# ECLSS: Extended Chaotic Map-Based Certificate less Signature Scheme

**[1*]Gajraj Singh, [2]Pankaj Kumar, [3]Garima Thakur, [4]Vinod Kumar, [5]Saurabh Rana**

[1]Discipline of Statistics, School of Sciences,
Indira Gandhi National Open University, Delhi-110068
[2] [3]Srinivasa Ramanujan Department of Mathematics,
Central University of Himachal Pradesh, Himachal Pradesh-176206, India
[4]Department of Mathematics, Shyam Lal College,
University of Delhi, New Delhi-110065, India
[5]Department of Mathematics, Bennet University,
Greater Noida, Gautam Buddh Nagar, Uttar Pradesh, India

E-mail: [1]gajraj76@gmail.com, [2] pkumar240183@gmail.com [3]garima48451@gmail.com,
[4]vkmaths@shyamlal.du.ac.in, [5]saurabhranapsm@gmail.com

**Abstract:** Chaos theory includes the study of the behavior of dynamical systems which are extremely delicate to complex constraints. Due to inherent traits of the chaotic system, such as susceptibility to the initial condition, ergodicity, and systemic parameterization, it has been extensively used in cryptography. A certificateless signature scheme is a very functional technique that covers the flaws of an identity-based signature scheme. This paper proposes a novel certificateless signature scheme (ECLSS) employing chaotic properties. We validate that the proposed scheme is unforgeable against adaptive chosen message attacks and prove the security by using the Random Oracle Model (ROM) under the computational hard Discrete Logarithm (C-DL) Problem that attains the required goal such as non-repudiation, confidentiality, and integrity. During the security analysis of the proposed scheme, we have shown that ECLSS satisfies many security attributes. Our ECLSS scheme is implemented with the widely accepted tool "AVISPA," and efficiency is demonstrated by fabricating comparisons in the form of tabular and graphical representations. In the performance analysis section, we show that the computational cost is effectively reduced by using extended chaotic property compared to the existing certificateless signature scheme.

**Keywords:** Certificate-less signature, Discrete Logarithm problem, Diffie-Hallman assumption, Security and Privacy, Random oracle model.

## 1. Introduction

Due to the meteoric evolution of the internet in the digital world, the security of networks like IoT, VANET, and Big Data has become more important than ever. Our life's dependency on digital services is increasing day by day. Data privacy and integrity have become important in our personal lives and social networks. We need more security services like authentication, integrity, and non-repudiation. Cryptography is a broad vision to achieve these required security services. The digital signature is one of the important cryptography techniques to provide security in the digital network. The digital signature includes security services integrity, non-repudiation, and confidentiality especially. We can categorize the Digital Signature techniques into two major groups: (a) chaos-based methods and (b) Non-chaos-based Methods. In the category of Non-chaos-based methods, more effort is required in computation. The researchers used many digital signature techniques like pairing and non-pairing based on an elliptic curve, RSA-based, and lattice-based. These techniques had many flaws, like efficiency and security. Chaos-based methods are used in encryption, image encryption, digital signature, and authentication. Chaos Theory is a branch of mathematics exploring non-linear dynamical systems with the butterfly effect. The butterfly effect of chaos theory shows a small amount of modification in the system can produce immense differences in the system. A chaotic system can be used in different scenarios like the population growth and decay models, climate change, electric circuits, and road traffic. Nowadays, the Butterfly effect of a chaotic system gives more attention to cryptography. Chaos-based cryptography is an emergent field of cryptography with the benefit of chaotic property from the last few years, as depicted in Fig 1.

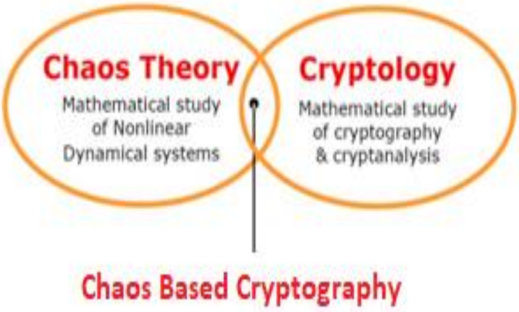

Chaos Theory
Mathematical study of Nonlinear Dynamical systems
Cryptology
Mathematical study of cryptography & cryptanalysis
Chaos Based Cryptography

**Fig 1:** Relation between Chaos Theory and Cryptography

Chaotic properties gain more attention in cryptography due to their precious properties that help to make more analyses of the behavior of the dynamic system. The main properties of chaos-based according to Devaneys definition, are 1) Sensitivity to initial Conditions 2) Deterministic 3) Topologically mixing orbits [1,2]. Sensitivity to the initial condition indicates the large scale of change by the small change in the system. Topology mixing orbit property implies the chaotic trajectory. The deterministic property means a dynamical network that can be considered an extension of the sensitivity characteristics. Most networks, such as VANET, IoT, healthcare, etc., are dynamic. In mathematical-rigorous terms, the sensitivity of networks is based upon their initial state, and the same is selected as the key in the dynamic network in chaos-based cryptography. The chaotic sensitivity property leads to extremely different outputs when iterations depend on the key. By this method, it observed that the exact output is extremely correlated with the key, and changing a single bit makes all the bits incorrect because of the diverging trajectories of randomly closed keys. The sensitivity measures the change of effects in the dynamic system by the small change, so sensitivity is very useful in encryption and signature due to its sensitive nature. As shown in fig 2, the behavior of the curve normally increases as r increases till the value 3.4 but the behavior of the curve changes very sharply when the value of r crosses the value 3.4. Chaotic cryptography combines cryptography and chaos theory, using chaotic properties to generate the private key and other attributes.

