes are presented in Table 3.

Table 3. Comparison of computational complexity

Scheme's	Setup	Extract	Sign	Verify	Total (ms)
K. G. Paterson 2002[10]	1PM ≈ 5.266	1H+1PM ≈ 6.371	2H+4PM+ 1PA+1MI ≈ 27.218	2H+3e ≈ 60.101	98.956
Florian Hess 2003[11]	1PM ≈ 5.266	1H+1PM ≈ 6.371	1H+2PM+1PA+ 1e+1Ex ≈ 43.081	1H+2e+1Ex ≈ 50.503	105.221
H. Jin et al. 2010 [18]	1PM ≈ 5.266	1H+1PM+ 1MM+1MA ≈ 7.401	1H+1PM+2MM+ 1MA+1MI ≈10.718	2H+3PM+2PA+ 2MM+1MI ≈ 24.727	48.112
G.Sharma et al. 2017 [30]	1PM ≈ 5.266	1H+3PM+1PA+ 1MM+1MA ≈ 19.276	1H+1PM+ 1MM+1MA ≈ 7.401	1H+2PM+2PA ≈ 14.323	46.266
N.Sharma et al. 2018 [31]	1PM ≈ 5.266	1H+1PM ≈ 6.371	2PM+1PA+ 1e+1Ex ≈ 41.976	1PA+1e+1Ex ≈ 31.444	85.057
Our Scheme	--	2H+2PM+ 2MM+1MA ≈ 14.488	2H+1PM+ 2MM+1MA ≈ 9.222	1H+2PM+1PA+ 2MM+1MI ≈ 17.013	40.723

The results of Table 3 show that the proposed digital signature scheme has better computational efficiency than other identity-based digital signature schemes. Thus, the proposed IB-GOST 34.10-2018 scheme has good experimental performance and has great potential for application in electronic document management systems.

6. APPLICATIONS IN DIGITAL GOVERNMENT

In this section, we propose to apply the IBS scheme according to the GOST 34.10-2018 standard into the digital government model.

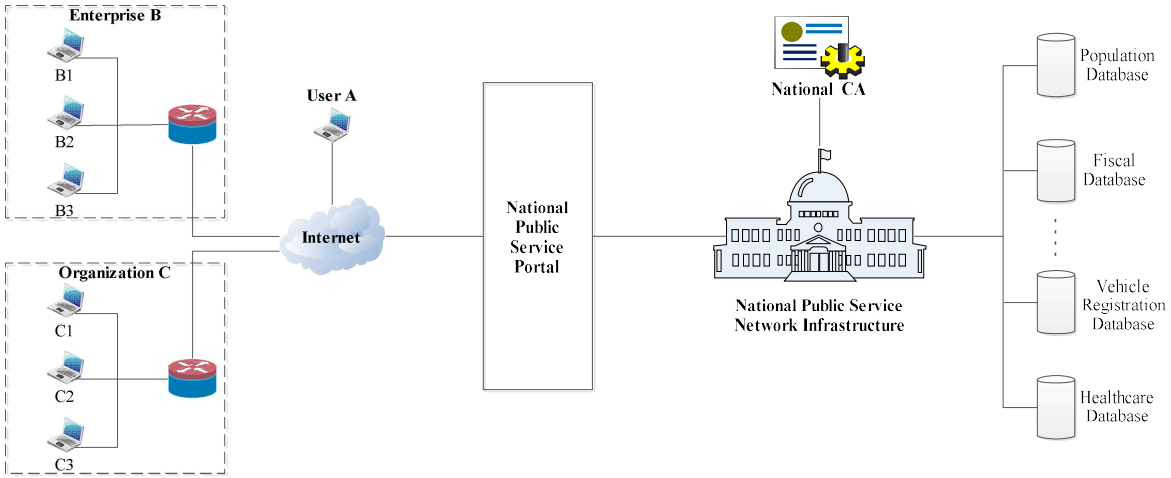

Fig. 1. Digital government model

Figure 1 illustrates a digital government paradigm comprising the following components:

- The user layer includes citizens, enterprises, and organizations. These are the main components that use online public services through digital platforms and are the direct beneficiaries of these public services.
- Internet infrastructure acts as a bridge, connecting users (citizens, businesses, and organizations) to public service servers.
- The National Public Service Portal is a central online platform that connects citizens, businesses, and organizations with government agencies to carry out administrative procedures and public services online in a digital environment.
- National CA plays a central role in issuing and managing digital certificates, providing digital signature authentication services to help verify electronic identities, and ensuring legal validity and information security in electronic transactions.
- National Public Service Network Infrastructure (NPSNI) is a crucial component of the digital government architecture. It is a network system for information technology designed and deployed to ensure secure, continuous, and efficient data exchange and information sharing within the digital government. Within the NPSNI, numerous cryptographic methods and security technologies are employed to guarantee data confidentiality, integrity, and authenticity. These include encryption, hashing, digital signatures, VPNs, firewalls, and network security monitoring.
- The database layer plays a crucial role in storing, managing, and sharing information within a digital government. Types of databases that can be used include national population databases, business databases, insurance databases, tax databases, and health and education databases.

The process of issuing identity-based private keys is handled by the National Certificate Authority (National CA). This process takes as input the National Certificate Authority's master key pair (m_{sk} - master secret key, m_{pk} - master public key) and the user's identification information (personal identification number on the citizen ID card). The output is a public/private key pair for individuals, businesses, or organizations. The identity-based private key pair is stored in the chip-embedded citizen ID card or the VNeID electronic identification software. Details on the algorithm for generating identity-based private keys for individuals, businesses, or organizations are presented in section 3b.

After obtaining a unique identification key pair, individuals, businesses, or organizations can conduct electronic transactions within the digital government environment and sign electronic documents using the IBS-GOST 34.10-2018 algorithm. Details of the IBS-GOST 34.10-2018 signature algorithm are presented in section 3c.

The verifier uses the signature verification algorithm outlined in section 3d to determine the validity of the signature.

Digital signature schemes based on the GOST 34.10-2018 standard offer a promising and effective solution at a low cost in a digital government environment. Citizens, businesses, and organizations can carry out legal procedures and transactions on the national public service portal using the IBS-GOST 34.10-2018 digital signature algorithm in many specific cases, such as issuing ordinary passports domestically, registering for enrollment in primary, secondary, and high schools, applying for college and university admissions, paying personal and corporate taxes, issuing electronic invoices, etc.

This paper presents the application of the IBS-GOST 34.10-2018 digital signature scheme in the process of issuing electronic invoices. In the electronic invoice issuance process, multiple parties are involved to ensure the legality, authenticity, and transparency of the invoice. These parties include the invoice issuer (seller), the invoice recipient (buyer), and the tax authority.

- Invoice issuer: This refers to the organization or individual that sells goods or provides services and is responsible for creating and issuing electronic invoices.

- The recipient of the invoice: This refers to the organization or individual that purchases the goods or services.

- The tax authority is the governmental agency responsible for receiving, managing, and overseeing the utilization of electronic invoices, as well as enforcing norms and regulations pertaining to them.

When a transaction occurs between a seller and a buyer, the seller uses electronic invoicing software to create an electronic invoice and digitally signs it. The seller then sends the created invoice to the tax authority, which provides an authentication code for the electronic invoice. This code is then attached to the invoice to ensure its validity and authenticity. Finally, the seller sends the electronic invoice to the buyer.

The tax authorities archive all electronic invoices in the General Department of Taxation's database within the government's system to facilitate cross-verification and validation of the invoices' legitimacy. Purchasers can access invoices on the General Department of Taxation's portal.

While the Vietnamese law does not require buyers to digitally sign electronic invoices, it simplifies the transaction process and saves costs. However, it also creates several security vulnerabilities and potential risks, such as loopholes in verifying the validity of transactions and an increased risk of tax fraud, as sellers can issue fictitious invoices to claim value-added tax deductions.

When the buyer does not sign the invoice, there is no digital proof that the buyer actually received the goods/services and agreed to the invoice's contents. This makes it possible for the invoice to be fictitious or contain incorrect information without the buyer's knowledge.

