A Lightweight Provably Secure Digital Short-Signature Technique Using Extended Chaotic Maps for Human-Centered IoT Systems

Chandrashekhar Meshram, Mohammad S. Obaidat, *Fellow, IEEE*, Jitendra V. Tembhurne, Shailendra W. Shende, Kailash W. Kalare, and Sarita Gajbhiye Meshram

***Abstract*—Internet of Things (IoT) consists of numerous smart devices for sharing sensed data through the availability of online services. Direct communication by smart devices with people to identify parameters of healthcare and send them to a central repository is crucial. There is a need to secure messages among the sender and recipient during data exchange in order to tackle the malicious attacks by human. To provide secure communication, various signature-based schemes are presented in the literature. However, smart devices require lightweight tasks by guaranteeing essential security strengths. The main difficulty in signature-based methods is more computational cost incurred for signature and verification stages involving large numbers. This article introduces a lightweight provably secure short digital signature technique for safe communication amongst smart devices in human-centered IoT (HCIoT), the security of which is closely related to an extended chaotic maps assumption in a random oracle model (ROM). Moreover, we used less comprehensive operations to accomplish processes of verification and signing, similar to human signing on legitimate documents and then check as per witness. The proposed technique provides a stronger guarantee of protection than existing signature techniques. The key advantage of the presented technique over the DSA techniques is that it takes less computation in the verification stage and signing length; it retains the degree of protection. The presented short signature takes less bandwidth for communication, storage, and computing resources.**

***Index Terms*—Confidentiality, digital signatures, extended chaotic maps and probability security analysis systems, Internet of Things (IoT) complex systems.**

List of Notations

u	Private key.
v	Public key.
$\mathcal{T}$	Chebyshev chaotic maps.
r	Random number per message.
$\mathscr{h}_1, \mathscr{h}_2$	One Way Hash Functions.
$\mathcal{M}$	Message.
$\mathcal{B}$	First parameter of signature.
$\mathscr{s}$	Second parameter of signature.
σ	Digital signature.
$\mathscr{q}$	Large prime number of bit length.
$\mathscr{p}$	Large prime factors of $\mathscr{q} - 1$.

I. Introduction

Today is the era of Internet of Things (IoT) wherein different types of devices are connected to the Internet. These devices can be home appliances, agricultural equipment, manufacturing devices, industry tools, energy meter, mining sensors, healthcare monitoring instruments, environment equipment, surveillance systems, smart homes, smart cities, and smart grids among others, which comprise the machine-to-machine (M2M) model. With the advent of IoT-enabled devices, it is very easy to monitor or control various kinds of systems on the finger tips. IoT devices are smart enough to share and exchange data over public Internet to store on cloud. IoT is a powerful tool to apply on varieties of domains and proves the vital role by providing significant advantages. Ashton presented the notion of "*IoT*" and IoT devices came into existence in 2005. Since then tremendous evolution in IoTs has been reported; starting from the invention of basic smart devices to human centered sophisticated devices [1]. Thus, IoT devices received wide acceptance to use in various areas such as smart environment and human-centered design. The different methodologies have been adopted by the researchers to develop and experiment with IoT-enabled systems in a wide range of applications [2]. In addition, the architectures presented to investigate real-world problems are developed using the notion of IoT [3]. This motivates the research in IoTs to explore more possibilities in order to utilize the tremendous power of IoTs.

Manuscript received May 16, 2020; revised September 6, 2020; accepted November 16, 2020. *(Corresponding author: Chandrashekhar Meshram.)*

Chandrashekhar Meshram is with the Department of Post Graduate Studies and Research in Mathematics, Jaywanti Haksar Government Post-Graduation College, College of Chhindwara University, Betul 480001, India (e-mail: csmeshram84pdf@gmail.com, cs_meshram@rediffmail.com).

Mohammad S. Obaidat is with the College of Computing and Informatics, University of Sharjah, Sharjah 27272, UAE, with KAST, The University of Jordan, Amman 11942, Jordan, and also with the University of Science and Technology Bejing, Bejing 100083, China (e-mail: msobaidat@gmail.com, m.s.obaidat@ieee.org).

Jitendra V. Tembhurne is with the Computer Science and Engineering Department, Indian Institute of Information Technology, Nagpur 440006, India (e-mail: jitendra.tembhurne@cse.iiitn.ac.in).

Shailendra W. Shende is with the Information Technology Department, Yeshwantrao Chavan College of Engineering, Nagpur 441110, India (e-mail: shailendra.shende@gmail.com).

Kailash W. Kalare is with the Computer Science and Engineering Department, PDPM Indian Institute of Information Technology, Design, and Manufacturing, Jabalpur 482005, India (e-mail: kailashkalare.kk2612@gmail.com).

Sarita Gajbhiye Meshram is with the Department for Management of Science and Technology Development, Ton Duc Thang University, HoChi Minh City 758307, Vietnam, and also with the Faculty of Environment and Labour Safety, Ton Duc Thang University, Ho Chi Minh City 758307, Vietnam (e-mail: saritagmeshram@tdtu.edu.vn).

Digital Object Identifier 10.1109/JSYST.2020.3043358

The application of IoT has been witnessed in various domains ranging from industry automation to healthcare. Mostly, all the efforts are utilized to develop hardware platform, applications, better communication, and less emphasis has been put on user involvement and experience, and policies for security and privacy. This means, less emphasis has been given to human-related IoT.

Subsequently, we investigate human-centered IoT (HCIoT) enabled devices to offer more preference to human viewpoint in technology. HCIoT is an upcoming field of research that connects to various aspects of life including smart cards, e-commerce, business, healthcare, and sensitive private data. For example, the smart systems still need the involvement of human to start the system initially. Moreover, observations obtained from the smart systems need to provide information as per the suitability of human rather than smart devices. With the availability of various smart devices, information exchange got new dimension in IoT and communication networks. The study shows the impact of socio-technical framework for the assessment and prediction of the growth of IoT in South Korea [4]. In addition, the complexity between technical and social aspects involving HCIoT, and various opportunities and challenges are highlighted. Nevertheless, the involvement of human–computer interface (HCI) in IoT opens an avenue for the design of HCI-IoT [5]. This will be very useful in HCIoT wherein more concern is given to human-centered design. Hence, IoT is not only M2M paradigm, but humans are also involved as an important base, which is called humanized IoT, i.e., H-IoT [6]. Consequently, semantic technologies are utilized to add intelligence to IoT systems, but it is still more device-centric rather than human-centric. An Open Things [7] platform is proposed to facilitate applications of semantic technologies to build human-centered systems in the context of spatial and temporal aspect.

Nevertheless, there are many opportunities and challenges in the design of HCIoT [8]. Thus, we need not only to focus on performance, communication, integration, and interoperability of the IoT system, but also and even more on features of user's application, need of a user and motivation in human-centered IoT.

In HCIoT, information is exchanged between the devices over the public communication channel. Hence, deceptive activities come in existence to steal or change the information. Due to this, maintaining secrecy and privacy during the time of transmission is the major challenge. Thus, we need more efficient and robust security mechanism for the exchange of information. IoT devices are resource-constrained and loose the resources on heavy computation. So, the regular digital signature algorithms (DSA) are not suitable to apply. For IoT, we need fast and lightweight short-size signature security scheme.

In [9], new security threats and challenges are highlighted based on new IoT features, i.e., new IoT devices will need new security mechanism. How the latest advancement in IoT features and IoT security have an effect on existing security methods? For the security of HCIoT, we need the lightweight security method to adjust with capability of devices. A lightweight and secure scheme for smart grid based on key agreement utilizing elliptic curve cryptography (ECC) is proposed in [10]. The article claimed that the scope of attack of the attacker is negligible. In HCIoT, the authentication is the first step to enter into the system, so we have to prevent from unauthorized user to access the system. Vijayakumar *et al.* [11] presented authentication scheme to handle user anonymity for IoT-enabled wireless body area networks (WBAN) to monitor patient's data and send data to distant doctors, maintaining the anonymity of both patient and doctor.

Recently, RSA and BLS (Boneh–Lynn–Shacham) are used to verify the data integrity for Cloud-IoT using short signature, as illustrated in [12]–[14]. The issue with this scheme is the overheads of RSA computation and low efficiency of hash function for batch signature. In [15], investigation is performed on Wang *et al.*'s [16] scheme, and improved proxy resignature using identity is proposed, which overcomes the universal forgery attack. The novel certificate-based proxy signature method for Industrial IoT (IIoT) is proposed in [17]. This scheme has utilized the random oracle model (ROM) to achieve security. A new group signature scheme using lattice is suggested by Luo and Jiang [18]. The signature of the person in a group can be verified on lattice. The security of this scheme is proved on the basis of anonymity and traceability using ROM.

A short-signature scheme using the chaotic map is more efficient and takes less computational cost. Thus, we adopt chaotic maps in our proposal of short-signature scheme for security in HCIoT. More efficient authentication schemes were proposed by Meshram *et al.* [19], [20] using extended chaotic maps, results obtained in these schemes witness the suitability of chaotic maps as a good choice for the proposed new security scheme.

Protection of data integrity using bilinear pairing for the ID-based combined signature method for wireless sensor network (WSN) is presented in [21]. The combined signature obtained from cluster specifically WSN is verified by a designated verifier. The prominent features of ID-based cryptography and aggregate signature are employed by the authors to provide data integrity with optimized bandwidth utilization. The proposed method achieves partial aggregation of signatures, and then needs pairings linearly. Moreover, to verify aggregation, it also requires more pairings. Progressively, bilinear pairing for group signature approach is also applied in IIoT [22] to secure event notification (in publish/subscribe platform), i.e., messages exchanged between IoT nodes. By using group signatures mechanism, management of certificates, violating node decoupling, and extreme resource utilization can be avoided.