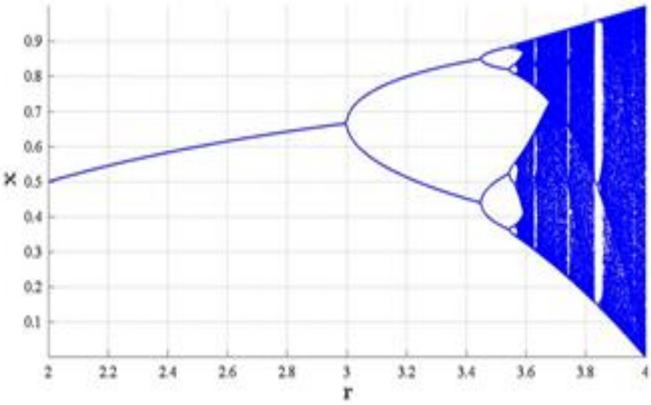


**Fig 2:** Sensitiveness in Chaos based cryptography [7]

### 1.1 Related work

A digital signature scheme is a cryptographic technique for securing information. Many digital signature methods have been introduced with different timelines such as symmetric key, asymmetric key, identity-based and certificateless infrastructure. Shamir proposed the ID-based public key infrastructure (ID-PKI) structure in 1984 [5]. The third trusted party (KGC) generates the user's private key in ID-PKI, leading to the keynote problem associated with the ID-PKI say key escrow problem (KEP). Al-riyami and Peterson [6] proposed a solution for KEP in the form of a certificateless signature scheme (CLS) and restricted the KGC’s power. In the CLS scheme, a third party generates the user’s private key partially instead of the original private key. Based on Al-riyami and Peterson's pioneering work, many CLS schemes were designed by the researchers [8-17] based on the pairing operation with the hard computational Diffie-Hellman (C-DH) Problem. Some CLSs are constructed without pairing operation [18, 24, 27] based on the hard problem discrete logarithm problem. In construction He et al. [24] claimed that pairing operation consumes efforts 20 times more than scalar multiplication operations. The study of chaotic frameworks provides a better alternative to cryptographic primitives. Chaotic applications have become widely considered in recent years and have been placed in the mainstream of cryptography. Over the last few years, Chaos-based cryptography has been focused worldwide because of its properties. Now chaotic map is frequently used to generate encryption techniques [19-22], hash functions [21-22], and S boxes [23]. Recently, several authentication techniques and key agreement protocols have been proposed by using the chaotic map [25-26, 28-29]. In digital signatures, many contributors have constructed their own schemes. Chain and Kuo [30] established new signature schemes with extended chaotic maps and recognized the closed relationship between chaotic systems and cryptosystems. Islam [32] designed encryption and digital scheme based on the extended chaotic map and proved the security with hard problem discrete logarithm (D-L) based on the chaotic map. Tahat [31] designed an identity-based cryptographic framework combining integer factoring problems and chaotic map. Meshram et al [33] proposed a fractional chaotic map-based short signature scheme for the human-centric Internet of Things framework and proved their security under the random oracle model. Meshram et al. [35] proposed an ID-based short signature scheme using extended chaotic maps and established that it is unforgeable against the chosen message attack and other attacks. Meshram [34] constructed a fractional chaotic map based on the online/offline environment. Most of the proposed chaotic-based signature is developed for the ID-based infrastructure and suffers from the key escrow problem that KGC knows the user’s private key. Our proposed ECLSS removes the flaws of the KEP generated in the previously designed ID-based chaotic map signature scheme that KGC generates the user’s partial private key instead of the user’s private key. The proposed ECLSS scheme has used the property of an extended chaotic map, effectively reducing the computation cost. The security of the proposed scheme is verified with widely accepted ROM. Thereafter, we implement our ECLSS scheme with the widely accepted tool "AVISPA" and demonstrate the efficiency of our ECLSS by comparison in the form of table and graphical representation. The comparison shows that our ECLSS scheme is more efficient than other environments such as pairing-based or without pairing-based CLS schemes and achieves security services.

### 1.2 Motivation and Contribution

After a comprehensive literature review of the existing CLS schemes, it was found that previous schemes [15, 17, 24, and 27] are based on hard problems i.e. elliptic curves, by using pairing and without pairing due to which it suffered from communication costs as well as high computational costs. Pairing operation has higher computation costs than other operations such as scalar multiplication or hash to map operation. Hence, the existing schemes are incompatible with small devices with limited computational power. This paper established a new infrastructure for CLS schemes with Chaotic Maps proven by the random oracle model (ECLSS). The proposed protocol has various significant benefits as mentioned below:

- The security framework is proved formally and informally in our proposed ECLSS construction.
- The proposed ECLSS enjoys the benefits of chaotic properties.
- The proposed ECLSS framework satisfies the desired security requirements and removes the KEP problem.
- The performance analysis shows that the proposed ECLSS framework has given a better performance of attributes.
- To our knowledge, the proposed ECLSS scheme is the first CLS scheme using an extended chaotic map with provable security under the environment of the random oracle model.

## 2. Mathematical preliminaries

We explain the Chebyshev extended chaotic maps as we utilize them in our proposed framework respectively,

**Table 1:** Notations

| Symbols | Description |
|---|---|
| KGC | Key generation center |
| $\alpha$ | KGC's master key |
| $U$ | KGC's public key |
| $ID$ | The user's identity |
| $k$ | User's pseudo key with identity $ID$. |
| $Params$ | The system parameters published by KGC |
| $t$ | user's private key with identity $ID$. |
| $B$ | user's public key with identity $ID$ |
| $m$ | The message is transferred by identity $ID$. |
| $\sigma$ | Generated signature by signer corresponding message $m$. |

### 2.1. Chebyshev chaotic maps

The most effective method to study the dynamical network in a chaotic system is the "Liapounov Exponent"[3,4].

Definition: Consider a one-dimensional system with the initial value $x_0$,

$$x_{k+1} = f(x_k), \quad f: I \to I, \text{where } x_0 \in I, f \text{ is a continuous function.}$$

$$\delta(x_0) = \lim_{T\to\infty} \frac{1}{T} \sum_{k=0}^{T-1} log|f'(x_k)|$$

$\delta(x_0)$ is called the Lyapunov exponent [4].

The Lyapunov exponent is independent of the initial value $x_o$ in the chaotic dynamical network. The Lyapunov exponent demonstrates the quantity of average in initial displacement is changed by repeated application of $f$. It is an important tool to measure the strength of variation in the initial conditions of a dynamic system.

Chebyshev polynomial $T_n(x)$ is a polynomial with the function of $x$ having degree $n$. Let $x \in \{-1,1\}$ be the sensitivity range and $n$ be an integer.

The clarification of the Chebyshev polynomial is given below.

$$T_n(x) = \cos\,(n \times arccos(x))$$

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \qquad n \geq 2$$

Where, $cos(x)$and $arccos(x)$are the trigonometric functions which are depicted as $arccos: [-1,1] \to [0, \pi]$ and $cos: R \to [-1,1]$.

:

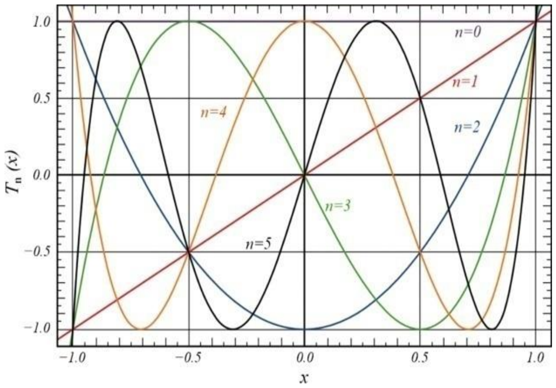


**Fig 3:** Chebyshev polynomial [38]

Chebyshev polynomial contains mainly two following properties [34].

1) The chaotic property: The Chebyshev polynomial described as $T_r[-1,1] \rightarrow [-1,1]$having degree $n > 1$ is a chaotic map corresponding to invariant function $f^*(x) = \frac{1}{x\sqrt{1-x^2}}$, for some positive Lyapunov exponent $\rho = \log n > 0$.
2) The semigroup property:

$$T_r\big(T_t(x)\big) = \cos\left(rcos^{-1}\big(\cos\big(tcos^{-1}(x)\big)\big) = \cos\left(rt(\cos^{-1}(x)\right) = T_{rt}(x) = T_{tr}(x)\right.$$
$$= T_t(T_r(x))$$

Where $x \in [-1,1]$, $t$, and $r$ are positive integers.

