Attribute-Based Authentication Based on Biometrics and RSA-Hyperchaotic Systems

Original Scientific Paper

Safa Ameen Ahmed*
Information Technology & Communications University,
Informatics Institute for Postgraduate Studies, Baghdad, Iraq
University of Technology,
College of Chemical Engineering,
Department of Chemical Engineering and Petroleum Pollution, Baghdad, Iraq
safa.a.ahmed@uotechnology.edu.iq

Ali Maki Sagheer
University of Anbar, College of Computer Science and Information Technology,
Department of Computer Networks Systems, Ramadi, Iraq
Ali_makki@uoanbar.edu.iq

*Corresponding author

Abstract *– Attribute-based authentication is a security technique that allows access to resources according to characteristics of human biometrics. It confirms the security and increases the efficiency of applications, devices, and resources by controlling them and avoiding cyberattacks. Standard feature extraction approaches can also give erroneous results and limit processing efficiency when processing complex biometric data. This paper proposes a hybrid method combining the Rivest-Shamir-Adleman (RSA) method with 6D hyperchaotic systems to generate dynamic and sophisticated keys for key exchange in a more secure and effective authentication system. User attributes like ID and fingerprint are used for attribute-based authentication by extracting fingerprint features (Minutiae extraction) for biometric matching. The 6D-hyperchaotic systems generate dynamic values over time, influenced by the initial values and input constants. Three of the generated sequences were used on the sender side, and the other three sequences were used on the receiver side after processing them to satisfy the RSA condition. The time consumed for generation numbers is about 0.0717 msec. The results indicated that the system that produces the keys has robust resistance to statistical attacks. The average time for authentication is 0.307867 sec. Thus, the research presents an integrated solution that improves authentication system security and dependability, especially in critical areas that demand sophisticated data and user protection. The user identity verification reached 95.14% accuracy, and the proposed method could be integrated with the extended system by adding a node.*

Keywords: *Authentication, Fingerprint, Minutiae extraction, Hyperchaotic Systems, RSA, Public key, and Private key*

Received: March 13, 2025; Received in revised form: June 20, 2025; Accepted: July 18, 2025

1. INTRODUCTION

Due to the rapid development of digital systems and the growing demand for secure and efficient authentication methods, Attribute-Based Authentication (ABA) has emerged as a potential approach for improving cybersecurity. User attributes, including username, ID, phone number, email, gender, address, age, birthday, and biometric information are used to generate public and private keys for policy access, encryption, and decryption to protect user data [1, 2]. Recent ABE methods use elliptic curves and pairing-based encryption, including the integer factorization problem with RSA. Reduced ciphertext sizes, faster key and signature generation, and improved security [3]. RSA uses public and private keys for secure key exchange and secret transmission, producing digital signatures for data integrity and sender identity [4]. Advancements in technology and hardware enable RSA to offer robust security with adequate key lengths [5]. However, the benefit is that a substantial key size renders decryption using established algorithms challenging [6]. Multiple attacks have made token- or password-only authentication solutions less easy and reliable [7]. Biometrics are one of the most important features in authentication systems to prevent access control, impersonation, and fraud [8]. Due to biometric unique ability to verify the identity of individuals based on their physiological or behavioral characteris-

tics, such as fingerprints, iris, or voice patterns [9]. Fingerprints are identified by their lines and curves. Their lifelong reliability makes them reliable [10]. Fingerprint classification relies on global and local characteristics, with global features like Singular Point (SP) enhancing matching and alignment, and local features like minutiae and ridge features that most identification algorithms use in matching [11, 12]. Fingerprint recognition technology provides safe biometric authentication, with current developments in picture capture, preprocessing, feature extraction, and matching for dependable identification [13, 14]. As online services become more prevalent, protecting sensitive data is crucial due to the rise of sophisticated cyber threats like tampering, spying, and phishing [15]. In this environment, dependence exclusively on conventional security measures like passwords or firewalls is inadequate [16]. However, strong encryption techniques must be incorporated to ensure high security and prevent potential attacks, such as impersonation [17]. Integrating biometric authentication and utilizing personal data such as fingerprints or iris patterns with the RSA algorithm provides enhanced security. Biometric authentication enhances privacy and security, increases system complexity, reduces hacking risk, and replaces traditional methods such as those using passwords and ID numbers [18].

This paper aims to introduce a feature-based authentication system that integrates biometric feature extraction (fingerprint), the RSA algorithm, and a six-dimensional chaotic system to produce dynamic and extremely unpredictable keys. The study assesses the system's efficacy regarding authentication precision, cryptographic robustness, key generation unpredictability, and response latency, in comparison to conventional techniques.

The following section includes the structured rest of the article. Section 2 presents a review of the relevant literature. The 6D Hyperchaotic system is described in section 3. The proposed method is detailed in Section 4. The experimental results are discussed in Section 5. Section 6 offers the conclusion of this work.

2. RELATED WORK

The proposed method can be divided into three sections: Biometric identification using fingerprint matching, key generation using a 6D hyperchaotic system, and authentication using RSA and ABE. Therefore, the related work will be divided into three sections.

In the first section of fingerprint matching, S. Socheat and T. Wang (2020) proposed a fingerprint enhancement, extraction, and matching architecture. Three steps made up this system. Preprocessing the fingerprint surface with brightness and Gabor filters darkens the ridgelines to improve image quality.

Next, extract features via binarizing, thinning, localizing minutiae, removing erroneous minutiae, and classifying to locate and count fingerprint minutiae.

The final step is minutiae matching to validate an individual's identity by comparing findings to the database. Previously registered individuals' results will be validated [19]. Situmorang and Herdianto (2022) developed a biometric identification system using fingerprint images from mobile phone cameras. The methodology involved pre-processing steps, including cropping, grayscale conversion, binarization, and thinning of fingerprint images, then fine feature extraction using the Crossing Number (CN) and minutiae-based matching methods. The accuracy of fingerprint fine-grained verification was up to 92.8% using a 5MP camera and 95.11% using an 8MP camera in fingerprint recognition. The average accuracy of fingerprint fine-grained verification was found to be 63% [20].

In the second section, the key generation using a hyperchaotic system, Akif *et al.* (2021) presented a Pseudorandom Bit Generator (PRBG) that utilizes a new two-dimensional chaotic logistic map, including mouse movement data and chaotic systems to generate unpredictability. The system employs algorithms to produce pseudorandom bits derived from mouse movement coordinates, augmenting nonlinearity and randomness. The control parameters a and b are established at a=0.8 and b within the range of 2 to 5, thereby guaranteeing non-redundancy in the chaotic system [21].