To do so, ID-based cryptosystem and bilinear pairings are adopted. The problem with this approach has to do with key revocation. Nevertheless, it takes time to test the bilinear maps used by the pairing-based short-signature schemes. Cui *et al.* [37] introduced new server-aided attribute-based signature with revocation, which not only securely mitigates user workloads in verifying and generating signatures, but also allows user revocation by making the server instantly avoid signature generations for revoked signers. However, these techniques are based on bilinear pairing. Therefore, the storage efficiency of the techniques in [22] and [37] comes at the cost of sacrificing efficiency in computation. In addition, these short-signature schemes are not as computationally efficient as the DSA-type signing schemes.

Therefore, the storage efficiency of pairing-based signatures comes at the cost of sacrificing efficiency in computation.

This article presents a lightweight provably secure digital short-signature technique (DSST) using extended chaotic maps for smart devices in HCIoT. It utilizes the less comprehensive operations based on extended chaotic maps to produce the security credentials during verification and signing operations. This technique's key benefit over the DSA signature scheme is reduction in the computation process for verification as well as the signature length by one-fourth. The technique is demonstrated with exemplary simple step-by-step values to display proof of notion. This reduces computation and communication overheads, and coordination along with increased flexibility compared to current comprehensive operations based on the real number in DSA-based schemes. In addition, we show that the security of our proposed technique is closely linked, if not strongly, to the complexity of solving extended chaotic maps. Under adaptive selected attacks in ROM, we present an effective proof of security for unforgeability exists, i.e., the presented technique offers superior security guarantees than current DL-based signature methods. The presented technique does not utilize pairings, which result in effortlessness implementation and higher efficiency; it neither depends on the fairly untested nor recent hardness assumptions associated with pairing-based cryptography. Results show that our technique is less time consuming for verification and signature processing when compared to competing schemes. Moreover, less time for verification of differences in message length, less communication cost required for messages with signatures, less bytes exposed by undermining devices, and less possibility of negotiating midway devices.

This article is structured as follows. Section II defines the description and terminology related to the present scheme. The proposed new techniques are listed in Section III. Section IV explains the security target of signature techniques, security models, and provably security in ROM, and we present a reductionist proof of security against forgery exists under the adaptive selected message attacks (EUF-CMA) in ROM. Section V compares our proposed scheme with other competing techniques. Finally, Section VI concludes the article.

II. Related Materials

There has been considerable interest in researching the behavior of chaotic processes over the past decade. Responsive reliance on initial conditions, resemblance to random behavior, and continuous broadband power spectrum define them. Chaos has potential applications in several functional blocks of a digital communication system: encryption; compression; and modulation. The probability for self-synchronization of chaotic oscillation has generated an avalanche of works on the application of chaos in cryptography. Chebyshev polynomial plays a crucial role in the study of chaos. Chebyshev polynomial maps have chaotic properties, which are suitable for cryptographic purposes. In addition to the semigroup property, the pseudorandomness of these polynomials is an attractive feature for cryptographically purposes such as enhancing the security and efficiency. In this section, we review Chebyshev chaotic maps and Chebyshev polynomial maps, which will be used in the proposed technique. We will then define some necessary notations used in the article the Nomenclature section.

A. Chebyshev Chaotic Maps

Here, we elaborate on the functionality of Chebyshev polynomials [23]. A polynomial $\mathcal{T}_k(z)$ is a Chebyshev polynomial with a degree k in the variant z. Let us have an exponent z and $z \in [-1,\ 1]$, and an integer n. The polynomial Chebyshev is defined as follows:

$$\mathcal{T}_k(z) = \cos(k \times \arccos(z)), \mathcal{T}_0(z) = 1, \mathcal{T}_1(z) = x, ..,$$
$$\mathcal{T}_k(z) = 2z\mathcal{T}_{k-1}(z) - \mathcal{T}_{k-2}(z); k \geq 2.$$

Here, $\cos(z)$ and $\arccos(z)$ are trigonometric [24] functions defined as $\mathrm{arcos}: [-1,\ 1] \to [0,\ \pi]$ and $\cos: \mathcal{R} \to [-1,\ 1]$. Two essential properties of Chebyshev polynomials [25], [26] are adopted, i.e., chaotic and semigroup property.

1) Chaotic property: The Chebyshev polynomial map, characterized as $\mathcal{T}_k : [-1,\ 1] \to [-1,\ 1]$ using degree $k > 1$, is a chaotic map with its exponent density function being $f^*(x) = \frac{1}{(\pi\sqrt{1-z^2})}$ with positive Lyapunov exponent $\lambda = \mathrm{In}\, k > 0$.
2) Semigroup property:

$$\mathcal{T}_\ell(\mathcal{T}_w(z)) = \cos\left(\ell \cos^{-1}\left(\cos\left(w \cos^{-1}(z)\right)\right)\right)$$
$$= \cos\left(\ell w \cos^{-1}(z)\right) = \mathcal{T}_{w\ell}(z) = \mathcal{T}_w(\mathcal{T}_\ell(z))$$

where w and ℓ serve as a positive integers with $z \in [-1,\ 1]$.

Chebyshev polynomials have two problems, and are difficult to handle in polynomial time.

1) The DL's task is to identify an integer w where objective is $\mathcal{T}_w(z) = y$ for known two components z and y.
2) Diffie–Hellman problem's (DHP) task is the estimation of exponent $\mathcal{T}_{\ell w}(z)$ for given three components z, $\mathcal{T}_w(z)$, and $\mathcal{T}_\ell(z)$.

B. Extended Chaotic Maps

In 2008, Zhang [27] showed that the semigroup property satisfies Chebyshev polynomials in the interval $(-\infty,\ +\infty)$. The enhancement can be done by the following equation:

$$\mathcal{T}_k(z) = (2z\mathcal{T}_{k-1}(z) - \mathcal{T}_{k-2}(z)) \pmod{q}$$

where $k \geq 2$, $z \in (-\infty,\ +\infty)$, and prime number q. Now, we find the recurrence relations, $\mathcal{T}_k(z) = 12\mathcal{T}_{k-1}(z) - \mathcal{T}_{k-2}(z))\ (\mathrm{mod}\ 13)$ with $\mathcal{T}_1(z) = 6$ and $\mathcal{T}_0(z) = 1$, where $q = 13$. Then, the values of $\mathcal{T}_k(z)$ are 1, 6, 6, 1, 6, 6, …, which are generated from aforementioned recurrence. The period chosen is $\mathcal{T} = 3$ [20], [28]. Obviously,

$$\mathcal{T}_\ell(\mathcal{T}_w(z)) \equiv \mathcal{T}_{w\ell}(z) \equiv \mathcal{T}_w(\mathcal{T}_\ell(z)) \pmod{q}.$$

Still holds semigroup property, and enhanced Chebyshev polynomials still can transform under composition. The notations used here are listed in Nomenclature section for DSST.

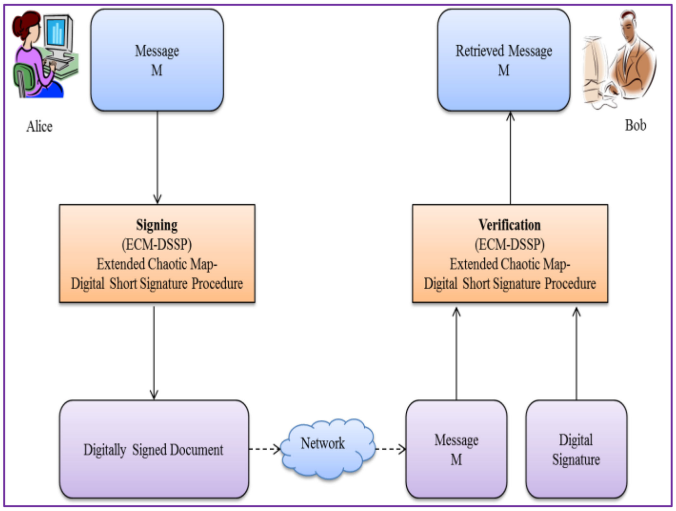

Fig. 1. Proposed digital short signature and verification using extended chaotic maps.

III. Proposed Digital Short-Signature Technique

A. Digital Short-Signature Technique (DSST) for HCIoT

In this section, we propose DSST, which uses a novel method for verification and signature operations based on extended chaotic maps. Compared to previous DL, integer factorization, and paring-based digital signature systems, our technique achieves better protection for smaller bit sizes.

Using ECM to build DSST makes it more stable than preliminary ones. With $p|(q-1)$, let q and p be large prime numbers. Also, let $\mathrm{G}_{y,q} = \{y^0, y^1, \ldots, y^{p-1}\}$ be a subgroup of z_q^*, the multiplicative group with order q, where y is a generator. We have adopted one way secure hash functions like h_1 and h_2, where $h_1 : \{0,1\}^* \times z_q^* \rightarrow \{0,1\}^{\frac{m_p}{2}}$ and $h_2 : \{0,1\}^* \rightarrow z_q^*$. We will ignore the "(mod p)″ and "(mod q)″ indicators for convenience in notation. We denote the p *by* $|p| = m_p$ bit length, and q *by* $|q| = m_{\mathrm{p}}$ bit length. The notation $b \overset{\mathrm{R}}{\leftarrow} \mathcal{S}$ implies that it is a randomly selected from a set $\mathcal{S}$ and maintaining uniformity.