**Definition 1:** A function $\varphi(x)$ is said to be negligible if for a given $\varepsilon \geq 0$ there exists a number $x_0$ such that $\varphi(x) \leq 1/t^{\varepsilon}$ for every $x \geq x_0$.
**Definition 2:** Given two-component x and y it is computationally infeasible to find the integer $w$ from $T_w(x) = y$ (Discrete Logarithm Problem)
**Definition 3:** For a given random instance $x, T_w(x)$ and $T_t(x)$, it is computationally infeasible to find $T_{wt}(x) \in G_1$*(Computational Diffie-Hellman).*

**Extended chaotic map:** Han and Chang [38] proposed that the semigroup property discussed in the above section holds on $(-\infty, +\infty)$ for the Chebyshev polynomial. For the sake of convenience above semigroup property can be improved as following:

$$T_n(x) = \big(2xT_{n-1}(x) - T_{n-2}(x)\big)mod\ p \text{ , for } n \geq 2$$

Where $x\epsilon\ (-\infty, +\infty)$ and $p$ is a large number. Obviously,

$$T_r\big(T_t(x)\big) = T_{rt}(x) = T_{tr}(x) = T_t(T_r(x))$$

## 3. Formal model of a CLS scheme

A CLS scheme consists of the following algorithms:-
***Setup*****:** KGC executes this algorithm by taking security parameter $l\ as\ input$. KGC then produces public key, master secret key, and some public parameters.
***Pseudo-Gen*****:** KGC generates a pseudo key $v$ after taking identity $ID$ as input and transfers it to the user.
***UK-Gen*****:**The user creates its private and public keys with the help of PPK.
***Sign*****:** User/Signer generates a certificateless signature (CLS) say $\sigma$ corresponding tuple$(m, t, v)$.
***Verification*****:** After taking an input$(\sigma, m, ID)$, the verifier decides the validity of the signature $\sigma$.
**Security model**: Suppose an adversary ב exists in our system that queries the following oracles in the polynomial time-bound:
***CreateUser(.)oracle*****:** Adversary submits a query on identity $ID \in \{0,1\}^*$. Oracle searches whether the appropriate entry exists in the database and returns $B$. Otherwise, it performs *Reveal-Pseudo* and *Reveal-UK* queries to obtain pseudo key $v$, and private key $t$. Thereafter, updates the list $L = (ID, t, B, v)$ and stores in the database. Finally, $B$ *is* returned to the adversary.

***RevealPseudo(.)oracle*:** Adversary submits a query on identity $ID \in \{0,1\}^*$, Oracle searches whether the appropriate entry exists in the database. If yes, then return $v$ to the adversary. Otherwise, returns $\perp$.

***RevealUK(.)oracle*:** Adversary submits a query on identity$ID \in \{0,1\}^*$, Oracle searches whether the appropriate entry exists in the database. If yes, then return $t$ to the adversary. Otherwise, returns $\perp$.

***Sign(.)oracle***: After a sign query corresponding to the tuple $(m, ID, B)$is submitted, oracle returns a signature $\sigma$ to the adversary.

Forge: Adversary submits a tuple $(m, \sigma, ID, B)$and wins if any of the following conditions is fulfilled;

i) An authentic signature $\sigma$ is generated without substituting private/public key pair.

ii) If the target identity $ID$ is not created then, $\perp$ is returned.

We construct a Game for adversary ℶ in the CLS scheme respectively.

**Game:** $\beta$ is the simulator/ challenger that deals with adversary ℶ and executes the subsequent steps to respond.

1. ℶ submit a query for ***CreateUser(.)oracle***, $\beta$ produces the KGC's master key and system parameters and transfers the corresponding response to ℶ.
2. In this step, ℶ can generate queries for ***RevealPPK(.)oracle, RevealUK(.)oracle,*** and ***Sign(.) oracle*** in the polynomial time-bound manner at any stage.
3. ℶ yields a signature $\sigma^*$to a corresponding identity $ID^*$on a message $m^*$ with public key $B_{ID^*}$.
4. ℶ is in the winning situation of the game if any of the subsequent conditions holds.
5. The signature $\sigma^*$is lawful on $m^*$corresponding to $ID^*$.
6. The oracle has never performed ***RevealPPK(.)oracle*** or ***RevealPPK(.)oracle*** query for receiving the partial private or private keys related to identity $ID^*$.
7. The ***Sign(.)oracle*** is never functioned for $m^*$corresponding to identity $ID^*$.

## 4. The ECLSS framework

Symbols used in the proposed CLS scheme are shown in table 1. As mentioned above in section 2, our ECLSS structure comprises of five algorithms *Master-K-Gen*, *PPK-Gen*, *Private-K-Gen*, *Sign*, and *Verification*.

***Master-K-Gen:*** This algorithm is executed by $KGC$ after considering a global parameter $x \in Z_p^*$ as input and then selecting a huge prime number $p$.

1. $KGC$ selects a random number$\alpha \leftarrow Z_p^*$ and computes $U \leftarrow T_\alpha(x)(mod\ p)$.
2. Choose a cryptographic chaotic hash function$H_1, H_2$such that$H_1: \{0,1\}^\infty \rightarrow Z_p^*$and $H_2: \{0,1\}^\infty \rightarrow Z_p^*$.
3. Finally, generate the system parameters $\{p, x, H, U\}$ called $Params$ and retain the master key $\alpha$ secretly.

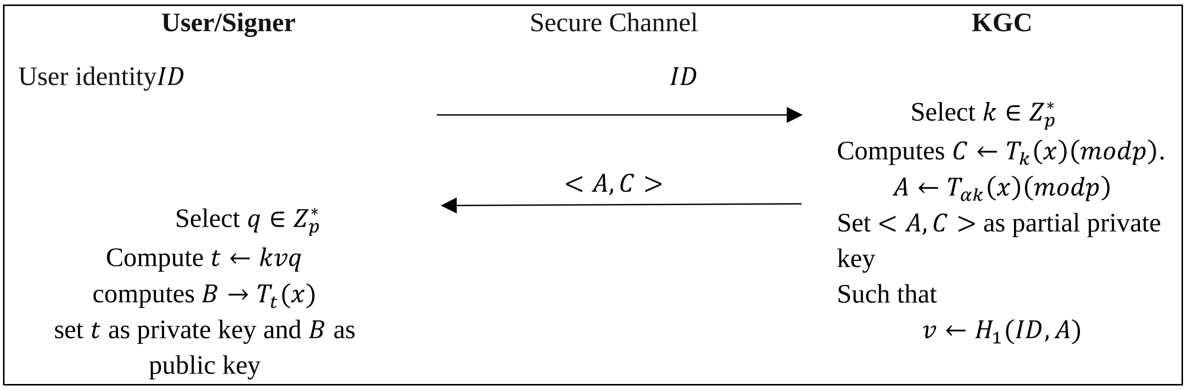
***PPK-Gen:*** This algorithm is executed by the KGC after taking a user id say $ID_i$ as an input.

1. KGC select a random number $k \in Z_p^*$.
2. KGC computes $C \leftarrow T_k(x)(modp)$.
3. KGC computes $A \leftarrow T_{\alpha k}(x)(modp)$.
4. KGC compute a pseudo key ($PPK$) say $< A, C >$ corresponding with user id $ID_i$such that $v \leftarrow H_1(ID_i, A)$.

Finally, KGC transfers $v$ $PPK$ to the user having identity $ID_i$ by a protected medium.