In the third section, the authentication section, Tahat *et al.* (2020) produced a novel secure cryptosystem by integrating the Dependent-RSA (DRSA) with chaotic maps (CM) that rely on both integer factorization and Chaotic Maps Discrete Logarithm (CMDL) [22]. Zhang *et al*. (2024) presented the Chebyshev–RSA Public Key Cryptosystem with Key Identification (CRPKC-Ki) public key cryptography algorithm, which improves security and efficiency by integrating Chebyshev polynomials with RSA. The procedure is achieved by selecting two large-prime integers and computing N and the totient, and presents alternative multiplication coefficients K_i, generated by Chebyshev chaotic mapping, established by the shared secret selection criteria among participants [23].

3. 6D HYPERCHAOTIC SYSTEM

The 6D hyperchaotic Systems exhibiting complex and implicit extreme multi-stability demonstrate the following dynamic phenomena on a line or equilibrium plane: hidden extreme multi-stability, transient chaos, bursting, and offset boosting. This chaotic system is the inaugural high-order system to manifest all these intricate dynamic behaviors [24].

$$\dot{x}_1=\alpha(1-\beta|x_6|)\,x_2-ax_1 \quad (1)$$

$$\dot{x}_2=cx_1+dx_2-x_1x_3+x_5 \quad (2)$$

$$\dot{x}_3=-bx_1+x_1^2 \quad (3)$$

$$\dot{x}_4=ex_2+fx_4 \quad (4)$$

$$\dot{x}_5=-rx_1-kx_5 \quad (5)$$

$$\dot{x}_6=-x_2 \quad (6)$$

The mentioned chaotic system comprises six equations, Eq. (1), Eq. (2), Eq. (3), Eq. (4), Eq. (5), and Eq. (6), that include variables and parameters. $x1$, $x2$, $x3$, $x4$, $x5$, and $x6$ represent the state variables, while $\alpha, \beta, a, b, c, d, e, f, r$, and k denote the system parameters. [25]. These equations constitute a highly intricate model that illustrates the interrelated dynamic interactions inside a chaotic system, showcasing the system's capacity to produce various states of dynamic stability and unstable motion, oscillating between phases of chaos and complicated dynamic equilibria. This system comprises a set of differential equations that delineate the temporal evolution of each of the six dynamic variables, contingent upon the present values of the other variables.

4. PROPOSED METHOD

The proposed method is utilized to verify the individuals who are using the systems that are part of the network. This process is carried out more simply by employing biometric characteristics to verify the individual's identity. Additionally, this process is carried out by creating a set of keys to prevent unauthorized access to the systems. Fig. 1 shows the structure of the general proposed model.

Fig. 1. General proposed model

First, the system authenticates users by scanning their fingerprints, performing feature extraction, and executing matching actions to train the system and give the user his unique ID during the registration phase which is conducted by an offline mechanism to guarantee user attribute information such as the username, identification number, phone number, email, gender, address, age, birthday, and biometric information must be secret. The server also generates the private key (E_c, N_c) and the public key (D_c, N_c) for the user; the private key for the server is (E_s, N_s), and the public key is (D_s, N_s), which are used for authentication. Upon attempting to log in to the system, the user's fingerprint is scanned then the identical preprocessing, feature extraction, and matching protocols employed during the training phase are performed to identify the user. Authentication takes place in the initial step following user identification. However, he will be denied access to the system if the user is not identified. During the second step, the user digests his ID using his private key (E_c, N_c) in agreement with the server.

The result is encrypted with the server's public key (D_s, N_s) and sent to the server. The server decrypts the data using its private key (E_s, N_s), the result is decrypted using the user's public key (D_c, N_c) to obtain the ID, and correlates it with the user's stored ID. This procedure confirms the user's authorization to use the system. Fig. 2 shows the proposed Attribute-Based Authentication method.