Digital signature consists of a series of bits determined by certain laws and criteria utilized to define to check the validity of message and signatory. A known parameters, which is specific to user community, i.e., public integer $(q, \mathcal{B})$ is used in DSST. The value of q must be high enough to avoid simple calculation of the extended chaotic maps. The message is signed by the signer's (like Alice) private key $u \overset{\mathrm{R}}{\leftarrow} z_p^*$ and signatures validation is done by analogous public key $v \leftarrow \mathcal{T}_u(y)$. The random number r is utilized to generate the signature for every message. The biggest popular devisor of $(u,\ q-1) = 1$ and the signer must select u secretly. The required signature is $\sigma = (\mathcal{B},\ s)$. Fig. 1 consists of the signature process at the signer and verification process at verifier likes Alice and Bob, respectively.

B. DSST—Signature and Verification Processes

The random number r is initially generated during the signature process, and then the hash function h_2 is applied over the message $\mathcal{M}$ to generate $\mathcal{A}$. Next, calculate k using $\mathcal{A}$, $\mathcal{T}$, and r in function F1. Applying hash function h_1 over $\mathcal{M}$and k produces digital signature $\mathcal{B}$, as a first component. Similarly, digital signature second part s is produced in F2 using $\mathcal{B}$, r, and u. To ensure its power, r is chosen absolutely random and very unique for every signature. It was measured for enhancing randomness by involving current hash and private key.

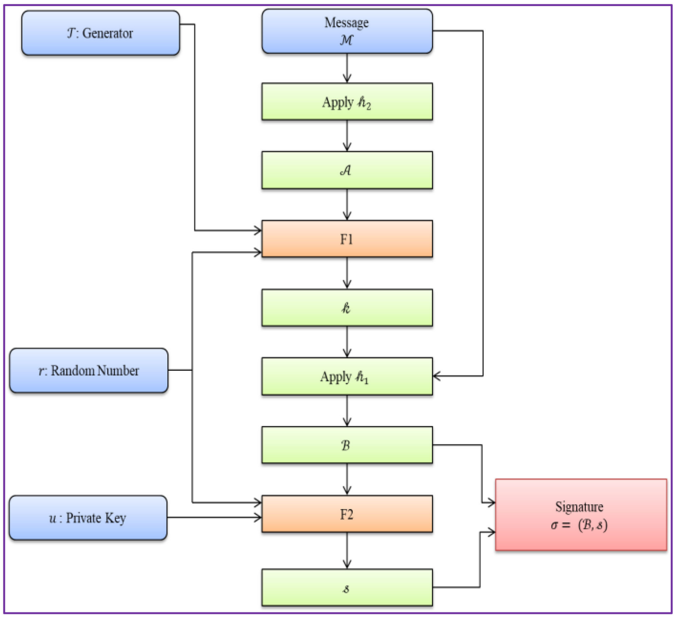

Fig. 2. Signature generation method for ECM-DSST based on F1 and F2.

In this section, we proposed *Alice* as a signer, and *Bob* as a receiver verifier. Initially, we choose prime numbers, which consist of 100 or 200 digits. Then, based on parameters, *Alice* selects u as a Private Key and generates $v \leftarrow \mathcal{T}_u(y)$ as a Public Key. Private key is confidential, and it is passed to *Bob* by public key. The $\mathrm{F1} = \mathcal{A}.\mathcal{T}_r(y)$ function and its SHA-1-based hash function $h_1(\mathcal{M}, k)$ is computed in pseudocode I to obtain $\mathcal{B}$ as per step 2. Likewise, according to step 3, $\mathrm{F2} = r - \mathcal{B}u$, which is equal to s. Now, *Alice* produces the digital signature $(\mathcal{B}, s)$, as illustrated in Fig. 2 and then forwards the public key to *Bob*.

During verification at receiver, *Bob* as verifier. Initially, compute $\mathcal{A} = h_2(\mathcal{M})$ and then sender's public key v, $\mathcal{A}$, generator $(\mathcal{T})$, and digital signature $(\mathcal{B}, s)$ are used for the computation of k' in function F3. Now, concatenation of $\mathcal{M}$ with k' is performed and it is added to Hash function h_1 to obtain $\mathcal{B}'$. Finally, comparison of $\mathcal{B}'$and $\mathcal{B}$ is performed, as shown in Fig. 3. If $\mathcal{B}'$and $\mathcal{B}$ are equal, then the message is original, which was sent by *Alice*, then any intermediate node or person may change the message. Furthermore, the proof of validity for checking the proposed scheme is provided in Proof I.

IV. Security Model and Proof of Security

First, we review the prototype of the security investigation for proposed DSSTs. Second, we see the concept of the random oracle and the "provable security." Last, we provide a near-reductionist technique to prove that our DSST is safe to prevent existential forgery (EUF) in the setup of chosen attacks

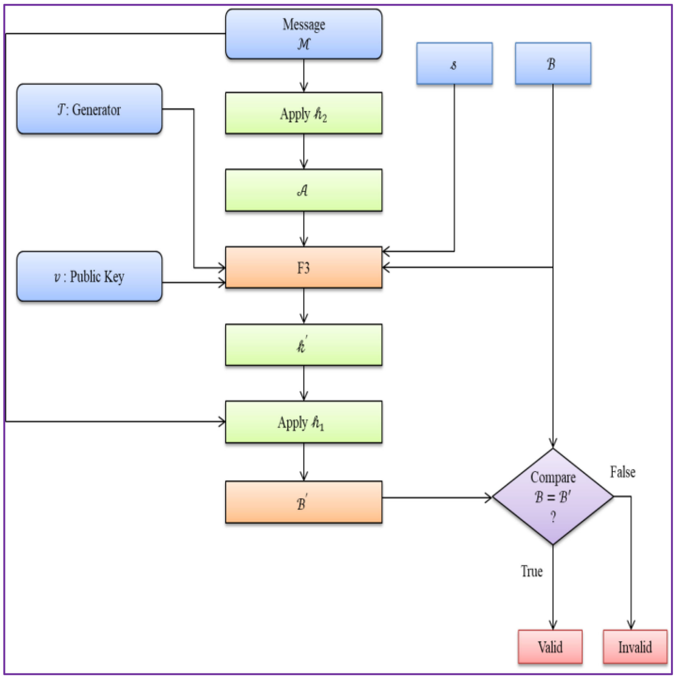

Fig. 3. Signature verification process for ECM-DSST using F3.

in ROM assuming extended chaotic maps are difficult under HCIoT environments.

A. Security Model of Signature Technique

Goldwasser *et al.* [30] provided the basic security description of the DSSTs, which is the first technique to satisfy. Three types of attacks are targeted: universal forgery, EUF, and total break. Nonetheless, the devices that the *foe* may use to undermine the signature's protection will also differ. First case, *foe* just knows the signer's public key. Second case, *foe* has access to a collection of correct pairs (message, signature). The adaptive chosen-message attack (CMA) is strong, where the enemy will allow the signer's sign for some of his/her chosen message based on previously received question responses. If the algorithm for signature generation is not deterministic, many signatures can exist that correspond to a single given message. Single-occurrence adaptive chosen-message attack (SO-CMA) [34], which compromises security, permits the *foe* to ask for each message at most one signature.

Definition 2 (DSST's Existential CMA Security): We conclude that $\mathrm{A}(\mathrm{t}, \mathscr{p}_{\hbar_1}, \mathscr{p}_{\mathrm{sig}}, \epsilon)$, probabilistic algorithm—breaks a digital short-signature procedure (SSP) when running for t steps, and creating adaptive queries of $\mathscr{p}_{\hbar_1}$ to the hash function oracles. Then demanding signatures $(\mathscr{p}_{\mathrm{sig}})$ for adaptively chosen messages, $(\mathcal{M}, \sigma)$ forged signature is produced by A for some message M with ϵ probability, where probability is based on coins of A, *Gen* algorithm, *Sig* algorithm, and hash function oracles. The DSST is $(\mathrm{t}, \mathscr{p}_{\hbar_1}, \mathscr{p}_{\mathrm{sig}}, \epsilon)$-secure if it $(\mathrm{t}, \mathscr{p}_{\hbar_1}, \mathscr{p}_{\mathrm{sig}}, \epsilon)$ cannot be broken by any forger.

Definition 3 (ECM Assumption): Probabilistic algorithm $(\mathfrak{S})$ is said to $(t, \mathcal{E})$—breaks ECM in a group $G_{\mathscr{y},\mathscr{q}}$, if $\mathfrak{S}$ runs in a maximum of t steps and calculates the extended chaotic maps $\mathrm{ECM}_{\mathscr{y},\mathscr{q}}\left(\mathcal{T}_{\mathscr{b}}(\mathscr{y})\right) = \mathscr{b}$ given input $(\mathscr{y}, \mathscr{q}, \mathscr{p})$ and $\mathcal{T}_{\mathscr{b}}(\mathscr{y})$ with $\mathcal{E}$ probability, where probability is based on uniformly selected coins of $\mathfrak{S}$ and $\mathscr{b}$ from $z_{\mathscr{q}}^{*}$. We say that group $G_{\mathscr{y},\mathscr{q}}$ is a $(t, \mathcal{E})$—ECM group if no algorithm in group $G_{\mathscr{y},\mathscr{q}}$ can split ECM.

B. Provable Security and ROM

The security assurance to users may be provided by the strong mathematics adapted by extended chaotic maps; factoring is hard to solve. Following the suggestions of [31] and [32] proposed a "ROM" to provide validated proven security for the cryptosystems. The hash function is utilized as an oracle to produce a random number for each and every new query. A reductionist technique of a logical assumption-contradicting method is used by *foe*. Probabilities are considered on coins tosses as well as random oracles. In reality, hash function, which is well-constructed, often does not produce random responses. Consequently, the value of the proofs performed in ROM is controversial. In [33], the authors developed "artificial" counterpart that is "provably secure" in ROM. Nevertheless, at least a random-model security proof will make a good argument for the SSP to secure.