***Private- k-Gen***: This algorithm is performed by the user with identity $ID$.

1. User selects a random number $q \in Z_p^*$.
2. Computes $v \leftarrow H_1(ID_i, A)$.
3. User computes $t \leftarrow Avq$ and set $t$ as private key.
4. User computes $B \leftarrow T_t(x)$ and set as public key.



**User/Signer** | Secure Channel | **KGC**

User identity$ID$

$ID$ →

Select $k \in Z_p^*$

Computes $C \leftarrow T_k(x)(modp)$.

$A \leftarrow T_{\alpha k}(x)(modp)$

Set $< A, C >$ as partial private key

Such that

$v \leftarrow H_1(ID, A)$

← $< A, C >$

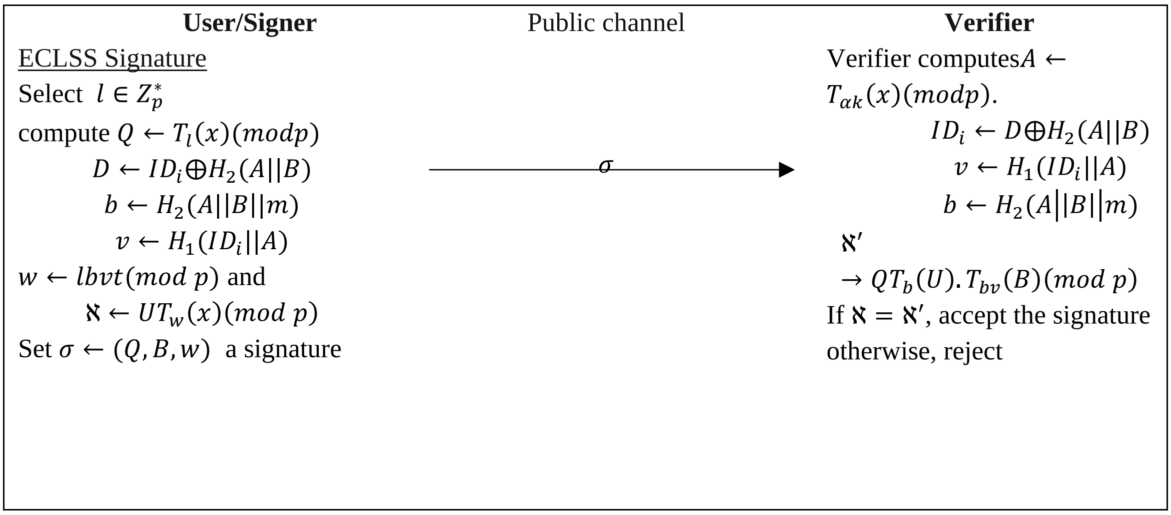
Select $q \in Z_p^*$

Compute $t \leftarrow kvq$

computes $B \rightarrow T_t(x)$

set $t$ as private key and $B$ as public key

**Fig 4:** Interaction between User and KGC



**User/Signer** | Public channel | **Verifier**

<u>ECLSS Signature</u>

Select $l \in Z_p^*$

compute $Q \leftarrow T_l(x)(modp)$

$D \leftarrow ID_i \oplus H_2(A||B)$

$b \leftarrow H_2(A||B||m)$

$v \leftarrow H_1(ID_i||A)$

$w \leftarrow lbvt(mod\ p)$ and

$\aleph \leftarrow UT_w(x)(mod\ p)$

Set $\sigma \leftarrow (Q, B, w)$ a signature

$\sigma$ →

Verifier computes$A \leftarrow T_{\alpha k}(x)(modp)$.

$ID_i \leftarrow D \oplus H_2(A||B)$

$v \leftarrow H_1(ID_i||A)$

$b \leftarrow H_2(A\big|\big|B\big|\big|m)$

$\aleph' \rightarrow QT_b(U).T_{bv}(B)(mod\ p)$

If $\aleph = \aleph'$, accept the signature otherwise, reject

**Fig 5:** Interaction between User and Verifier

***Sign:*** *A user/ authorized* signer with identity $ID_i$ takes $params$, a pseudo key $v$ and its secret key $t$, signs a message $m \in \{-\infty, +\infty\}$.

1. User select a random number $l \in Z_p^*$, then calculates $Q \leftarrow T_l(x)(modp)$.
2. $D \leftarrow ID_i \oplus H_2(A||B)$
3. $b \leftarrow H_2(A||B||m)$
4. $v \leftarrow H_1(ID_i||A)$
5. User computes $w \leftarrow lbvt(mod\ p)$and $\aleph \leftarrow UT_w(x)(mod\ p)$.

$\sigma \leftarrow (Q, B, w, C, D)$is a signature corresponding to message $m$.

Verification**:** A verifier takes a signature $\sigma \leftarrow (Q, B, w, C, D)$ corresponding to user identity $ID$ and the following algorithm can verify their public key B.

1. Verifier computes$A \leftarrow T_{\alpha k}(x)(modp)$.
2. $ID_i \leftarrow D \oplus H_2(A||B)$
3. $v \leftarrow H_1(ID_i||A)$
4. $b \leftarrow H_2(A\big|\big|B\big|\big|m)$
5. Verifies $\aleph' \rightarrow QT_b(U).T_{bv}(B)(mod\ p)$.

a. If $\aleph = \aleph'$ then the signature is verified and accepted otherwise, rejected.

**Correctness of the algorithm:**

To show the correctness of the algorithm, we compute $Q \leftarrow T_l(x)(modp)$ then,

$$QT_b(U).T_{bv}(B)(mod\ p) = T_l\ (x).T_b(T_\alpha(x).T_{bv}\big(T_t(x)\big)(mod\ p) = T_l(x)T_{b\alpha}(x)T_{bvt}(x)(mod\ p)$$
$$= T_\alpha(x)T_{blvt}(x)(mod\ p) = UT_w(x)(mod\ p)$$

## 5. Security analysis

The security analysis of ECLSS is discussed in the following manner:

### 5.1 Formal security analysis

Here, we investigate the security of our proposed ECLSS scheme under the Random Oracle Model (ROM) [38]. The security of the proposed ECLSS is proved against adaptive chosen message and identity attacks with the help of the following theorems.

**Theorem 5.1:** The constructed ECLSS procedure$(\in, t, q_H, q_E, q_s)$is unforgeable against the adaptive chosen message and identity attacks, assuming that the $(\in', t')$-extended chaotic hypothesis hold in $Z_p^*$ in random oracle, where $\in' = \left(1 - \frac{1}{p}\right)\left(\frac{1}{q_{H_2}} - \frac{(q_E+q_s)q_{H_1}}{p}\right) \in$, $T' = T + O(q_s + q_E)\tau$, $q_E$, $q_{H_1}, q_{H_2}, T$and $q_s$ are the number of extraction inquiries, chaotic hashing inquiries, time for exponentiation operation and signing inquiries.

**Proof**: Suppose that ∃ an adversary say ℶ in the system. We establish a simulator $\beta$ that helps the adversary to solve the extended chaotic map with discrete logarithm. Now we will build a procedure$\beta$ to attain our target.

Adversary ℶcollaborates with$\beta$ in this game.$\beta$ arbitrarily selects a user identity $ID_i$ and sends the security parameter $params = \{p, x, H, U\}$ to ℶ. The simulator $\beta$ finds$Z_p^*$ , with a global parameter and a variable$G \in Z_p^*$. $\beta$ maintains a list $L = (ID_i, t, B, v)$. Now $\beta$ is prepared to implement the oracle query.