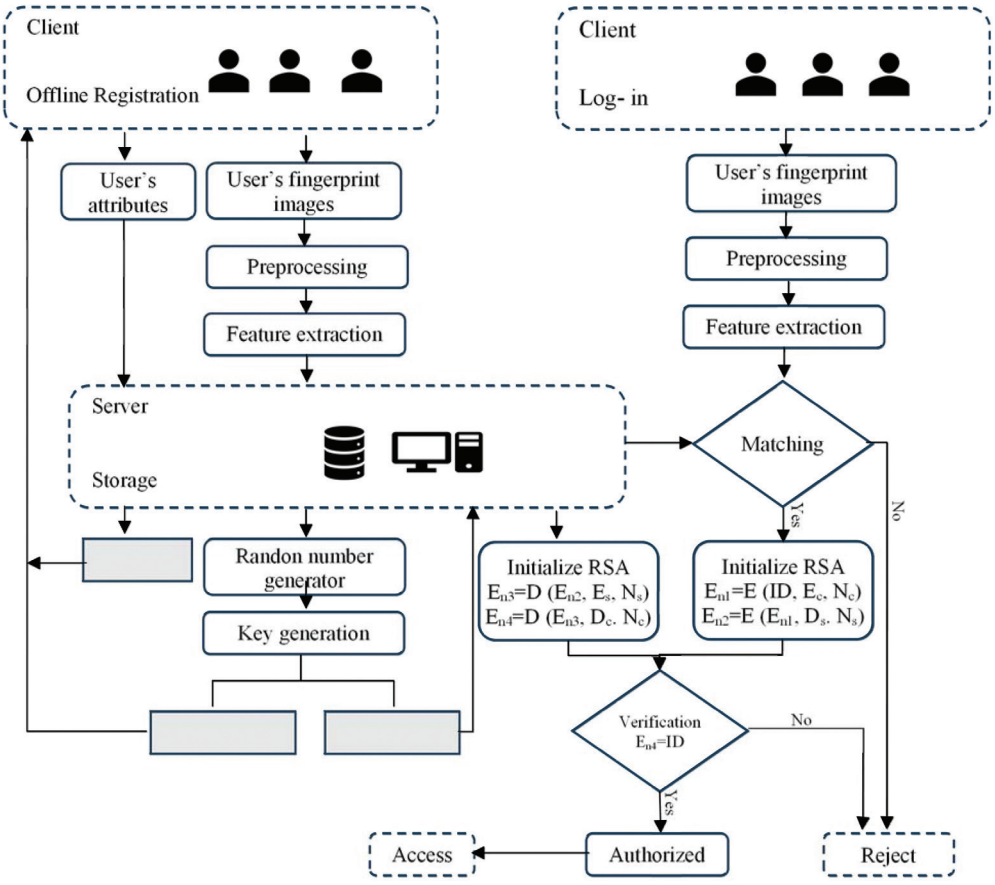


Fig. 2. The proposed Attribute-Based Authentication Method

4.1. REGISTRATION PHASE

The registration phase is conducted by an offline mechanism to guarantee that user attribute information, such as the username, identification number, phone number, email, gender, address, age, birthday, and biometric information, is kept secret. All user attributes are stored on the server. Then, the server gives the user their unique ID during the registration phase. The proposed system starts by scanning the user's fingerprint, preprocessing it, and extracting features from the fingerprint image used to train the system.

4.1.1. Preprocessing

Several processes have been performed on row fingerprint images in the dataset to generate standardized images. This ensures that the system receives consistent inputs. The preprocessing steps are noise removal, histogram equalization, resizing, binarization, and thinning. Fig. 3 shows the block diagram for the preprocessing step.

Fig. 3. The preprocessing stages

Noise removal is essential for reducing noise during fingerprint acquisition. As a result, a fine fingerprint image is obtained, and valleys and ridges are detected, as shown in Fig. 4.

(a) (b)

Fig. 4. Noise removal: (**a**) Original fingerprint image and (**b**) Noise removal fingerprint image

Histogram equalization improves grayscale image contrast by redistributing pixel values, reducing tonal density, creating a more homogeneous distribution of pixels, and enhancing detail clarity. This strategy greatly improves low-contrast images and improves feature extraction by highlighting fingerprint ridges and valleys, as shown in Fig. 5.

(a) (b)

Fig. 5. Histogram equalization (**a**) noise removed image, and (**b**) histogram equalized image

In the resizing step, the fingerprint image dimensions and orientation are changed to a standard size, and feature matching depends on this step, as shown in Fig. 6.

(a) (b)

Fig. 6. Image Resizing (**a**) equalized image, and (**b**) resized image

Finally, the fingerprint image is thinned to one-pixel lines without losing the structure. Biometric identification systems reduce changes via fingerprint analysis and minutiae extraction. This method enhances binary image feature extraction, recognition, and analysis, as shown in Fig. 7.

(a) (b)

Fig. 7. Edge Detection (**a**) resized image, and (**b**) thinning edge image

The preprocessing steps are crucial for enhancing the fingerprint image for the following processes, including feature extraction and feature matching, which improves the efficacy of the fingerprint verification system.

4.1.2. Feature extraction

The CN method is used to extract features from fingerprint images to identify the individual carrying the fingerprint. This process involves identifying minutiae, such as ridge ends and bifurcation points, and classifying them using coordinates, angles, and types. The crossing number method is based on studying the pixel arrangement around the fingerprint, which helps classify minutiae, as shown in Eq. (7).

$$CN = \frac{1}{2}\sum_{i=1}^{8}|P_i - P_{i-1}| \tag{7}$$

Where P_i is currently pixel while the $P_{i\text{-}1}$ Is the neighbor pixel. The extracted features are used for future matching. This method enhances fingerprint verification reliability by matching only relevant data. Fig. 8 shows feature extraction of the biometric image.

(a) (b)

Fig. 8. The feature extraction (**a**) thinning edge image, and (**b**) feature extraction of fingerprint image

4.2. LOG-IN PHASE

In this phase, the fingerprint image of the user is processed in the same way that the user's fingerprint is processed in the registration phase of the system, including preprocessing and feature extraction.

4.3. IDENTIFICATION PHASE

In this phase, the feature extraction obtained from the registration phase is used to train and test the system in order to identify the user in the log-in phase.

4.3.1. FEATURE MATCHING

In this phase, the extracted features of the two fingerprints are compared using a similarity score to verify if the two fingerprints match and belong to the same user. The Orientation-based Matching (OM) technique was employed to address the issue of misorientation fingerprint images resulting from inadequate image acquisition, which leads to increased matching accuracy. The feature matching procedure starts by computing the similarity of the features extracted from the two fingerprint images by calculating the Euclidean distance and the absolute orientation difference between the features extracted from the two fingerprint images to confirm the feature identity. Eq. (8) and Eq. (9) show the Euclidean distance and the absolute orientation difference.

$$distance = \sqrt{x^2 + y^2} \tag{8}$$

$$\theta = |\theta_1 - \theta_2| * \frac{180}{\pi} \tag{9}$$

Finally, in the feature-matching procedure, the ultimate similarity score is computed based on the number of matched features and their respective distances, employing Eq. (10).

$$score = \sqrt{\frac{2n^2}{f_1^2 + f_2^2}} \tag{10}$$

Where n is the number of matched features, $f1$ and $f2$ are the total number of features in the two fingerprints being compared.

4.4. AUTHENTICATION PHASE

This phase includes generating random numbers using a Six-Dimensional chaotic system, processing the number generator to check if the numbers satisfy the RAS condition, generating keys (private key and public key) for both the server and client, storing them in the server, and using RSA to authenticate the user.

4.4.1. RANDOM NUMBER GENERATOR

The 6D hyperchaotic system mentioned above consists of six equations, Eq. (1), Eq. (2), Eq. (3), Eq. (4), Eq. (5), and Eq. (6), containing variables and parameters. The state variables are $x1, x2, x3, x4, x5, x6$, and $\alpha, \beta, a, b, c, d, e, r$, and k are system parameters. When initial points ($x1(0)$, $x2(0)$, $x3(0)$, $x4(0)$, $x5(0)$, and $x6(0)$) = (0.0512, 0.0011, 0.2023, 1.0054, 0.0075, and 0.5001) Consequently, and under α = 15.4778, β = 0.1291, a = 4.8234, b =1.8789, c = 8.5012, d = -0.0154, e= 1.0001, f = -0.0137, r = 0.1723, and k= 0.0019. The number of times in this system is employed is dictated by the need to determine the values of the state variables. This approach enables the server to generate a collection of random numbers for each user, which is utilized to create private and public keys and the session key for each user.