Sellers may generate fraudulent invoices for a "shell" company or a legitimate company that engages in no genuine transactions. They subsequently utilize these invoices to offset value-added tax, resulting in deficits to the state budget.

In some cases, third parties (intermediaries, hackers, or internal employees) can create fake electronic invoices, impersonating the seller and sending them to the buyer, requesting payment, which leads to financial and legal risks.

To minimize these risks, not only sellers but also buyers should digitally sign electronic invoices. However, this leads to the problem that buyers need to have their key pair and purchase a digital certificate from a trusted third party, which adds extra costs for the buyer.

The current proposal is to utilize the IBS-GOST 34.10-2018 digital signature method for signing electronic invoices; the private key pair is derived from the identification information of individuals (personal identification number on the Citizen ID). Consequently, any purchaser can electronically sign the invoice without incurring extra costs for acquiring or renewing the digital certificate. The IBS-GOST 34.10-2018 digital signature solution minimizes users' reliance on digital certificates for digital signature services, hence facilitating cost savings.

7. CONCLUSION

This work presents an identity-based signature scheme, IBS-GOST 34.10-2018, based on the GOST 34.10-2018 standard utilizing elliptic curves. The suggested digital signature method is demonstrated to be secure within the ROM model, effectively rejecting identity fraud and signature forgery attacks. The IBS-GOST 34.10-2018 scheme demonstrates outstanding potential and applicability for authenticating electronic documents in the digital government context, owing to its enhanced security and reduced computational costs relative to similar identity-based signature schemes discussed in the paper.

ACKNOWLEDGEMENTS

This work has been supported by Academy of Cryptography Techniques, Vietnam under Project/Lab.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

[1] Shamir, A. (1985). Identity Based Cryptosystems and Signature Schemes, *Proc. of Advances in Cryptology (CRYPTO' 84)* (Santa Barbara, CA), 47–53.

[2] Fiat, A. & Shamir, A. (1987). How to prove yourself: Practical solutions to identification and signature problems, *Proc. of Advances in Cryptology (CRYPTO' 86)* (Santa Barbara, CA), 186–194.

[3] Bellare, M. & Neven, G. (2006). Identity-Based Multi-signatures from RSA, *Proc. of Topics in Cryptology (CT-RSA 2007)* (San Fancisco, CA), 145–162.

[4] Lein, H. & Jian, R. (2008). Efficient identity-based RSA multisignatures, *Computers and Security*, **27**(1–2), 12–15.

[5] Ko, H., et. al. (2019). Forward Secure Identity-based Signature Scheme with RSA, *Proc. of ICT Systems Security and Privacy Protection (IFIP)* (Lisbon, Portugal), 314–327.

[6] Said, K., Kamer, K. & Ali, A. S. (2007). Generalized ID-based ElGamal signatures, *Proc. of 2007 22nd International Symposium on Computer and Information Sciences* (Ankara, Turkey), 1–6.

[7] Wang, S., Yu, H. & Liu, D. (2018). A new identity based blind signature scheme and its application. *Proc. 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference*. Chongqing, China, 672–676.

[8] Choon, J.C., Hee Cheon, J. (2003). An Identity-Based Signature from Gap Diffie-Hellman Groups, *Proc. of Public Key Cryptography (PKC 2003)* (Miami, FL), 18–30.

[9] Boneh, D., Lynn, B. & Shacham, H. (2001). Short Signatures from the Weil Pairing, *Proc. of Advances in cryptology (ASIACRYPT 2001)* (Gold Coast, Australia), 514–532.

[10] Paterson, K. G. (2002). ID-based Signatures from Pairings on Elliptic Curves, *Electronics Letters*, **38**(18), 1025–1026.

[11] Hess, F. (2003). Efficient Identity Based Signature Schemes Based on Pairings, *Proc. of Selected Areas in Cryptography (SAC 2002)* (Newfoundland, Canada), 310–324.

[12] Du, H. &Wen, Q. H. (2007). An Efficient Identity-Based Short Signature Scheme from Bilinear Pairings, *Proc. of 2007 International Conference on Computational Intelligence and Security (CIS 2007)* (Harbin, Heilongjiang, China), 725–729.