***$H_1$* query:**When $H_1$ takes an identity $ID_i$, according to Bellare et al [34] proof $\beta$ selects a random number $\gamma \in Z_p^*$ such that $D = T_\gamma(x)(mod\ p)$ and then $\beta$ maintains a list $L_{H_1} = (ID_i, D, \gamma)$ to answer the adversary's queries.

***$H_2$*query:** $\beta$ maintains a list $L_{H_2}$with the tuple $(Q, B, m)$. In the response to query regarding$H_2$ oracle, $\beta$ searches in the list$L_{H_2}$. If list $L_{H_2}$holds the tuple $(Q, B, m)$ then $\beta$ returns $c$ toℶ. Otherwise $\beta$ selects $c \in Z_q^*$, sets $c \leftarrow H_2(Q, A, m)$ and inserts it in the list $L_{H_2}$.

***Reveal-pseudo-key-queries:*** Adversary ℶ submits a query for pseudo key corresponding identity$ID_i$, $\beta$ recalls the list $L_{H_1}$. If pseudo key corresponding identity $ID$ is not found then $\beta$ proceedsto failure and discontinues the simulation else$\beta$checks the list $L_{H_1}$ and selects a random number $d \in Z_q^*$ such that $d \leftarrow H_1(ID_i, A)$ and update the list $L_{H_1}$.

***Reveal-Secret-Key-queries*:** Adversary ℶ submits a query for a secret key corresponding to an identity$ID_i$, $\beta$ recalls the list $(ID_i, t, B, d)$and ℶ searches in the list. If $t \neq \perp$, then $\beta$ sends $t$ to ℶ. If $t = \perp$then $\beta$ selects $t \in Z_q^*$, puts $B = T_t(x)(mod\ p)$, and returns $e$ to ℶand then updates the list $(ID_i, t, B, d)$.

***Sign*:** When ℶ submits a request on $(ID_i, m), \beta$examines the list$L$, $L_{H_1}, L_{H_2}$ and performs the subsequent procedure. If the list contains$(ID_i, t, B, d)$, $l, \alpha, t, d$ then $\beta$ examines whether $t = \perp$or not. If $t \neq \perp$ then$\beta$ reverts$B$ to ℶ. If $t = \perp$, then $\beta$ submits *Reveal-Public-key* query to generate $B = T_t(x)(mod\ p)$ where $t \in Z_q^*$. If the list does not contain $(ID_i, t, B, d)$ then $\beta$ makes *Reveal-Public-key* query on $ID_i$and updates the list$(ID_i, t, B, d)$.

For generating the signature, The adversary ℶ yields a fake signature $\sigma_1^* = (Q^*, B^*, w_1^*)$ on identity $ID^*$and the message$m^*$. Then the system $\beta$ switches to the adversary ℶ with the argument where it enquires $h(Q^*, B^*, w_1^*)$ and conveys with additional value. Adversary ℶ yield two more extra signature $\sigma_2^* = (Q^*, B^*, w_2^*)$ and $\sigma_3^* = (Q^*, B^*, w_3^*)$. Noted that $B^*$ and $Q^*$ should be the same unavoidably. Let $n_1, n_2, n_3$be the results of the queries $h(Q^*, B^*, m^*)$ made three times.

By$l, \alpha, t, d \in Z_q^*$, now we use the extended chaotic maps for computing$B, Q$ and $D$separately i.e. $E = T_\alpha(x)(mod\ p)$, $Q = T_l(x)(mod\ p)$ and $B = T_t(x)(mod\ p)$ then we have

$$w_j^* = E^{-1}(l * \alpha * n_j * t * n_j * d)(mod\ p)\text{for}j = 1,2,3$$

Only l,α, t,d are known to β in these mathematical declarations. The system $\beta$ resolves the above three linear numerical equations for $j = 1,2,3$ and yields $t$as the resolve of extended chaotic maps.

Now, we examine the probability to solve a discrete logarithm problem in a time-bound manner by the adversary.

In the simulation, the oracle flops taking $h(ID_i, A\ )$ causes irregularity with the probability of at least $\frac{q_{H_1}}{p}$. The simulation process is running$q_s + q_E$ times (ensuing to $h(ID_i, A\ )$ and may like method be queried in the singing oracle, if identity has not been requested in the extraction oracle) with probability happening

$$\left(1 - \frac{(q_E + q_{s)}q_{H_1}}{p}\right) \leq \left(1 - \frac{q_{H_1}}{p}\right)^{q_s + q_E}$$

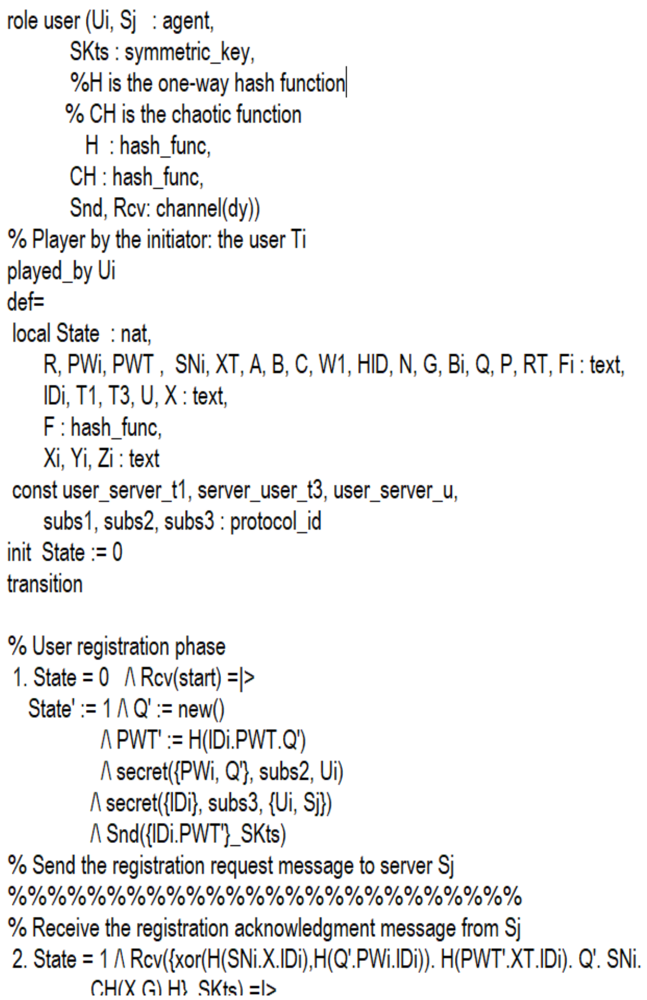
Since the random oracle gives the random values, an inquiry $h(Q^*, B^*, m^*\ )$ with a probability of at least $(1 - \frac{1}{p})$ exists. System $\beta$ guesses it is exactly as the motive for reverse with probability at least $(\frac{1}{q_{H_2}})$.Hereafter, the probability is:

$$\left(1 - \frac{1}{p}\right)\left(\frac{1}{q_{H_2}} - \frac{(q_E + q_s)q_{H_1}}{p}\right)$$