4.4.2. KEY GENERATION

The RSA algorithm utilizes integers from the preceding stage to fulfill the specifications. The numbers are multiplied by 104 to obtain integer components that meet the RSA requirements. The numbers are assessed for primality and incremented if they are not prime. In every cycle, six integers are generated from a 6D hyperchaotic map, including three for the server and three for the client. The initial prime number is P, the subsequent is Q, and the third is the private key (E). The public key (D) has been established. If a number does not satisfy the criteria, the six numbers are eliminated, and the identical method is applied to the server and client numbers.

The following steps briefly describe the key generation

Step1: use 6DHyper Chaotic System to generate (X_1, X_2, X_3, X_4, X_5, X_6)

Step 2: Multiply (X_1, X_2, X_3, X_4, X_5, X_6) by a power of ten, such as (104), to get an integer value

Step 3: All numbers are tested to be prime; if not prime, increment them by 1 until they reach the nearest prime.

Step 4: Set the result from the (X_1, X_2, X_3) as (P_s, Q_s, E_s) for the server. (P_s, Q_s are the two primes for the server, E_s is the private key for the server)

Set the result from the (X_5, X_6, X_4) as (P_c, Q_c, E_c) for the client. (P_c, Q_c are the two primes for the client, Ec is the private key for the client)

Step 5: $N_s = P_s \times Q_s$ (N_s: N for server)

$N_c = P_c \times Q_c$ (N_c: N for client)

$\varphi(N_s)=(P_s\text{-}1)\,(Q_s\text{-}1)$

$\varphi(N_c)=(P_c\text{-}1)\,(Q_c\text{-}1)$

Step 6: Find the P_s (public key for server) and P_c (public key for client)

$(E_s * D_s) \bmod \varphi(N_s)=1$

$(E_c * D_c) \bmod \varphi(N_c)=1$

4.4.3. INITIALIZE RSA

The private and public keys produced in the preceding stage are employed to initiate the RSA algorithm, where (E_s, N_s) represents the server's private key, and (D_s, N_s) denotes the server's public key. (E_c, N_c) constitutes the client's private key, while (D_c, N_c) represents the client's public key. The process occurs on the server to verify the client's legitimacy by recognizing and verifying the client's ID. Thus, both endpoints of the connection are authenticated, ensuring that any variations in these keys will render the connection between source and destination insecure. The efficacy of the encryption and decryption process is evaluated to confirm the authenticity and integrity of the resultant keys, hence ensuring the dependability of the outcomes in practical applications such as security systems and encrypted communications.

5. EXPERIMENTAL RESULTS

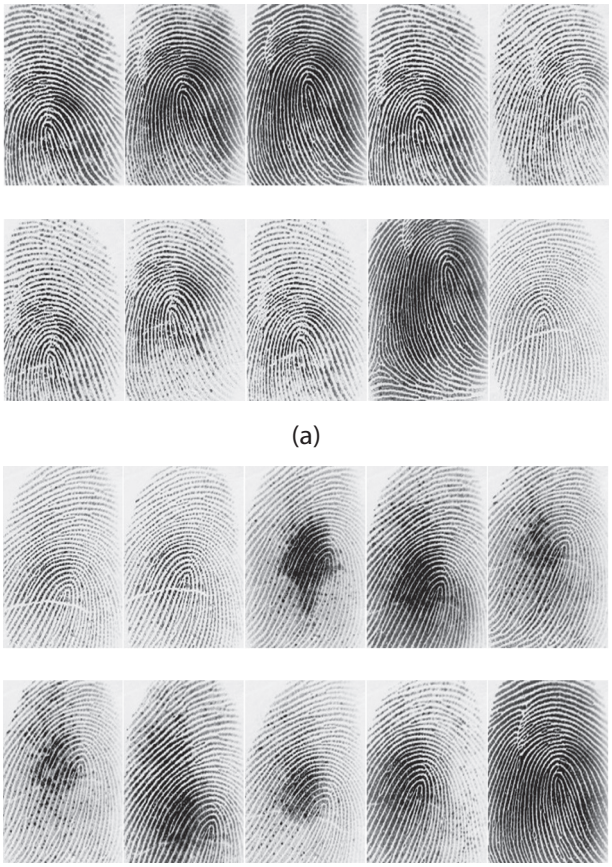
The experimental result for the proposed method is implemented and analyzed in MATLAB 2023b. This method utilized an Intel(R) Core (TM) i7-10510U CPU @ 1.80 GHz, 2.30 GHz processor, 16.0 GB (15.7 GB usable), and Windows 11 Home, 64-bit operating system, x64-based processor.

5.1. BIOMETRIC DATASET

Two datasets of fingerprint images are used to train and test the proposed method. The first one is Neurotechnology CrossMatch, which contains 408 fingerprint images for 51 fingers, with 8 impressions taken per finger. These images were captured using an optical scanner at 500 dpi with dimensions of 504×480 pixels. Those images are saved in TIFF format. Fig. 9 shows the fingerprint image of the Neurotechnology CrossMatch dataset.

Fig. 9. The Fingerprint image of the Neurotechnology CrossMatch dataset.

The second dataset is the local dataset, which consists of biometric images of the users' fingerprints. These biometric images are employed to authenticate the client. The fingerprint images are collected using a ZK9500 USB Fingerprint Scanner with a 280 MHz CPU, the fingerprint image quality is 2 megapixels and 500 dpi resolution. This dataset includes 1200 images of thumb fingerprints for 60 users; each user has 10 fingerprint images for the right hand and 10 fingerprints for the left hand. Fig. 10 shows the fingerprint image of the local dataset.



(a)

(b)

Fig. 10. Fingerprint images of the local dataset: (**a**) for the left thumb, and (**b**) for the right thumb

5.2. FINGERPRINT MATCHING RESULT

The Neurotechnology CrossMatch and local datasets improved as the number of images used in the experiment increased, but the local dataset had much superior accuracy in any scenario. The Neurotechnology CrossMatch dataset had 90.82% accuracy with 10 images, whereas the local dataset had 91.58%, demonstrating a minor advantage at low image counts. With 40 images, the Neurotechnology CrossMatch dataset's accuracy increased to 92.90%, while the local dataset achieved 93.08%, demonstrating its stability and capacity to gain from more data. The local dataset performed better at 50 images, scoring 95.14% against 92.24% for the Neurotechnology CrossMatch dataset, suggesting data adaptability as shown in Table 1.