C. Security Proof of the Presented DSST Using ECM

The proposed DSST is a general digital signature technique considered in [36]. Given an input message $\mathcal{M}$, we can produce $(\sigma_1, \mathcal{B}, \sigma_2)$ in which σ_1 randomly choses its value in a set that consists of larger values. $\mathcal{B}$ is a hash value $(\hbar_1(\mathcal{M}, \sigma_1), \hbar_2(\mathcal{M}))$ and σ_2 depends only on σ_1, $\mathcal{M}$, and $\mathcal{B}$. The following generic result can be obtained through explicit use of techniques in [36].

Theorem 1 (Forking lemma): Let $\mathcal{U}$ be a Turing machine with probabilistic polynomial time, the input of which contains public information only. By R and $\mathcal{O}$, we denote count of relevant queries $\mathcal{U}$ may request from the random oracle, and count of relevant queries A may request from the signer, respectively. Suppose in time limit T, $\mathcal{U}$ produces a valid $(\mathcal{M}, \sigma_1, \mathcal{B}, \sigma_2)$ signature with a probability of $\mathcal{E} \geq \frac{10(\mathcal{O}+1)(\mathcal{O}+\mathcal{R})}{2^k}$. If the triple $(\sigma_1, \mathcal{B}, \sigma_2)$ can be simulated with an indistinguishable probability of distribution without knowing the secret key, then there is another system that has control over the machine, which can be obtained from $\mathcal{U}$ by replacing the interaction with the signer with a simulation and which produces two valid signatures $(\mathcal{M}, \sigma_1, \mathcal{B}, \sigma_2)$ and $(\mathcal{M}, \sigma_1, \mathcal{B}', \sigma_2')$ such that $(\hbar_1(\mathcal{M}, \sigma_1), \hbar_2(\mathcal{M})) \neq (\hbar_1'(\mathcal{M}, \sigma_1), \hbar_2(\mathcal{M}))$ in the predicted time $T' \leq 120686T/\mathcal{E}$.

In the notation of our presented technique, we find two equations: $\hbar_2(\mathcal{M})\mathcal{T}_{\mathscr{s}}(\mathscr{y})\mathcal{T}_{\hbar_1(\mathcal{M},\mathscr{k})}(v) = \mathscr{k}$ and $\hbar_2(\mathcal{M})\mathcal{T}_{\mathscr{s}'}(\mathscr{y})\mathcal{T}_{\hbar_1'(\mathcal{M},\mathscr{k})}(v) = \mathscr{k}$.

From these, the secret value can be determined

$$u = (\mathscr{s} - \mathscr{s}')\ /\hbar_1'(\mathcal{M}, \mathscr{k}) - \hbar_1(\mathcal{M}, \mathscr{k})).$$

Though, the reductionist technique of Forking lemma is not effective, due to the fact that DSST's security and ECM's hardness relation gets compromised. Naturally, when simulator responds to the query $\hbar_2(\mathcal{M})$ by $\mathscr{k}\mathcal{T}_{\mathrm{t}}(\mathscr{y})$, as a replacement for a random number t in $\mathscr{z}^*_{\mathscr{q}}$ (note: simulator responds to the $\mathscr{h}_2$-query $\mathscr{h}_2(\mathcal{M})$ by random number t in $\mathscr{z}^*_{\mathscr{q}}$ as per proof of Theorem 1), then the secret value $u = (-\mathscr{s} - \mathrm{t})/\mathscr{h}_1(\mathcal{M}, \mathscr{k})$ would be obtained. Therefore, oracle replay attack is not needed. Thus, we want to propose a more powerful reductionist technique in depth. The following theorem provides an intimate relationship between the DSST security and the ECM problem's hardness.

Theorem 2: Let $\mathrm{G}_{\mathscr{y},\mathscr{q}}$ be a (t', ϵ')- ECM group, then the DSST in the ROM is $(\mathrm{t}, \mathcal{Q}_{\mathscr{h}_1}, \mathcal{Q}_{\mathscr{h}_2}, \mathcal{Q}_{\mathrm{sig}}, \epsilon)$secure against EUF-CMA, where $\mathrm{t}' \approx 3(\frac{\mathrm{t}}{2} + \mathcal{C}_e(\mathcal{Q}_{\mathscr{s}} + \frac{\mathcal{Q}_{\mathscr{h}_2}}{2}))$ and $\epsilon' \approx (\frac{\epsilon}{2} - \frac{1}{2^{(m_{\mathscr{p}}/2+1)}} - \frac{\mathcal{Q}_{\mathscr{s}}(\mathcal{Q}_{\mathscr{s}}+\mathcal{Q}_{\mathscr{h}_1})}{2^{(m_{\mathscr{q}}+1)}}) + (\epsilon - \frac{1}{2^{m_{\mathscr{p}}/2}} - \frac{\mathcal{Q}_{\mathscr{s}}(\mathcal{Q}_{\mathscr{s}}+\mathcal{Q}_{\mathscr{h}_1})}{2^{m_{\mathscr{q}}}})^3$ $(\frac{1}{16} + \frac{1}{8\mathcal{Q}_{\mathscr{h}_1}})$. Here, $\mathcal{C}_e$ refers to the cost of computing a long exponentiation in $\mathrm{G}_{\mathscr{y},\mathscr{q}}$ group.

Proof: We utilize the ROM to prove security of DSST. Suppose we get a EUF-CMA forger $\mathcal{U}$ that is $(\mathrm{t}, \mathcal{Q}_{\mathscr{h}_1}, \mathcal{Q}_{\mathscr{h}_2}, \mathcal{Q}_{\mathrm{sig}}, \epsilon)-$ splits the DSST. Here, $\mathcal{U}$ is a probabilistic polynomial time program that is provided with long public series of random bits, and can query a polynomial number of questions to the random oracles $\mathscr{h}_1, \mathscr{h}_2$, S.

As a "simulator," we want to create an algorithm $\mathfrak{S}$, which takes $(\mathscr{q}, \mathscr{p}, \mathscr{y}, v)$ as input. $\mathfrak{S}$ attempts to use $\mathcal{U}$ to compute the ECM, i.e., $\mathcal{T}_v(\mathscr{y})$ as a computer program. Algorithm $\mathfrak{S}$ simulates one or two DSST runs to forger A. Algorithm $\mathfrak{S}$ responds to hash queries $\mathscr{h}_1$ and $\mathscr{h}_2$ by A, S signature queries, and attempts to turn A's potential forgeries $(\mathcal{M}, \sigma)$ into an ECM, i.e., $\mathcal{T}_v(\mathscr{y})$ solution. Algorithm $\mathfrak{S}$ begins the first simulation by supplying $(\mathscr{q}, \mathscr{p}, \mathscr{y}, v)$ and a long series of random bits for A. Then, $\mathfrak{S}$ responds as follows to A's queries.

Responding $\mathscr{h}_1$-Oracle Queries: If A exposes a random oracle query $(\mathcal{M}_{\mathrm{i}}, \mathscr{k}_{\mathrm{i}})$ in which $1 \leq \mathrm{i} \leq \mathcal{Q}_{\mathscr{h}_1}$, $\mathfrak{S}$ lookup the $\mathscr{h}_1$-list (query-response list) in which entries contain of tuples $((\mathcal{M}_{\mathrm{i}}, \mathscr{k}_{\mathrm{i}})\, \mathcal{B}_{\mathrm{i}})$ to acquire the conforming answer. If tuple $((\mathcal{M}_{\mathrm{i}}, \mathscr{k}_{\mathrm{i}}), \mathcal{B}_{\mathrm{i}})$ is in the $\mathscr{h}_1$-list, $\mathfrak{S}$ replies with $\mathcal{B}_{\mathrm{i}}$, then $\mathfrak{S}$ uniformly at random generates $\mathcal{B}_{\mathrm{i}}$from $\mathscr{z}^*_{\mathscr{p}}$, responds with it, and enhances tuple $((\mathcal{M}_{\mathrm{i}}, \mathscr{k}_{\mathrm{i}}), \mathcal{B}_{\mathrm{i}})$ to the $\mathscr{h}_1$-list.