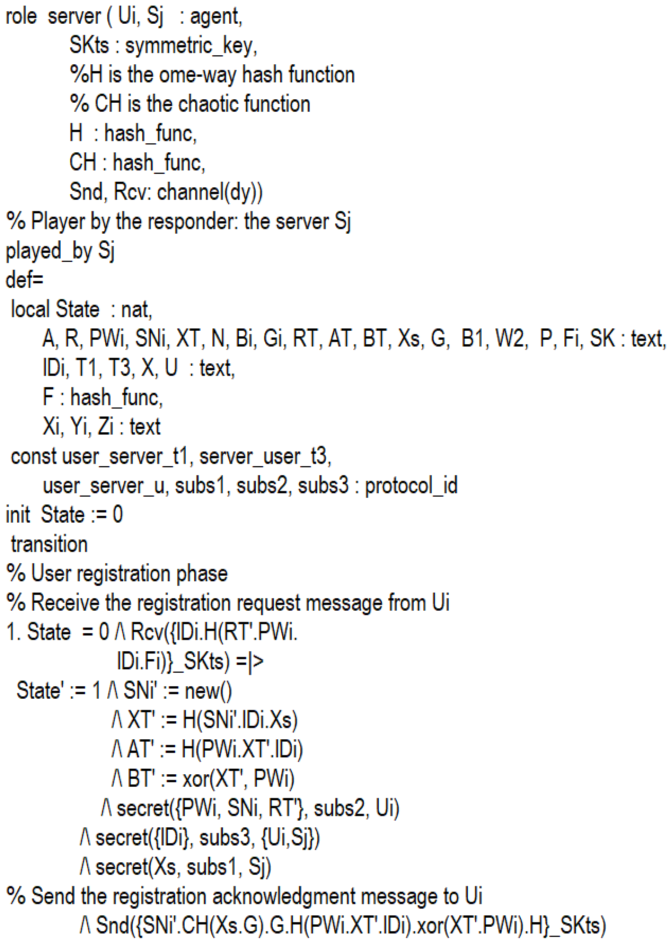
The system's time complexity is directed exponentially and is performed in the signing and extracts inquiries that are equivalent to $T' = T + O(q_s + q_E)\tau$.

## 6. Implementation

We simulate the proposed CLS scheme under the widely accepted "Automated Validation of Internet Security Protocol and Application" (AVISPA). AVISPA includes four ends to implement the different segments of analysis mechanism. The proposed ECLSS scheme is implemented using CL-ATSE and OFMC backend. Overall descriptions about CL-ATSE and OFMC are available in \cite{armando2005avispa, armando2006avispa}. The required security protocols are simulated in "High-Level Protocol Specification" (HLPSL).Then, the intermediate format generates the output on all four ends after accepting the input from one.



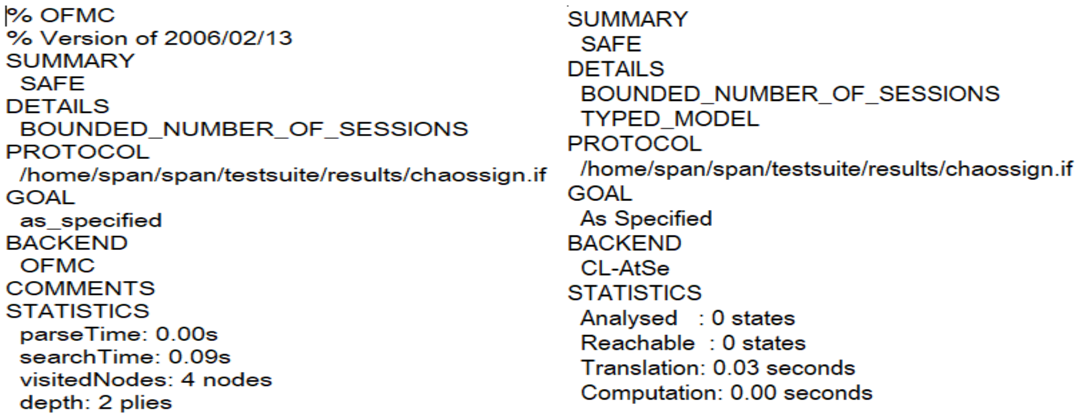
```
role user (Ui, Sj  : agent,
          SKts : symmetric_key,
          %H is the one-way hash function
         % CH is the chaotic function
            H  : hash_func,
          CH : hash_func,
          Snd, Rcv: channel(dy))
% Player by the initiator: the user Ti
played_by Ui
def=
 local State  : nat,
      R, PWi, PWT ,  SNi, XT, A, B, C, W1, HID, N, G, Bi, Q, P, RT, Fi : text,
      IDi, T1, T3, U, X : text,
      F : hash_func,
      Xi, Yi, Zi : text
 const user_server_t1, server_user_t3, user_server_u,
      subs1, subs2, subs3 : protocol_id
init  State := 0
transition

% User registration phase
 1. State = 0   /\ Rcv(start) =|>
   State' := 1 /\ Q' := new()
              /\ PWT' := H(IDi.PWT.Q')
              /\ secret({PWi, Q'}, subs2, Ui)
             /\ secret({IDi}, subs3, {Ui, Sj})
             /\ Snd({IDi.PWT'}_SKts)
% Send the registration request message to server Sj
%%%%%%%%%%%%%%%%%%%%%%%%
% Receive the registration acknowledgment message from Sj
 2. State = 1 /\ Rcv({xor(H(SNi.X.IDi),H(Q'.PWi.IDi)). H(PWT'.XT.IDi). Q'. SNi.
             CH(X.G).H}_SKts) =|>
```



```
role  server ( Ui, Sj   : agent,
          SKts : symmetric_key,
          %H is the ome-way hash function
          % CH is the chaotic function
          H  : hash_func,
          CH : hash_func,
          Snd, Rcv: channel(dy))
% Player by the responder: the server Sj
played_by Sj
def=
 local State  : nat,
     A, R, PWi, SNi, XT, N, Bi, Gi, RT, AT, BT, Xs, G,  B1, W2,  P, Fi, SK : text,
     IDi, T1, T3, X, U  : text,
     F : hash_func,
     Xi, Yi, Zi : text
 const user_server_t1, server_user_t3,
     user_server_u, subs1, subs2, subs3 : protocol_id
init  State := 0
 transition
% User registration phase
% Receive the registration request message from Ui
1. State  = 0 /\ Rcv({IDi.H(RT'.PWi.
                  IDi.Fi)}_SKts) =|>
  State' := 1 /\ SNi' := new()
                 /\ XT' := H(SNi'.IDi.Xs)
                 /\ AT' := H(PWi.XT'.IDi)
                 /\ BT' := xor(XT', PWi)
                /\ secret({PWi, SNi, RT'}, subs2, Ui)
             /\ secret({IDi}, subs3, {Ui,Sj})
             /\ secret(Xs, subs1, Sj)
% Send the registration acknowledgment message to Ui
             /\ Snd({SNi'.CH(Xs.G).G.H(PWi.XT'.IDi).xor(XT'.PWi).H}_SKts)
```