Table 1. Accuracy results of fingerprint matching

Number of images	10	20	30	40	50
Neurotechnology CrossMatch	90.82	91.77	91.80	92.90	92.24
Local dataset	91.58	92.79	93.05	93.08	95.14

Fig. 11 shows that the local dataset surpasses the Neurotechnology CrossMatch dataset. When the number of images is limited, the disparity between the two datasets is negligible; however, when the number of images increases, the Local dataset acquires a distinct accuracy. This may be due to the local dataset aligning more closely with the test or the superior image analysis techniques.

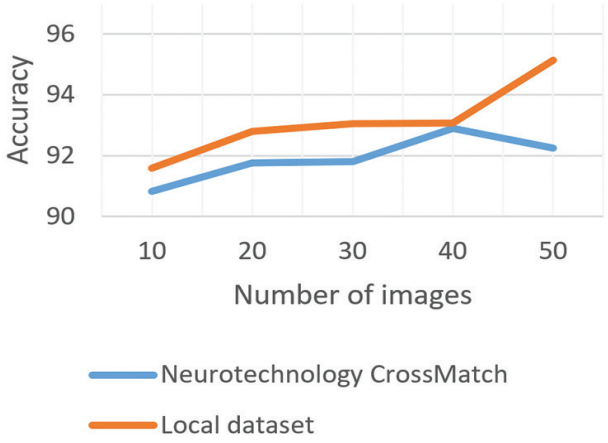
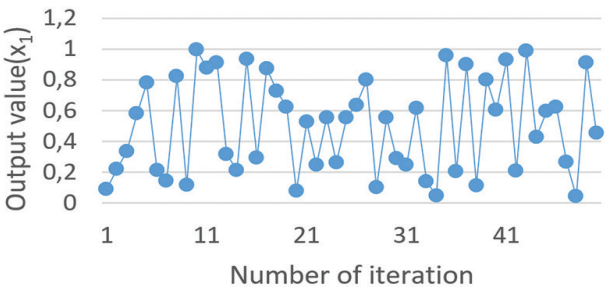


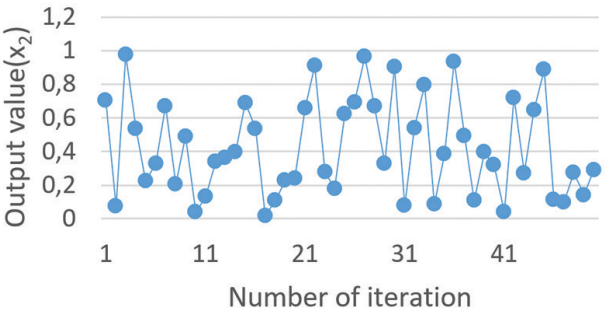
Fig. 11. Accuracy result of fingerprint matching

5.3. RANDOM NUMBER GENERATOR RESULT

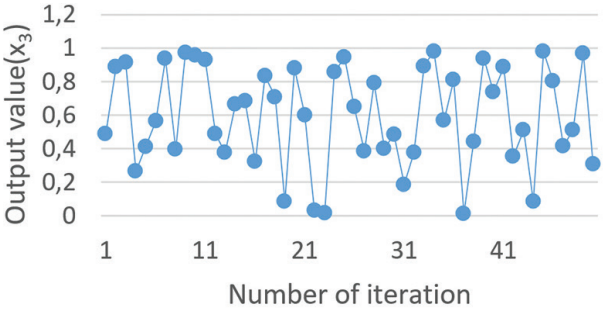
The 6D hyperchaotic system was employed to model the dynamics of a nonlinear system through mathematical equations reliant on interdependent and iterative variables. The system underwent testing through 12 computation cycles, during which each of the six primary variables ($x1$ to $x6$) exhibited complex periodic variations, indicative of the nonlinear interactions among the mathematical equations. The modulo function was utilized to normalize values inside the interval [0,1]. Fig. 12 explains the plotting of the six-dimensional sequences to visualize the randomness distribution.



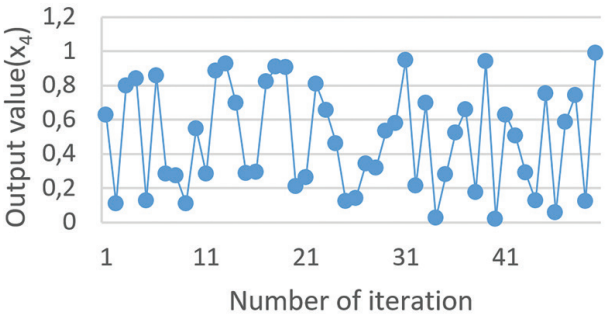
(a)



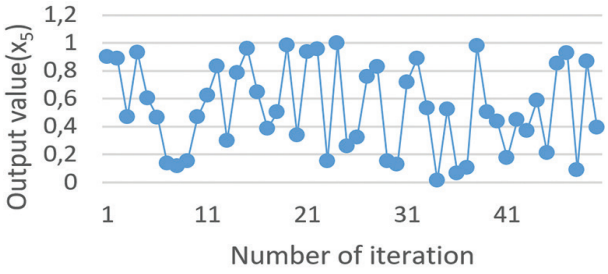
(b)



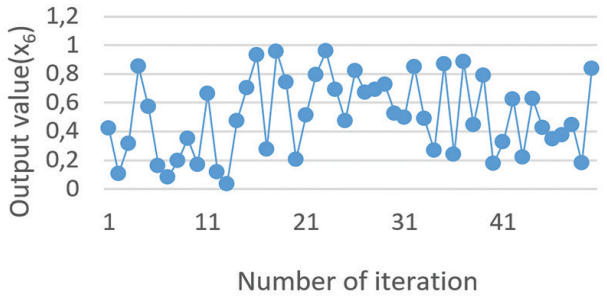
(c)



(d)



(e)



(f)

Fig. 12. The plotting of Six-Dimensional sequences (**a**) the first sequence (**b**) the second sequence (**c**) the third sequence (**d**) the fourth sequence (**e**) the fifth sequence, and (**f**) the sixth sequence

Table 2 shows that the values vary dynamically across twelve cycles, illustrating the system's nonlinear characteristics. Recursive functions and nonlinear equations significantly influence the ultimate values of each variable, and potential disorder in the values can be detected, rendering the application appropriate for the analysis of chaotic or cryptographic systems. This system demonstrated computation efficiency, completing the iterative process in a minimal duration, hence suggesting its appropriateness for time-sensitive applications. The results indicated that the system produces dynamic values that progress over time, influenced by the initial values and input constants. The mentioned values of the variables ($x1$ to $x6$) were processed and examined to transform them into prime numbers.