Responding $\mathscr{h}_2$-Oracle Queries: If $\mathcal{U}$ subjects a random oracle query $(\mathcal{M}_{\mathrm{i}})$, where $1 \leq i \leq \mathcal{Q}_{\mathscr{h}_2}$, $\mathfrak{S}$ lookup the $\mathscr{h}_2$-list (list of query–reply) where entries contain of tuples $((\mathcal{M}_{\mathrm{i}}), \mathcal{A}_{\mathrm{i}}, \mathrm{t}_{\mathrm{i}})$ to acquire the conforming response. If the $\mathscr{h}_2$-list includes a tuple $((\mathcal{M}_{\mathrm{i}}), \mathcal{A}_{\mathrm{i}}, \mathrm{t}_{\mathrm{i}})$, then $\mathfrak{S}$ responds with $\mathcal{A}_{\mathrm{i}}$. If $(\mathcal{M}_{\mathrm{i}})$ is a fresh query, $\mathfrak{S}$ will lookup the $\mathscr{h}_1$-list. If the $\mathscr{h}_1$-list contains some tuples $((\mathcal{M}_{\mathrm{i}}, \mathscr{k}_{\mathrm{i}})\, \mathcal{B}_{\mathrm{i}})$, $\mathfrak{S}$ picks up one $\mathscr{k}_{\mathrm{i}}$, generates t_{i} from $\mathscr{z}^*_{\mathscr{p}}$ uniformly at random, calculates $\mathcal{A}_{\mathrm{i}} = \mathscr{k}_{\mathrm{i}}\, \mathcal{T}_{\mathrm{t}_{\mathrm{i}}}(\mathscr{y})$ and responds with $\mathcal{A}_{\mathrm{i}}$. $\mathfrak{S}$ attaches $\mathscr{h}_2$-list with the tuple $((\mathcal{M}_{\mathrm{i}}), \mathcal{A}_{\mathrm{i}}, \mathrm{t}_{\mathrm{i}})$. If no tuple $((\mathcal{M}_{\mathrm{i}}, \mathscr{k}_{\mathrm{i}})\, \mathcal{B}_{\mathrm{i}})$ exists in the $\mathscr{h}_1$-list, $\mathfrak{S}$ uniformly generates t_{i} from $\mathscr{z}^*_{\mathscr{p}}$ at random, calculates $\mathcal{A}_{\mathrm{i}} = \mathcal{T}_{\mathrm{t}_{\mathrm{i}}}(\mathscr{y})$, and responds with $\mathcal{A}_{\mathrm{i}}$. $\mathfrak{S}$ attaches the $\mathscr{h}_2$-list with the tuple $((\mathcal{M}_{\mathrm{i}}), \mathcal{A}_{\mathrm{i}}, 0)$.

Responding S-Oracle Queries: If $\mathcal{U}$ exposes a query for signature $(\mathcal{M}_{\mathrm{i}})$ in which $1 \leq \mathrm{i} \leq \mathcal{Q}_{\mathscr{s}}$, $\mathfrak{S}$ looks up the S-list (list of query–response) in which entries contain $(\mathcal{M}_{\mathrm{i}}, \mathcal{B}_{\mathrm{i}}, \mathscr{s}_{\mathrm{i}})$ to acquire the correct response. If there is a tuple $(\mathcal{M}_{\mathrm{i}}, \mathcal{B}_{\mathrm{i}}, \mathscr{s}_{\mathrm{i}})$ occurs in S-list then $\mathfrak{S}$ responds with $(\mathcal{B}_{\mathrm{i}}, \mathscr{s}_{\mathrm{i}})$. In the circumstance that $(\mathcal{M}_{\mathrm{i}})$ is a fresh query for signature, $\mathfrak{S}$ looks up the $\mathscr{h}_2$-list for the first time. If the $\mathscr{h}_2$-list includes a tuple $((\mathcal{M}_{\mathrm{i}}), \mathcal{A}_{\mathrm{i}}, \mathrm{t}_{\mathrm{i}})$, $\mathfrak{S}$ picks $\mathcal{A}_{\mathrm{i}}$. Otherwise, $\mathfrak{S}$ uniformly produces t_{i} at random from $\mathscr{z}^*_{\mathscr{p}}$, computes $\mathcal{A}_{\mathrm{i}} = \mathcal{T}_{\mathrm{t}_{\mathrm{i}}}(\mathscr{y})$, and adds the tuple $((\mathcal{M}_{\mathrm{i}}), \mathcal{A}_{\mathrm{i}}, 0)$ to the $\mathscr{h}_2$-list. Then, $\mathfrak{S}$ uniformly selects $\mathscr{s}_{\mathrm{i}}$, $\mathcal{B}'_{\mathrm{i}}$ from $\mathscr{z}^*_{\mathscr{p}}$ at random and calculates $\mathscr{k}_{\mathrm{i}} = \mathcal{A}_{\mathrm{i}}\, \mathcal{T}_{\mathcal{B}'_{\mathrm{i}}}(v)\mathcal{T}_{\mathscr{s}_{\mathrm{i}}}(\mathscr{y})$. $\mathfrak{S}$ replies with $(\mathcal{M}_{\mathrm{i}}, \mathcal{B}'_{\mathrm{i}}, \mathscr{s}_{\mathrm{i}})$, improves the tuple $(\mathcal{M}_{\mathrm{i}}, \mathcal{B}'_{\mathrm{i}}, \mathscr{s}_{\mathrm{i}})$, to S-list, and improves the tuple $(\mathcal{M}_{\mathrm{i}}, \mathcal{B}'_{\mathrm{i}}, \mathscr{k}_{\mathrm{i}})$ to $\mathscr{h}_1$-list. If tuple $((\mathcal{M}_{\mathrm{i}}, \mathscr{k}_{\mathrm{i}})\, \mathcal{B}_{\mathrm{i}})$ is in the $\mathscr{h}_1$-list with $\mathcal{B}_{\mathrm{i}} \neq \mathcal{B}'_{\mathrm{i}}$, the simulation will be aborted and restarted (this unfortunate occurrence is at most probability, $\frac{(\mathcal{Q}_{\mathscr{h}_1}+\mathcal{Q}_{\mathscr{h}_2})}{2^{m_{\mathscr{p}}/2}})$.

Clearly, the simulation is done using oracles that produce the outputs which are totally different than real attacks. By supposition, forger $\mathcal{U}$ proceeds a new legal message and signature tuple $(\mathcal{M}, \mathcal{B}, \mathscr{s})$ with probability ϵ. If $\mathcal{U}$ has not inquired $\mathscr{h}_2\ (\mathcal{M})$ or $\mathscr{h}_1(\mathcal{M}, \mathscr{k})$, the probability is $Pr\{\mathscr{h}_1\ (\mathcal{M}, \mathscr{h}_2\ (\mathcal{M})\mathcal{T}_{\mathscr{s}}(\mathscr{y})\mathcal{T}_{\mathcal{B}}(v)) = \mathcal{B}\} \leq \frac{1}{2^{m_{\mathscr{p}}/2}}$, since both $\mathscr{h}_2\ (\mathcal{M})$ and $\mathscr{h}_1(\mathcal{M}, \mathscr{k})$ are chosen randomly. Therefore, with the probability $(\epsilon - \frac{1}{2^{m_{\mathscr{p}}/2}} - \frac{\mathcal{Q}_{\mathscr{s}}(\mathcal{Q}_{\mathscr{s}}+\mathcal{Q}_{\mathscr{h}_1})}{2^{m_{\mathscr{q}}}})$, the forger $\mathcal{U}$ proceeds a fresh signature $(\mathcal{M}, \mathcal{B}, \mathscr{s})$ such that $\mathscr{h}_1\ (\mathcal{M}, \mathscr{h}_2\ (\mathcal{M})\mathcal{T}_{\mathscr{s}}(\mathscr{y})\mathcal{T}_{\mathcal{B}}(v)) = \mathcal{B}$ and $\mathscr{h}_2\ (\mathcal{M}) \in \mathscr{h}_2 -$ list, $\mathscr{h}_1(\mathcal{M}, \mathscr{k}) \in \mathscr{h}_1 -$ list.The $\mathscr{h}_2$-list contains two types of entries. If $\mathscr{h}_2\ (\mathcal{M}) = \mathscr{k}\mathcal{T}_{\mathrm{t}}(\mathscr{y})$, then $\mathscr{h}_2\ (\mathcal{M})\mathcal{T}_{\mathscr{s}}(\mathscr{y})\mathcal{T}_{\mathcal{B}}\ (v) = \mathscr{k}$ implies $\mathcal{T}_1(\mathcal{T}_{\mathscr{s}}(\mathscr{y}))\mathcal{T}_{\mathcal{B}}\ (v) = 1$, and $u = (-\mathrm{t} - \mathscr{s})/\mathcal{B}$. It is assumed that the number of $\mathscr{h}_1$-query $(\mathcal{M}, \mathscr{k})$ with $\mathscr{h}_2\ (\mathcal{M}) = \mathscr{k}\mathcal{T}_{\mathrm{t}}(\mathscr{y})$ is $\gamma_{\mathscr{p}_{\mathscr{h}_1}}$. Thus, in the first simulation, the probability of solving the ECM is γ. Suppose, $\mathfrak{S}$ gets the signature and message pair $(\mathcal{M}_{\mathrm{j}}, \mathcal{B}_{\mathrm{j},}\mathscr{s}_{\mathrm{j}})$ in the first simulation, with $\mathcal{B}_{\mathrm{j}} = \mathscr{h}_1\ (\mathcal{M}_{\mathrm{j}}, \mathscr{k}_{\mathrm{j}})$ and $\mathscr{h}_2(\mathcal{M}_{\mathrm{j}}) \neq \mathscr{k}_{\mathrm{j}}\mathcal{T}_{\mathrm{t}_{\mathrm{j}}}(\mathscr{y})$. By providing the same $(\mathscr{q}, \mathscr{p}, \mathscr{y}, v)$, Algorithm $\mathfrak{S}$ will start the second simulation with the probability $(1 - \gamma)$. $\mathfrak{S}$ gives the forger $\mathcal{U}$ the same random bits sequence, similar random responses to hash function and signature queries as those in the first simulation before $\mathcal{U}$ requests for $\mathscr{h}_1(\mathcal{M}_{\mathrm{j}}, \mathscr{k}_{\mathrm{j}})$. At this argument, $\mathfrak{S}$ provides various series of random bits, signatures, and different values for random functions. The only difference is that if the $\mathscr{h}_2$-query $(\mathcal{M}_{\mathrm{j}})$ is requested after this argument, $\mathfrak{S}$ responds with the same value which is at the time of first simulation. Here, we utilize "Forking lemma" in [36]. We expect that it will yield signature $(\mathcal{M}_{\mathrm{j}}, \mathcal{B}_{\mathrm{j},}\mathscr{s}_{\mathrm{j}})$ this time around such that $\mathscr{h}_2(\mathcal{M}_{\mathrm{i}}) \neq \mathscr{k}_{\mathrm{i}}\mathcal{T}_{\mathrm{t}_{\mathrm{i}}}(\mathscr{y})$ and $\mathscr{h}_2(\mathcal{M}_{\mathrm{i}})\mathcal{T}_{\mathscr{s}_{\mathrm{i}}}(\mathscr{y})\mathcal{T}_{\mathcal{B}_{\mathrm{i}}}(v)$ or the signature $(\mathcal{M}_{\mathrm{j}}, \mathcal{B}'_{\mathrm{j}}, \mathscr{s}'_{\mathrm{j}})$ with $\mathcal{B}'_{\mathrm{j}} \neq \mathcal{B}_{\mathrm{j}}$.