**Fig 6:** Role specification for the User and Verifier in ECLSS

All stages of the proposed ECLSS are simulated in HLPSL. We introduced the four characters, defined as a user, session, server and environment in HLPSL simulation. In the initial mode, any $i^{th}$ user $U_i$ gets started from signal receiving and updates the initial state from 0 to1. $U_i$ requests with registration $ID_i$ through secure channel $Snd()$.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/chaossign.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.09s
  visitedNodes: 4 nodes
  depth: 2 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/chaossign.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.03 seconds
  Computation: 0.00 seconds
```

**Fig 7:** The analysis result by using OFMC and CL-AtSe of ECLSS

After getting the request from $U_i$, KGC approves the request and sends$< A, C >$ to $U_i$. $U_i$ inputs the identity $ID_i$ with biometric uniform randomly 1 to login the device. Then it sends request$< Q, B, W, C, D >$to KGC for a signature verification. Upon $Rcv()$ the request from $U_i$, the verifier generates a random number b and calculates $< V', N', b' >$. Thus, the verifier responds on behalf of trusted KGC. The security of $Rcv()$, Snd() are based on Dolev-Yao threat model [37].

## 7. Performance analysis

In this segment, we examine the performance analysis of the proposed ECLSS below:

### 7.1 Comparison of the Operational Cost

In this subsection, we illustrate the efficiency comparison between five efficient CLS schemes: Zhang and Zhang [15], Xiong et al. [17], He et al. [24], Karati et al. [27] and the ECLSS scheme. The first two CLS scheme, such as Zhang and Zhang [15], Xiong et al. [17] used pairing operations with scalar multiplication. Pairing operation cost is 20 times higher than the multiplication cost, which makes the CLS scheme very costly [24]. He et al. [24] and Karati et al. [27] ignore their CLS scheme's pairing cost, making it more effective. Our ECLSS scheme uses extended chaotic properties in the proposed ECLSS scheme that takes only one scalar multiplication during the execution of the sign algorithm and 2 scalar multiplications during the execution of the verification algorithm. We validate that the proposed ECLSS scheme is proven effective when compared with the other previous efficient schemes based on Table 2.

**Table 2:** Comparison between operational costs of previous CLS

| CL-AS | Sign | Individual verification | Total cost (sign + verification) | Hardness assumption |
|---|---|---|---|---|
| Zhang and Zhang [15] | 3 $T_s$ | 4 $T_p$ | $4T_p + 2T_s$ | ECDHP |
| Xiong et al. [17] | 3 $T_s$ | $3T_p + 2T_s$ | $3T_p + 5T_s$ | ECDHP |
| He et al. [24] | 1 $T_s$ | 3 $T_s$ | 4 $T_s$ | ECDLP |
| Karati et al. [27] | 1 $T_s$ | 3 $T_s$ | 4 $T_s$ | ECDLP |
| **ECLSS** | 1 $T_s$ | 2 $T_s$ | 3 $T_s$ | CDLP |

*$T_s$–time for Scalar multiplication, $T_p$ -time for Pairing, ECDLP-Elliptic Curve Discrete Logrithm Problem, ECDHP-Elliptic Curve Diffie-Hellman Problem, CDLP- Chaotic Discrete Logarithm Problem*

### 7.2 Comparison of the Computation Cost

**Table 3:** Comparison between the computation costs of previous CLS

| CL-AS | Sign | Individual verification | Total cost (sign + verification) | Total running time | Hardness assumption |
|---|---|---|---|---|---|
| Zhang and Zhang [15] | 3 $T_s$ | 4 $T_p$ | $4T_p + 3T_s$ | 14.34 *ms* | ECDHP |
| Xiong et al [17] | 3 $T_s$ | $3T_p + 2T_s$ | $3T_p + 5T_s$ | 12.13 *ms* | ECDHP |
| He et al. [24] | 1 $T_s$ | 3 $T_s$ | 4 $T_s$ | 2 *ms* | ECDLP |
| Karati et al. [27] | 1 $T_s$ | 3 $T_s$ | 4 $T_s$ | 2 *ms* | ECDLP |
| **ECLSS** | 1 $T_s$ | 2 $T_s$ | 3 $T_s$ | 1.8 *ms* | CDLP |

*$T_s$–time for Scalar multiplication, $T_p$ -time for Pairing, ECDLP-Elliptic Curve Discrete Logrithm Problem, ECDHP-Elliptic Curve Diffie-Hellman Problem, CDLP- Chaotic Discrete Logarithm Problem*

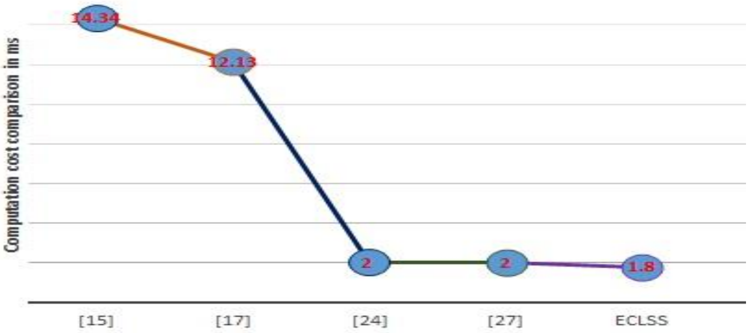


**Fig 8:** Comparison of the computation

## 8. Conclusion

We have designed an efficient novel Certificateless signature scheme (ECLSS) in the paper. The proposed ECLSS is based on the properties of the extended chaotic map and is protected against adversaries under the difficult assumption of the DLP proven by the universally accepted random oracle model. We demonstrated that the proposed ECLSS scheme is secure against security attacks and achieves security requirements. Also, the comparisons among the performance based on the implementation in “Avispa” with other related CLS signature schemes confirm that our ECLSS scheme using chaotic properties is computationally efficient. Thus, ECLSS is suitable for real application in the communication system.

## References

[1] M S Baptista, “Cryptography with Chaos”. Mar. 1998. url: http://cmup.fc.up.pt/cmup/murilo.baptista/baptist-a_PLA1998.pdf.

[2] Nathan Holt, “Chaotic Cryptography: Applications of Chaos Theory to Cryptography”, 2014.

[3] F. Dachselt, W. Schwarz "Chaos and cryptography”. IEEE Trans Circuits Syst I Fundam Theory Appl vol. 48(12), 1498–1509, 2005.

[4] A. Politi, “Lyapunov exponent”. url: http://scholarpedia.org/ article/Lyapunov_exponent.

[5] A. Shamir, “Identity Based Cryptosystems and Signature Schemes ‘ Crypto’84, LNCS 196, 1984.

[6] S. Al-Riyami, K. Paterson” Certificateless Public Key Cryptography”, Asiacrypt’ 03, LNCS 2894, 452-473, 2003.

[7] X. Wang, N. Guan, H. Zhao, S.Wang, Y. Zhang, “A new image encryption scheme based on coupling map lattices with mixed multi-chaos” Scientific Report, 10,2020. DOI10.1038/s41598-020-66486-9.

[8] D. H.Yum, P. J Lee, “Generic Construction of Certificateless Signature”, Information Security and Privacy, LNCS3108, 200–211, 2004. DOI 10.1007/978-3-540-27800-9_18.