The values are multiplied by 10^4 to transform them into integers while preserving their precision. Subsequently, the primacy of each element is assessed. If the number is not prime, an iterative procedure is employed that increments the value by 1 until the first prime number is attained. The resultant prime numbers are retained.

Table 2. Random number generator results

iteration	$X1$	$X2$	$X3$	$X4$	$X5$	$X6$
1	0.2310	0.4324	0.0936	0.0127	0.0088	0.0011
2	0.8058	0.0012	0.4875	0.4326	0.0398	0.4324
3	0.9037	0.4973	0.8647	0.0071	0.1389	0.0012
4	0.0554	0.6104	0.5146	0.4975	0.1560	0.4973
5	0.1079	0.6078	0.1010	0.6172	0.0098	0.6104
6	0.1864	0.9470	0.2143	0.6163	0.0186	0.6078
7	0.4061	0.5782	0.3155	0.9555	0.0322	0.9470
8	0.8142	0.6213	0.9279	0.5914	0.0700	0.5782
9	0.8252	0.2455	0.8669	0.6294	0.1404	0.6213
10	0.4751	0.8743	0.2313	0.2541	0.1424	0.2455
11	0.3955	0.0852	0.6670	0.8779	0.0821	0.8743
12	0.0768	0.7093	0.8995	0.0972	0.0683	0.0852

The results presented in Table 3 demonstrate exceptional efficiency in the rapid and precise conversion of values to prime numbers, rendering them appropriate for applications necessitating numerical data analysis or the utilization of prime numbers, such as cryptography and the examination of dynamic systems.

Table 3. Prime number result

Iteration	X1	X2	X3	X4	X5	X6
1	2310	4324	936	127	88	11
2	8058	12	4875	4326	398	4324
3	9037	4973	8647	71	1389	12
4	554	6104	5146	4974	1560	4973
5	1079	6078	1010	6172	98	6104
6	1864	9470	2143	6163	186	6078
7	4061	5782	3155	9555	322	9470
8	8142	6213	9279	5914	700	5782
9	8252	2455	8669	6294	1404	6213
10	4751	8743	2313	2541	1424	2455
11	3955	852	6670	8779	821	8743
12	768	7093	8995	972	683	852

The National Institute of Standards and Technology (NIST) has 16 statistical tests that are widely used to analyze bit sequence randomness. The statistical significance level for each NIST test has been established at 0.01. Hence, Sequences pass a test if the P value is greater than 0.01 and less than 1. In the numerical experiment of the proposed method, 1000 sets of 106-bit sequences are created and randomly selected for NIST testing. The results in Table 4 showed high randomness, with all tests passing and P values exceeding all thresholds. The recommended method improved randomization statistical performance, with higher P-value averages in some tests than in reference [21]. These findings make random sequence generation suitable for encryption, cybersecurity, and statistical modeling that require high-quality random numbers.

Table 4. The P-value of the frequency test

Test type	Average of the p-value of the proposed method	Average of the p-value of the reference [21]	Status
Frequency test	0.488	0.663	success
Frequency within a block	0.290	0.528	success
Run Test	0.778	0.654	success
The longest run of ones in a block test	0.752	0.744	success
Fast Fourier Transform test	0.453267	0.666	success
Overlapping template matching test	0.200	0.412	success
approximate entropy test	0.513506	0.954	success
Cumulative sum test	0.489497	0.450	success

5.4. KEY GENERATION RESULT

The fundamental coefficients associated with prime numbers and their features are computed for utilization in cryptographic applications. The variable value from Table 3 is used to determine Ns and Nc values, where X_1 and X_2 represent the prime numbers for the server (P_s and Q_s), and X_3 represents the private key for the server (E_s). X_5 and X_6 represent the prime numbers for the client (P_c and Q_c), and X_4 represents the private key for the client (E_c). $N_s = P_s \times Q_s$ and $N_c = P_c \times Q_c$. The Euler function (φ) corresponding to each of N_s and N_c is computed using the formulas $\varphi(N_s)=(P_s\text{-}1)\ (Q_s\text{-}1)$ and $\varphi(N_c)=(P_c\text{-}1)\ (Q_c\text{-}1)$, which emphasizes the need to develop mathematical models utilized in cryptography. The values of E_s and E_c are evaluated for compatibility with the Euler function using the Greatest Common Divisor (GCD), while the equivalent numbers D_s and D_c are ascertained through iterative calculations to provide the encryption keys. The conclusive results are recorded as fundamental coefficients for the server P_s, Q_s, E_s, and D_s. In contrast, the coefficients for the client are P_c, Q_c, E_c, and D_c for five users, as shown in Table 5.

Table 5. Public and private key results

user	Ps	Qs	Es	Ds	Pc	Qc	Ec	Dc
1	2311	4327	937	3775393	89	11	127	783
2	8059	11	4877	19133	401	4327	4327	592663
3	9041	4973	8647	40372663	1399	11	71	12011
4	557	6113	5147	3373843	1559	4973	4987	5385339
5	1867	9473	2143	7571359	191	6079	6163	455707
6	4073	5783	3163	11485571	331	9473	9587	714683
7	8147	6217	9281	40476785	701	5783	5923	890387
8	8263	2459	8669	5973629	1409	6217	6299	947603
9	4751	8747	2333	1798497	1427	2459	2543	3320411
10	3967	853	6673	2017393	821	8747	8779	3620579

The experiment demonstrated that the model delivers great precision in calculating these fundamental coefficients, rendering it an efficient instrument for creating public and private keys utilized in safe encryption schemes, such as RSA. The execution time is justifiable given the many procedures involved, which improve the program's efficiency and applicability in the real world, as shown in Table 6.