Thus, we utilize the "Splitting lemma" [29] to measure the probability at which $\mathcal{U}$ would work as expected. Let U be the set of probable random bits series and random function estimates that carry forger $\mathcal{U}$ up to the argument where $\mathcal{U}$ requests for $\mathscr{h}_1(\mathcal{M}_{\mathrm{j}}, \mathscr{k}_{\mathrm{j}})$. Let V be the set of probable random bit series and random function estimates after that. By inference, the probability at which $\mathcal{U}$ supplying the series of random bits and random estimates $(u||v)$, produces a forgery is ϵ for any ubiquity $u \in$ U, $v \in$ V. Using "Splitting lemma," a "agreeable" subset occurs $\Omega \in$ U such that

1) $Pr\{u \in \Omega\} \geq \epsilon/2$;
2) When $\mathscr{b} \in \Omega$; $v \in V$, the probability that $\mathcal{U}$ delivered the arbitrary bits and arbitrary values sequences $(\mathscr{b}||v)$, generates a forgery at least $\mathcal{E}/2$.

Expect the sequences of arbitrary bit and arbitrary function values given up to the argument in first simulation is $\mathscr{b}$. Accordingly, the probability that A provided $(\mathscr{b}||\mathrm{v})$ generates a forgery in second simulation in the circumstance of any $v \in \mathrm{V}$ is $(\epsilon/2)^2$. Forged signature probability $(\mathscr{M}_{\mathrm{i}}, \mathscr{B}_{\mathrm{i}}, \mathscr{s}_{\mathrm{i}})$ with $\mathscr{h}_2(\mathscr{M}_{\mathrm{i}}) = \mathscr{k}_{\mathrm{i}}\, \mathscr{T}_{\mathrm{t_i}}(\mathscr{y})$ and $\mathscr{h}_2(\mathscr{M}_{\mathrm{i}})\mathscr{T}_{\mathscr{s}_{\mathrm{i}}}(\mathscr{y})\mathscr{T}_{\mathscr{B}_{\mathrm{i}}}(\mathrm{v}) = \mathscr{k}_{\mathrm{i}}$ is δ. The forged signature probability $(\mathscr{M}_{\mathrm{j}}, \mathscr{B}'_{\mathrm{j}}, \mathscr{s}'_{\mathrm{j}})$ with $\mathscr{B}'_{\mathrm{j}} \neq \mathscr{B}_{\mathrm{j}}$ is $(1-\gamma)/((1-\gamma)\ \mathcal{Q}_{\mathscr{h}_1} = 1/\mathcal{Q}_{\mathscr{h}_1}$. The probability of $\mathfrak{S}$ solving the ECM in the second simulation is, thus, $(\epsilon - \frac{1}{2^{(m_{\mathscr{p}}/2)}} - \frac{\mathcal{Q}_{\mathscr{s}}(\mathcal{Q}_{\mathscr{s}}+\mathcal{Q}_{\mathscr{h}_1})}{2^{(m_{\mathscr{q}})}}) \quad (\frac{\epsilon}{2} - \frac{1}{2^{(m_{\mathscr{p}}/2+1)}} - \frac{\mathcal{Q}_{\mathscr{s}}(\mathcal{Q}_{\mathscr{s}}+\mathcal{Q}_{\mathscr{h}_1})}{2^{(m_{\mathscr{q}}+1)}})^2$ $(\gamma + \frac{1}{\mathcal{Q}_{\mathscr{h}_1}}) = (\epsilon - \frac{1}{2^{m_{\mathscr{p}}/2}} - \frac{\mathcal{Q}_{\mathscr{s}}(\mathcal{Q}_{\mathscr{s}}+\mathcal{Q}_{\mathscr{h}_1})}{2^{m_{\mathscr{q}}}})^3\ (\frac{\gamma}{4} + \frac{1}{4\mathcal{Q}_{\mathscr{h}_1}})$.

To sum up the probabilities, we observe that Algorithm $\mathfrak{S}$ at least resolves the ECM with probability as follows:

$$\gamma\left(\epsilon - \frac{1}{2^{(m_{\mathscr{p}}/2)}} - \frac{\mathcal{Q}_{\mathscr{s}}\left(\mathcal{Q}_{\mathscr{s}} + \mathcal{Q}_{\mathscr{h}_1}\right)}{2^{(m_{\mathscr{q}})}}\right) + (1-\gamma)\left(\epsilon - \frac{1}{2^{m_{\mathscr{p}}/2}} - \frac{\mathcal{Q}_{\mathscr{s}}\left(\mathcal{Q}_{\mathscr{s}} + \mathcal{Q}_{\mathscr{h}_1}\right)}{2^{m_{\mathscr{q}}}}\right)^3 \times \left(\frac{\gamma}{4} + \frac{1}{4\mathcal{Q}_{\mathscr{h}_1}}\right).$$

In one simulation, the computation phase is $(\mathrm{t} + (2\mathcal{Q}_{\mathscr{s}} + \mathcal{Q}_{\mathscr{h}_2})\mathcal{C}_e)$. Then, the final phase in the calculation is $\gamma(\mathrm{t} + (2\mathcal{Q}_{\mathscr{s}} + \mathcal{Q}_{\mathscr{h}_2})\mathcal{C}_e) + (1-\gamma)2\ (\mathrm{t} + (2\mathcal{Q}_{\mathscr{s}} + \mathcal{Q}_{\mathscr{h}_2})\mathcal{C}_e) = (2-\delta)\ (\mathrm{t} + (2\mathcal{Q}_{\mathscr{s}} + \mathcal{Q}_{\mathscr{h}_2})\mathcal{C}_e)$. The approximation of the probability γ, is comparable to the marginally weaker SO-CMA security framework, only one $\mathscr{h}_1$-query and one $\mathscr{h}_2$-query are permitted for each $\mathscr{M}$ request, i.e., the forger $\mathcal{U}$ requests $(\mathscr{M}, \mathscr{k})$ for both $\mathscr{h}_1$-query and one $\mathscr{h}_2$-query. Algorithm $\mathfrak{S}$ responds simultaneously with $\mathscr{h}_1\ (\mathscr{M}, \mathscr{k}) = \mathscr{B}$, and $\mathscr{h}_2\ (\mathscr{M}) = \mathscr{k}\, \mathscr{T}_{\mathrm{t}}(\mathscr{y})$. If this is the case, $\gamma = 1$. We are getting a strongly reductionist proof of protection.

Conflicting to this positive estimate, each $\mathscr{h}_1$-query $(\mathscr{M}, \mathscr{k})$ is subsequently the $\mathscr{h}_2$-query $(\mathscr{M})$. Therefore, as with the Schnorr signature scheme, we find a movable reductionist security proof like that when $\gamma = 0$. Since the set of series of arbitrary bits and arbitrary function values that $\mathfrak{S}$ provides to forger $\mathcal{U}$ is arbitrary, we are letting $\gamma = 1/2$. Therefore, $\mathrm{t}' \approx 3(\frac{\mathrm{t}}{2} + \mathcal{C}_e(\mathcal{Q}_{\mathscr{s}} + \frac{\mathcal{Q}_{\mathscr{h}_2}}{2}))$ and $\epsilon' \approx (\frac{\epsilon}{2} - \frac{1}{2^{(m_{\mathscr{p}}/2+1)}} - \frac{\mathcal{Q}_{\mathscr{s}}(\mathcal{Q}_{\mathscr{s}}+\mathcal{Q}_{\mathscr{h}_1})}{2^{(m_{\mathscr{q}}+1)}}) + (\epsilon - \frac{1}{2^{m_{\mathscr{p}}/2}} - \frac{\mathcal{Q}_{\mathscr{s}}(\mathcal{Q}_{\mathscr{s}}+\mathcal{Q}_{\mathscr{h}_1})}{2^{m_{\mathscr{q}}}})^3(\frac{1}{16} + \frac{1}{8\mathcal{Q}_{\mathscr{h}_1}})$. Notice that this reductionist technique's efficiency depends on $\mathscr{h}_1$-query command and $\mathscr{h}_2$-query for the identical message that forger A is requesting. Therefore, we make assumption that this reductionist proof is complete; lying among tight and loose (Goh and Tarecki [35]).