[9] M. Gorantla, A. Saxena, “An Efficient Certificateless Signature Scheme”, In Computational Intelligence and Security, LNCS 3802,110–116 DOI 10.1007/11596981_16.

[10] Z. Gong, Y. Long, X. Hong, K. Chen, “Two Certificateless Aggregate Signatures from Bilinear Maps”, Proceedings of the IEEE SNPD, 3, 188–193, 2007. DOI 10.1109/snpd.2007.132.

[11] X. Huang, Y. Mu, W. Susilo, D. S. Wong, W. Wu, “Certificateless Signatures: New Schemes and Security Models”, The computer journal, vol 55(4),457-474, 2012. DOI 10.1093/comjnl/bxr097.

[12] Y.C. Chen, R. Tso, W. Susilo, X. Huang, G Horng., “Certificateless Signatures: Structural Extensions of Security Models and New Provably Secure Schemes”, IACR Cryptology ePrint Archive.

[13] L. Zhang, B. Qin, Q. Wu, F. Zhang, “Efficient Many-to-One Authentication with Certificateless Aggregate Signatures”, Comput. Netw. Vol 54 (14) pp 2482–2491, 2010. DOI 10.1016/j.comnet.2010.04.008.

[14] L. Cheng, Q. Wen, Z. Jin, H. Zhang, L. Zhou, “Cryptanalysis and Improvement of a Certificateless Aggregate Signature Scheme”, Inform. Sci. 295, 337-46, 2015.DOI 10.1016/j.ins.2014.09.065

[15] L. Zhang, F. Zhang, “A New Certificateless Aggregate Signature Scheme”, Comput. Commun. Vol

32 (6),1079–1085, 2009. DOI 10.1016/j.comcom.2008.12.042.

[16] P. Kumar, S. Kumari, V. Sharma, A. K. Sangaiah, J. Wei, and X. Li, "A certificateless aggregate signature scheme for healthcare wireless sensor network," Sustainable Computing: Informatics and Systems, vol. 18, pp. 80–89, 2018. DOI 10.1016/j.suscom.2017.09.002.

[17] H. Xiong, Z. Guan, Z. Chen, F Li, "An Efficient Certificateless Aggregate Signature with Constant Pairing Computation", Inform. Sci. vol 219, pp 225–235, 2014.DOI 10.1016/j.ins.2014.07.019

[18] K-H.Yeh, K-Y.Tsai, C-Y.Fan. "An efficient certificateless signature scheme without bilinear pairings". Multimedia Tools and Applications, vol 74(16), 6519–6530, 2015. DOI 10.1007/s11042-014-2154-4

[19] S. Jye, "A speech encryption using fractional chaotic systems," Nonlinear Dynamics, vol. 65, pp. 103–108, 2011.

[20] X. Wang, X. Wang, J. Zhao, and Z. Zhang, "Chaotic encryption algorithm based on alternant of stream cipher and block cipher," Nonlinear Dynamics, vol. 63, no. 4, pp. 587–597, 2011.

[21] S. Deng, Y. Li, and D. Xiao, "Analysis and improvement of a chaos-based Hash function construction," Communications in Nonlinear Science and Numerical Simulation, vol. 15, no. 5, pp. 1338–1347, 2010.

[22] D. Xiao, F. Y. Shih, and X. Liao, "A chaos-based hash function with both modification detection and localization capabilities," Communications in Nonlinear Science and Numerical Simulation, vol. 15, no. 9, pp. 2254–2261, 2010.

[23] Y. Wang, K.-W. Wong, X. Liao, and T. Xiang, "A block cipher with dynamic S-boxes based on tent map," Communications in Nonlinear Science and Numerical Simulation, vol. 14, no. 7, pp. 3089–3099, 2009.

[24] D. He, J. Chen, R. Zhang, "An efficient and provably-secure certificateless signature scheme without bilinear pairings", Int. J. Commun. Syst., vol 25, pp 1432–1442, 2012. DOI 10.1002/dac.1330

[25] C.-C. Lee, C.-L. Chen, C.-Y. Wu, and S.-Y. Huang, "An extended chaotic maps-based key agreement scheme with user anonymity," Nonlinear Dynamics, vol. 69, no. 1-2, pp. 79–87, 2012.

[26] C.-C. Lee and C.-W. Hsu, "A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps," Nonlinear Dynamics, vol. 71, no. 1-2, pp. 201–211, 2013.

[27] A. Karati, SK H Islam, G.P. Biswas "A Pairing-free and Provably Secure Certificateless Signature Scheme", Information Sciences, vol 450, pp 378-391, 2018. DOI 10.1016/j.ins.2018.03.053

[28] C.-C. Lee, C.-T. Li, S.-T. Chiu, and Y.-M. Lai, "A new three party-authenticated key agreement scheme based on chaotic maps without password table," Nonlinear Dynamics, vol. 79, no. 4, pp. 2485–2495, 2014.

[29] C.-C. Lee, D.-C. Lou, C.-T. Li, and C.-W. Hsu, "An extended chaotic-maps-based protocol with key agreement for multiserver environments," Nonlinear Dynamics, vol. 76, no. 1, pp. 853–866, 2014.

[30] K Chain, WC Kuo WC (2013) "A new digital signature scheme based on chaotic maps", Nonlinear Dynamics, vol. 74 (4), pp 1003-1012, 2013. DOI 10.1007/s11071-013-1018-1.

[31] N.Tahat, A. K. Alomari, A. A. Freedi, O. M. A. Hazaimeh, M.F. A.Jamal "An Efficient Identity-Based Cryptographic Model for Chebyhev Chaotic Map and Integer Factoring Based Cryptosystem", Journal of Applied Security Research, Vol. 14, 2019. 10.1080/19361610.2019.1621513.

[32] SK. H. Islam "Identity-based encryption and digital signature schemes using extended chaotic maps". IACR Cryptol. ePrint Arch. 2014: 275 (2014)

[33] C. Meshram, R. W. Ibrahim, A. J. Obaid, S. G. Meshram, A. k. Meshram, A. Mohamed, A. E. Latif "fractional chaotic maps based short signature scheme under human-centered IoT environments" Journal of Advanced Research 32 (2021) 139–148. DOI 10.1016/j.jare.2020.08.015

[34] Meshram C, Lee CC, Meshram SG, Li CT. An efficient ID-based cryptographic transformation model for extended chaotic-map-based cryptosystem. Soft Comput 2019;23(16):6937–46.

[35] C. Meshram, C. T. Li, S. G. Meshram, " An efficient online/offline ID-based short signature procedure using extended chaotic maps" Soft computing, vol 23 (3) pp 747-753, 2019. DOI 10.1007/s00500-018-3112-2

[36] M. Bellare, C. Namprempre, G. Neven, "Security proofs for identity based identification and

signature schemes”. J Cryptol, vol. 22, pp 1–61, 2004.

[37] V. Kumar, M. Ahmad, D. Mishra, S. Kumari and M. K. Khan, “RSEAP: RFID based Secure and Efficient Authentication Protocol for Vehicular Cloud Computing”, Vehicular Communications, Elsevier, 22, 2020, 10021326, DOI: 10.1016/j.vehcom.2019.100213

[38] S. Han, E. Chang,”Chaotic map based key agreement without clock synchronization”. Choas Soliton Fractals vol. 39(3), pp1283–1289, 2009.