Table 6. Number generation and checking the RSA parameter time

Iteration	number generation time (msec)	Check the RSA parameter time (msec)
1	0.1183	0.0122
2	0.0383	0.0094
3	0.0549	0.0104
4	0.0911	0.0131
5	0.0630	0.0141
6	0.0894	0.0089
7	0.0647	0.0178
9	0.0645	0.0155
10	0.0919	0.0098
11	0.0262	0.0109
12	0.0868	0.0145
Average	**0.0717**	**0.01138**

From the previous table, the Average executing time for the generation of random numbers through 6D Hyperchaotic Systems, followed by testing and parameter generation to satisfy RSA criteria, is 0.08308 msec. The Shannon entropy value measures bit randomness and determines the authentication protocol's statistical attack resistance. A key with a greater entropy value is more difficult to analyze and crack. Table 7 shows the entropy of the key's size.

Table 7 shows similar and high Shannon entropy values, approaching 8, which indicates that they all provide keys with robust resistance to statistical attacks.

Table 7. The entropy of the size of the key

Size of keys (bits)	Entropy
800	7.301877
1024	7.350909
1600	7.394059
2048	7.45548
3200	7.609392
4096	7.417282
5000	7.665345
6000	7.539276
7000	7.649151
8192	7.749044

5.5. AUTHENTICATION USING RSA RESULT

RSA is one of the earliest public-key cryptosystems founded on the so-called "factoring problem". It remains the most extensively utilized system in practice. The verification method involves comparing the original user input ID with the final values obtained after completing all encryption and decryption phases. The procedure commences with the encryption of the original user input ID utilizing the client's private key (E_c, N_c) to generate the value En_1. Subsequently, En_1 is encrypted with the server's public key (D_s, N_s) to yield the value En_2.

$$En_1 = ID^{Ec} \mod N_c$$
$$En_2 = En_1^{Ds} \mod N_s$$

The server decrypts En_2 using its private key (E_s, N_s) to obtain En_3, which is decrypted by using the public key of the client (D_c, N_c) to yield the value En_4.

$$En_3 = En_2^{Es} \mod N_s$$
$$En_4 = En_2^{Dc} \mod N_c$$

After this series, the procedure's accuracy is confirmed by comparing the retrieved value En4 with the original value of the user ID, as shown in Table 8 for six users.

Table 8. Authentication using RSA result

User	ID	En_1	En_2	En_3	En_4	P_s	Q_s	E_s	D_s	P_c	Q_c	E_c	D_c	Time (sec)
1	141	190	276643	190	141	2311	4327	937	3775393	89	11	127	783	0.0708
2	106	14813	26434939	14813	106	9041	4973	8647	40372663	1399	11	71	12011	0.6483
3	132	944659	15799198	944659	132	1867	9473	2143	7571359	191	6079	6163	455707	0.1363
4	104	816440	20802234	816440	104	4073	5783	3163	11485571	331	9473	9587	714683	0.2067
5	114	2037635	19253650	2037635	114	8147	6217	9281	40476785	701	5783	5923	890387	0.6653
6	127	1143780	6118202	1143780	127	8263	2459	8669	5973629	1409	6217	6299	947603	0.1198

If the two values (ID and En_4) are the same, the process is deemed successful, and the user is authorized to proceed; however, if they are not identical, the user is unauthorized to proceed. The last column in Table 8 shows the authentication time for each user. The results demonstrated significant efficacy in maintaining verification accuracy and data integrity, establishing it as a dependable instrument for encryption applications necessitating elevated security and verification standards, including key management in banking and cybersecurity systems.

Table 9. Proposed RSA time

Reference	Method	Length of prime	key Generation Times (msec)
[23]	RSA	64	20.27
		128	25.47
		256	40.50
		512	76.55
		1024	820.14
[23]	CRPKC-Ki	64	35.02
		128	54.19
		256	105.99
		512	177.18
		1024	1250.09
	Proposed	64	16.96
		128	20.53
		256	36.38
		512	64.56
		1024	680.88

Table 9 presents a comparison of key generation times across several key sizes, including 64, 128, 256, 512, and 1024, as well as the encryption and decryption times for the proposed method, RSA [23], and CRPKC-Ki [23]. RSA [23] and CRPKC-Ki [23] are implemented and analyzed in SageMath Jupyter Notebook. SageMath is a Python-based software that offers mathematical computing, data analysis, graph drawing, and programming capabilities on several operating systems. The proposed method is implemented and analyzed using an 11th Gen Intel® Core™ i5-11320H @ 3.20 GHz processor, 16.0 GB RAM (15.8 GB accessible), and Windows 10, 64-bit operating system. By observing the key generation time in Table 9 and Fig. 13, it is obvious that CRPKC-ki has the longest key generation time, while the proposed method has a little different key generation time from RSA. The time is acceptable considering enhanced security.

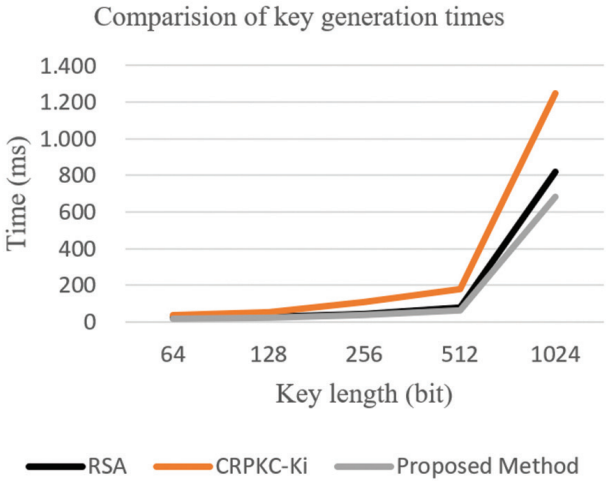


Fig. 13. Key generation times comparison of various key sizes

6. CONCLUSION

In this paper, an attribute-based authentication (ABA) system is presented that integrates biometric fingerprint recognition, the RSA algorithm, and a 6D hyperchaotic system. The proposed method exhibited robust outcomes for security and efficiency, achieving a fingerprint matching accuracy of 95.14% and an average verification time of 0.3078 seconds, so illustrating the system's efficacy in settings necessitating rapid and secure performance Random encryption keys were created using a chaotic system, giving a Shannon entropy value of 7.7490, indicating strong resilience to statistical and brute force attacks. The result using the NIST test showed high randomness, with all tests passing and P values exceeding all thresholds. In comparison to other RSA-based encryption systems like CRPKC-Ki, the proposed system exhibited a reduced key generation

time while preserving a high security level, illustrating that the integration of chaotic systems with conventional encryption algorithms improves performance and robustness against threats.