The security of the hash functions: We let $\mathscr{p}$ be 160 bits in order to get a short signature. It is easy to recover $\mathscr{M}$ and $\mathscr{M}'$ messages such that $\mathscr{h}_1\ (\mathscr{M}, \mathscr{k}) = \mathscr{h}_1\ (\mathscr{M}', \mathscr{k})$; by birthday attacks subsequently the hash value of $\mathscr{h}_1$ is 80 bits. If the invader demands a signature on $\mathscr{M}$, then the signature returned by the signer is based on a random number $\mathscr{k}'$ instead of $\mathscr{k}$. Although, we are not assured that finding other $\mathscr{M}'$ message with $\mathscr{h}_1\ (\mathscr{M}, \mathscr{k}) = \mathscr{h}_1\ (\mathscr{M}', \mathscr{k}')$ is feasible, we are sure that finding $\mathscr{M}'$ with $\mathscr{h}_2\ (\mathscr{M}') = \mathscr{h}_2\ (\mathscr{M})$ is unfeasible, because the hash value of $\mathscr{h}_2$ is at least 1024 bits. In the meantime, no process will recover $\mathscr{s}$, $\mathscr{B}$ from the multivariate congruence $\mathscr{h}_1(\mathscr{M}, \mathscr{h}_2(\mathscr{M})\mathscr{T}_{\mathscr{s}}(\mathscr{y})\mathscr{T}_{\mathscr{B}}(v))$ or find $\mathscr{k}, \mathscr{s}$ from $\mathscr{h}_2(\mathscr{M})\mathscr{T}_{\mathscr{s}}(\mathscr{y})\mathscr{T}_{\mathscr{h}_1(\mathscr{M},\, \mathscr{k})}\ (v) = \mathscr{k}$. Since the ROM adopts that hash functions are perfect, the probability is as follows:

$$\mathrm{Pr}_{\mathscr{k}\in \mathrm{z}_{\mathscr{q}}}\ \left\{\mathscr{h}_1\ (\mathscr{M}, \mathscr{k}) = \mathscr{B} | \forall\, \mathscr{M} \in \{0,1\}^*\ \ \forall\ \mathscr{B} \in 2^{m_{\mathscr{p}}/2}\right\} = \frac{1}{2^{m_{\mathscr{p}}/2}}.$$

TABLE I
NOTATIONS USED FOR COMPARATIVE ESTIMATIONS

SN.	Notation	Meaning
1	$\mathbb{t}_{\mathrm{exp}}$	Execution time for a modular exponentiation in group
2	$\mathbb{t}_{\mathrm{chaotic}}$	Execution time for Chebyshev chaotic map operation
3	$\mathbb{t}_{\mathrm{mul}}$	Execution time for a modular multiplication
4	$\mathbb{t}_{\mathrm{hash}}$	Execution time for one way hash function
5	$\mathbb{t}_{\mathrm{pair}}$	Execution time for one bilinear pairing operation
6	$\mathbb{t}_{\mathrm{inv}}$	Execution time for one modular inverse operation

V. Performance Comparison

The performance of the proposed scheme has been evaluated based on the storage cost, communication cost, and the computational cost metrics. The performance has been compared based on the cost for the signing stage and verification stage. It has been noted that the signing stage and verification stage require more computational costs compared to the stage of installation and extraction. Therefore, the comparative study has been done based on the computational cost for the signing stage and verification stage. The state-of-the-art studies discussed in Cui *et al.* [37], Shen *et al.* [21], Esposito *et al.* [22], Mughal *et al.* [38], Verma *et al.* [17], and Seo [39] have been compared with the proposed DSST work on performance metrics. The relations between $\mathbb{t}_{\mathrm{exp}}, \mathbb{t}_{\mathrm{chaotic}}, \mathbb{t}_{\mathrm{mul}}, \mathbb{t}_{\mathrm{ecsm}}, \mathbb{t}_{\mathrm{sym}}$, and $\mathbb{t}_{\mathrm{pair}}$ with respect to $\mathbb{t}_{\mathrm{hash}}$ ($\mathbb{t}_{\mathrm{hash}} = 0.503$ ms) have been established in [20] and [40]. The proposed work has used the above-mentioned notations and their relations are described in Table I. The relationship and computational complexity order among the metrics are shown as $\mathbb{t}_{\mathrm{chaotic}} \approx \mathbb{t}_{\mathrm{hash}}, \mathbb{t}_{\mathrm{mul}} \approx 2.5\, \mathbb{t}_{\mathrm{hash}},\ \mathbb{t}_{\mathrm{inv}} \approx 7.5\mathbb{t}_{\mathrm{hash}}, \mathbb{t}_{\mathrm{exp}} \approx 600\ \ \mathbb{t}_{\mathrm{hash}},\ \ \mathbb{t}_{\mathrm{pair}} \approx 1550\mathbb{t}_{\mathrm{hash}}$ and $\mathbb{t}_{\mathrm{hash}} \approx \mathbb{t}_{\mathrm{chaotic}} < \mathbb{t}_{\mathrm{mul}} < \mathbb{t}_{\mathrm{inv}} < \mathbb{t}_{\mathrm{exp}} < \mathbb{t}_{\mathrm{pair}}$.

Fig. 4 shows the comparative analysis between the existing schemes and the proposed scheme based on the computational cost for the signing stage. The proposed scheme has been found to be effective as compared to existing competing schemes. It requires 2.7665 ms for the signing stage.

Fig. 5 shows the comparison on the computational cost for the verification stage. It is seen that the proposed technique is also efficient in the verification stage. Table II presents the quantitative analysis of the proposed technique and shows the comparison based on the total cost including the signing stage and the verification stage. It is seen from Table II that the total cost has been reduced to 7.2935 ms as compared to competing techniques discussed in [17], [21], [22], [37], [38], and [39]. Thus, the proposed technique is found to be efficient as compared to the other competing techniques reported in the literature. As

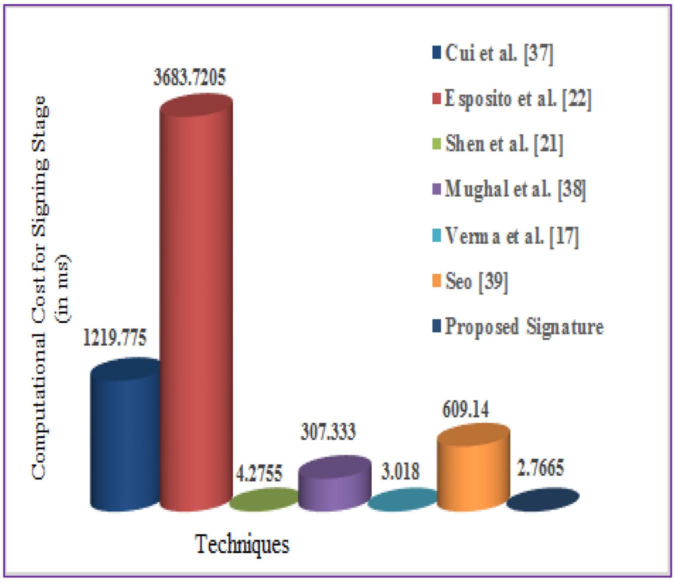

Fig. 4. Comparison based on computational cost for signing stage.

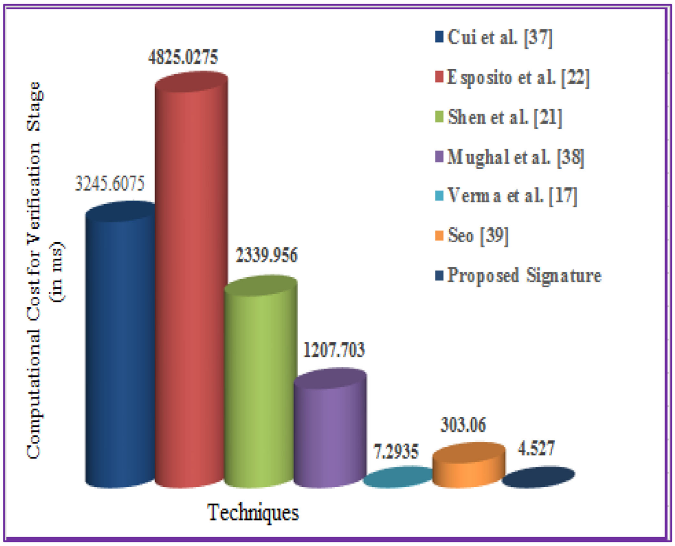

Fig. 5. Comparison based on computational cost for verification stage.

TABLE II
QUANTITATIVE ANALYSIS BASED ON TOTAL COMPUTATIONAL COST INCLUDING SINGING STAGE AND VERIFICATION STAGE

Techniques/stages	**Signing stage**	**Verification stage**	**Total (ms)**
Cui et al. [37]	$4\mathfrak{t}_{exp} + 10\mathfrak{t}_{mul}$	$3\mathfrak{t}_{pair} + 3\mathfrak{t}_{exp} + \mathfrak{t}_{mul}$	4465.38
Esposito et al. [22]	$\mathfrak{t}_{hash} + 7\mathfrak{t}_{exp} + 2\mathfrak{t}_{pair} + 3\mathfrak{t}_{mul} + 2\mathfrak{t}_{inv}$	$3\mathfrak{t}_{exp} + 5\mathfrak{t}_{pair} + 5\mathfrak{t}_{mul} + 4_{inv}$	8508.74
Shen et al. [21]	$\mathfrak{t}_{hash} + 3\mathfrak{t}_{mul}$	$2\mathfrak{t}_{hash} + 3\mathfrak{t}_{pair}$	2344.23
Mughal et al.[38]	$1\mathfrak{t}_{hash} + \mathfrak{t}_{exp} + \mathfrak{t}_{mul} + \mathfrak{t}_{inv}$	$\mathfrak{t}_{hash} + 4\mathfrak{t}_{exp}$	1515.03
Verma et al. [17]	$2\mathfrak{t}_{mul} + \mathfrak{t}_{hash}$	$5\mathfrak{t}_{mul} + 2\mathfrak{t}_{hash}$	10.31
Seo [39]	$1\mathfrak{t}_{hash} + 2\mathfrak{t}_{exp} + \mathfrak{t}_{mul} + \mathfrak{t}_{inv}$	$\mathfrak{t}_{exp} + \mathfrak{t}_{mul}$	912.19
Proposed Signature	$2\mathfrak{t}_{hash} + \mathfrak{t}_{chaotic} + \mathfrak{t}_{mul}$	$2\mathfrak{t}_{hash} + 2\mathfrak{t}_{chaotic} + 2\mathfrak{t}_{mul}$	7.29

compared to [17], [21], [22], [37], [38], and [39], the main advantage of proposed DSST over the DSA signature technique is that it has a one-fourth reduction in both the verification computation and signature length; the level of security is preserved. It utilizes less comprehensive operations based on extended chaotic maps to produce the security credentials during verification and signing operations. The presented technique does not utilize pairings and RSA, which result in effortless implementation and higher efficiency. Furthermore, it does not depend on the fairly untested or recent hardness assumptions associated with pairing-based cryptography and RSA-based cryptography, respectively. The new DSST are needed for low-bandwidth communication, low-storage and low-computation environments. It will be particularly very useful and applicable to smart cards and wireless devices.