[13] Zhang, L., Hu, Y. & Wu, Q. (2010). Short Signature from the Bilinear Pairing, *Proc. of Information Computing and Applications* (Tangshan, China), 111–118.

[14] Raylin, T., Takeshi, O. & Eiji, O. (2009). Efficient Short Signatures from Pairing. *Proc. of 2009 Sixth International Conference on Information Technology: New Generations*. (Las Vegas, NV), 417–422.

[15] NIST (2020). *Cryptographic Key Length Recommendation*, [Online]. Available https://www.keylength.com/en/4/

[16] Jansma, N. & Arrendondo, B. (2004). *Performance Comparison of Elliptic Curve and RSA Digital Signatures*, [Online]. Available https://fog.misty.com/perry/ccs/ec /performance_comparison_of_elliptic_curve_and_rsa_digital_signatures.pdf.

[17] Saho, N. J. G. & Ezin, E. C. (2020). *Comparative Study on the Performance of Elliptic Curve Cryptography Algorithms with Cryptography through RSA Algorithm*, [Online]. Available https://hal.science/hal-02926106v1/document.

[18] Jin, H., Debiao, H. & Jianhua, C. (2010). An Identity Based Digital Signature from ECDSA, *Proc. of 2010 Second International Workshop on Education Technology and Computer Science* (Wuhan, China), 627–630.

[19] Galindo, D. & Garcia, F. D. (2009). A Schnorr-Like Lightweight Identity-Based Signature Scheme, *Proc. of Progress in Cryptology (AFRICACRYPT 2009)* (Gammarth, Tunisia), 135–148.

[20] M. T. Le, et. al. (2026). Digital Document Signing Solution Without PKI Infrastructure for Small and Medium Enterprises, *Proc. of International Conference on Computational Intelligence in Engineering Science* (Ho Chi Minh City, Vietnam), 159–170.

[21] Chen, L., Cheng, Z. & Smart, N. P. (2007). Identity-based Key Agreement Protocols from Pairings, *International Journal of Information Security*, **6**, 213–241.

[22] Josefsson, S. & Liusvaara, I. (2016). *Edwards-Curve Digital Signature Algorithm (EdDSA)*, *RFC 8032*, [Online]. Available https://datatracker.ietf.org/doc/html/rfc8032.

[23] Interstate Council for Standardization, Metrology and Certification (2018). *GOST 34.10-2018, Information technology. Cryptographic data security. Signature and verififcation processes of electronic digital signature*, [Online]. Available https://docs.cntd.ru/document/1200161706

[24] Nakov, S. (2018) *Practical Cryptography for Developers*, [Online]. Available https://cryptobook.nakov.com/

[25] Stalling, W. (2017) *Cryptography and network security* 7th edition, [Online]. Available https://staff.ustc.edu.cn/~mfy/moderncrypto/crypto7ed.pdf

[26] Brickell, E., Pointcheval, D., Vaudenay, S. & Yung, M. (2000). Design validations for discrete logarithm based signature schemes, *Proc. of International Workshop on Public Key Cryptography* (Melbourne, Victoria, Australia), 276–292.

[27] Malone-Lee, J. & Smart, N. P. (2003). Modifications of ECDSA, *Proc. of International Workshop on Selected Areas in Cryptography* (Newfoundland, Canada), 1–12.

[28]. Pointcheval, D. & Stern, J. (1996). Security proofs for signature schemes, *Proc. of the 15th Annual international conference on Theory and application of cryptographic techniques (EUROCRYPT'96)* (Saragossa, Spain), 387–398.

[29]. Pointcheval, D. & Stern, J. (2000). Security arguments for digital signatures and blind signatures, *Journal of cryptology*, **13**(3), 361–396.

[30] Sharma, G., Bala, S. & Verma, A. K. (2017). PF-IBS: Pairing-Free Identity Based Digital Signature Algorithm For Wireless Sensor Networks, *Wireless personal communications*, **97**, 1185–1196.

[31] Sharma, N. & Sharma, B. K. (2018). Identity-Based Signature Scheme Using Random Oracle Model, *Journal of Computer and Mathematical Sciences*, **9**(4), 254–263.