VI. Conclusion

The protection of sensitive data is necessary in HCIoT to provide a security from forgery attacks. In asymmetric cryptography, digital signature is the secure choice for ensuring the ownership and validity of contact parties. This article presents a lightweight provably secure DSST using extended chaotic maps for secure communication in HCIoT. Under EUF-CMA in ROM, it is existentially unforgeable. Using this model (EUF-CMA in ROM) and proofs, we can easily investigate the technique's real security. In general, the security proofs obtained (EUF-CMA in ROM) are not enough in comparison to the SM. Results show the superiority of our technique as it takes less overheads based on the cost of computation and communication besides resilience analysis as compared with competing. The proposed DSST achieves less computation time and less overhead communication in verification and signature operations, besides improved resilience to capture attacks. Therefore, it is very difficult to crack ECM-based DSST compared with DSA that is DLP-based. In future, we plan to design the short-signature technique in public key in standard model environment.

Acknowledgment

The authors would like to thank anonymous reviewers of the IEEE System Journal for their careful and helpful comments.

REFERENCES

[1] J. Chin, V. Callaghan, and S. B. Allouch, "The Internet-of-Things: Reflections on the past, present and future from a user-centered and smart environment perspective," *J. Ambient Intell. Smart Environ.*, vol. 1, pp. 45–69, 2019.

[2] H. Wei, H. Luo, Y. Sun, and M. S. Obaidat, "Cache-aware computation offloading in IoT systems," *IEEE Syst. J.*, vol. 14, no.no. 1, pp. 61–72, Mar. 2020.

[3] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ.—Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, Jul. 2018.

[4] D. Shin, "A socio-technical framework for Internet-of-Things design: A human-centered design for the Internet of Things," *Telematics Informat.*, vol. 31, no. 4, pp. 519–531, 2014.

[5] T. L. Koreshoff, T. W. Leong, and T. Robertson, "Approaching a human-centered Internet of Things," in *Proc. 25th Australian Comput.-Human Interact. Conf.: Augmentation, Appl., Innov., Collab.*, Nov. 25–29, 2013, pp. 363–366.

[6] A. Pintus, D. Carboni, A. Serra, and A. Manchinu, "Humanizing the Internet of Things—Toward a human-centered Internet-and-web of things," in *Proc. 11th Int. Conf. Web Inf. Syst. Technol.*, 2015, pp. 498–503.

[7] M. A. Calderona, S. E. Delgadilloa, and J. A. Garcia-Macias, "A more human-centric Internet of Things with temporal and spatial context," in *Proc. 7th Int. Conf. Ambient Syst., Netw. Technol., Procedia Comput. Sci.*, 2016, pp. 553–559.

[8] A. Wafa, C. A. Zayaniy, I. Amousy, and F. S'edes, "User-centric IoT: Challenges and perspectives," in *Proc. 12th Int. Conf. Mobile Ubiquitous Comput., Syst., Survey Technol.*, 2019, pp. 27–34.

[9] W. Zhou, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1606–1616, Apr. 2018.

[10] F. Wu, L. Xu, X. Li, S. Kumari, M. Karuppiah, and M. S. Obaidat, "A lightweight and provably secure key agreement system for a smart grid with elliptic curve cryptography," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2830–2838, Sep. 2019.

[11] P. Vijayakumar, M. S. Obaidat, M. Azees, S. H. Islam, and N. Kumar, "Efficient and secure anonymous authentication with location privacy for IoT-Based WBANs," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2603–2611, Apr. 2020.

[12] D. Boneh and X. Boyen, "Short signatures without random oracles and the SDH assumption in bilinear groups," in *Proc. EUROCRYPT*, 2004, pp. 56–73.

[13] F. Guo, Y. Mu, and W. Susilo, "Short signatures with a tighter security reduction without random oracles," *Comput. J.*, vol. 54, no. 4, pp. 513–524, 2011.

[14] H. Zhu *et al.*, "A secure and efficient data integrity verification scheme for Cloud-IoT based on short signature," *IEEE Access*, vol. 7, pp. 90036–90044, 2019.

[15] J. Zhang, "Improvement of ID-based proxy re-signature scheme with pairing-free, wireless networks," *Wireless Netw.*, vol. 25, no. 7, pp. 4319–4329, 2019.

[16] Z. Wang, A. Xia, and M. He, "ID-based proxy re-signature without pairing," *Telecommun. Syst.*, vol. 69, pp. 217–222. 2018.

[17] G. K. Verma, B. B. Singh, N. Kumar, M. S. Obaidat, D. He, and H. Singh, "An efficient and provable certificate-based proxy signature scheme for IIoT environment," *Inf. Sci.*, vol. 518, pp. 142–156, May 2020.

[18] Q. Luo and C.-Y. Jiang, "A new constant-size group signature scheme from lattices," *IEEE Access*, vol. 8, pp. 10198–10207, 2020.

[19] C. Meshram, C. C. Lee, S. G. Meshram, and C.-T. Li, "An efficient ID-based cryptographic transformation model for extended chaotic-map-based cryptosystem," *Soft Comput.*, vol. 23, no. 16, pp. 6937–6946, 2019.

[20] C. Meshram, C.-T. Li, and S. G. Meshram, "An efficient online/offline ID-based short signature procedure using extended chaotic maps," *Soft Comput.*, vol. 23, no. 3, pp. 747–753, 2019.

[21] L. Shen, J. Ma, X. Liu, F. Wei, and M. Miao, "A secure and efficient ID-Based aggregate signature scheme for wireless sensor networks," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 546–554, Apr. 2017.

[22] C. Esposito, A. Castiglione, F. Palmieri, and A. D. Santis, "Integrity for an event notification within the industrial Internet of Things by using group signatures," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3669–3678, Aug. 2018.

[23] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. Boca Raton, FL, USA: Chapman & Hall, 2003.

[24] P. Bergamo, P. D'Arco, A. Santis, and L. Kocarev, "Security of public key cryptosystems based on Chebyshev polynomials," *IEEE Trans. Circuits Syst.-I, Reg. Papers*, vol. 52, no. 7, pp. 1382–1393, Jul. 2005.

[25] S. Han and E. Chang, "Chaotic map based key agreement with/out clock synchronization," *Choas Soliton Fractals*, vol. 39, no. 3, pp. 1283–1289, 2009.

[26] C. T. Li, C. L. Chen, and C. C. Lee, "A novel three-party password-based authenticated key exchange protocol with user anonymity based on chaotic maps," *Soft Comput.*, vol. 22, pp. 2495–2506, 2018.

[27] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos Solitons Fractals*, vol. 37, no. 3, pp. 669–674, 2008.

[28] F. Chen, X. Liao, K. W. Wong, Q. Han, and Y. Li, "Period distribution analysis of some linear maps," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, pp. 3848–3856, 2012.

[29] D. Pointcheval and J. Stern, "Security proofs for signature schemes," in *Proc. Eurocrypto*, 1996, pp. 387–398.

[30] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988.

[31] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proc. Crypto*, 1986, pp. 186–194.

[32] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocol," in *Proc. 1st ACM Conf. Comput. Commun. Secur.*, 1993, pp. 62–73.

[33] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology," in *Proc. 30st Annu. ACM Symp. Theory Comput.*, 1998, pp. 209–218.

[34] Z. Shao, "A provably secure short signature scheme based on discrete logarithms," *Inf. Sci.*, vol. 177, no. 23, pp. 5432–5440, 2007.

[35] E.-J. Goh and S. Jarecki, "A signature scheme as secure as the Diffie–Hellman problem," in *Proc. Eurocrypto*, 2003, pp. 401–415.

[36] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *J. Cryptol.*, vol. 13, no. 3, pp. 361–396, 2000.

[37] H. Cui, R. H. Deng, J. K. Liu, X. Yi, and Y. Li, "Server-Aided attribute-based signature with revocation for resource-constrained Industrial-Internet-of-Things devices," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3724–3732, Aug. 2018.

[38] M. A. Mughal, X. Luo, A. Ullah, S. Ullah, and Z. Mahmood, "A lightweight digital signature based security scheme for human-centered Internet of Things," *IEEE Access*, vol. 6, pp. 31630–31643, 2018.

[39] J. H. Seo, "Efficient digital signatures from RSA without random oracles," *Inf. Sci.*, vol. 512, pp. 471–480, 2020.

[40] A. M. Benasser and A. Samsudin, "A new identity based encryption (IBE) scheme using extended Chebyshev polynomial over finite fields ZP," *Phys. Lett. A*, vol. 374, no. 46, pp. 4670–4674, 2010.