The recommended method improved randomization statistical performance, and then the client attribute (user ID) is encrypted and decrypted using the RSA algorithm to ensure authentication and protection against various attacks. This approach aims to achieve strong and effective authentication in terms of security and computational efficiency, making it suitable for applications that require high levels of protection, such as sensitive banking and information systems. The security analysis indicates that the algorithm benefits from the security advantages of both 6D hyperchaotic systems and RSA, making it robust against typical RSA cipher attacks.

7. REFERENCES

[1] Z. Leyou, W. Jun, M. Yi, "Secure and Privacy-Preserving Attribute-Based Sharing Framework in Vehicles Ad Hoc Networks", IEEE Access, Vol. 8, 2022, pp. 116781-116795.

[2] M. M. Nayyef, A. M. Sagheer, S. S. Hamad, "Attribute Based Authentication System using Homomorphic Encryption", Journal of Engineering and Applied Sciences, Vol. 13, No. 10, 2018, pp. 352-355.

[3] S. F. Tan, G. C. Chung, "Attribute-Based Encryption in Securing Big Data from Post-Quantum Perspective: A Survey", Cryptography, Vol. 6, No. 3, 2022, pp. 40-55.

[4] X. Liu, W.-B. Lee, Q.-A. Bui, C.-C. Lin, H.-L. Wu, "Biometrics-based RSA cryptosystem for securing real-time communication", Sustainability, Vol. 10, No. 10, 2018, pp. 3588-3603.

[5] A. T. Lo'ai, G. Saldamli, "Reconsidering big data security and privacy in cloud and mobile cloud systems", Journal of King Saud University Computer and Information Sciences, Vol. 33, No. 7, 2021, pp. 810-819.

[6] P. Matta, M. Arora, D. Sharma, "A comparative survey on data encryption Techniques: Big data perspective", Materials today: Proceedings, Vol. 46, 2021, pp. 11035-11039.

[7] R. Ryu, S. Yeom, D. Herbert, J. Dermoudy, "The design and evaluation of adaptive biometric authentication systems: Current status, challenges and future direction", ICT Express, Vol. 9, No. 6, 2023, pp. 1183-1197.

[8] V. Vekariya, M. Joshi, S. Dikshit, "Multi-biometric fusion for enhanced human authentication in information security", Measurement Sensors, Vol. 31, 2024, pp. 100973-100981.

[9] D. Osorio-Roig, L. J. González-Soler, C. Rathgeb, C. Busch, "Privacy-preserving multi-biometric indexing based on frequent binary patterns", IEEE Transaction on Information Forensics and Secuity, Vol. 19, 2024, pp. 4835-4850.

[10] H. Djebli, S. Ait-Aoudia, D. Michelucci, "Quantized random projections of SIFT features for cancelable fingerprints", Multimedia Tools and Applications, Vol. 82, No. 5, 2023, pp. 7917-7937.

[11] S. Bakheet, S. Alsubai, A. Alqahtani, A. Binbusayyis, "Robust fingerprint minutiae extraction and matching based on improved SIFT features", Applied Sciences, Vol. 12, No. 12, 2022, p. 6122.

[12] Z. Zhang, S. Liu, M. Liu, "A multi-task fully deep convolutional neural network for contactless fingerprint minutiae extraction", Pattern Recognition, Vol. 120, 2021, pp. 108189-108201.

[13] S. O. Husain, R. A. Reddy, P. K. Pareek, S. Jagannathan, M. Jyothi, "Robust Fingerprint Minutiae Extraction and Matching Using Fully Connected Deep Convolutional Neural Network and Improved SIFT", Proceeding of the International Conference on Data Science and Network Security, Tiptur, India, 26-27 July 2024, pp. 1-5.

[14] M. Siddiqui, S. Iqbal, B. AlHaqbani, B. AlShammari, T. Khan, I. Razzak, "A Robust Algorithm for Contactless Fingerprint Enhancement and Matching", Proceeding of the International Conference on Digital Image Computing: Techniques and Applications, Perth, Australia, 27-29 November 2024, pp. 214-220.

[15] K. Al-Mannai, E. Bentafat, S. Bakiras, J. Schneider, "Secure Biometric Verification in the Presence of Malicious Adversaries", IEEE Access, Vol. 13, 2024, pp. 5284-5295.

[16] A. H. Y. Mohammed, R. A. Dziyauddin, L. A. Latiff, "Current multi-factor of authentication: Approaches, requirements, attacks and challenges",

International Journal of Advanced Computer Science and Applications, Vol. 14, 2023, No. 1, pp. 166-178.

[17] O. F. Boyraz, E. Guleryuz, A. Akgul, M. Z. Yildiz, H. E. Kiran, J. Ahmad, "A novel security and authentication method for infrared medical image with discrete time chaotic systems", Optik, Vol. 267, 2022, pp. 169717-169735.

[18] Y.-Y. Fanjiang, C.-C. Lee, Y.-T. Du, S.-J. Horng, "Palm vein recognition based on convolutional neural network", Informatica, Vol. 32, No. 4, 2021, pp. 687-708.

[19] I. K. V S. Socheat, T. Wang, "Fingerprint enhancement, minutiae extraction and matching techniques", Journal of Computer and Communications, Vol. 8, No. 5, 2020, pp. 55-74.

[20] B. H. Situmorang, "Identification of Biometrics Using Fingerprint Minutiae Extraction Based on Crossing Number Method", Komputasi Journal Ilmiah Ilmu Komputer dan Mathmatika, Vol. 20, No. 1, 2022, pp. 71-80.

[21] O. Z. Akif, S. Ali, R. S. Ali, A. K. Farhan, "A new pseudorandom bits generator based on a 2D-chaotic system and diffusion property", Bulletin Electrical Engineering and Informatics, Vol. 10, No. 3,2021, pp. 1580-1588.

[22] N. Tahat, A. A. Tahat, M. Abu-Dalu, R. B. Albadarneh, A. E. Abdallah, O. M. Al-Hazaimeh, "A new RSA public key encryption scheme with chaotic maps", International Journal Electrical and Computer Engineering, Vol. 10, No. 2, 2020, pp. 1430-1437.

[23] C. Zhang, Y. Liang, A. Tavares, L. Wang, T. Gomes, S. Pinto, "An Improved Public Key Cryptographic Algorithm Based on Chebyshev Polynomials and RSA", Symmetry, Vol. 16, No. 3, 2024, pp. 263-278.

[24] F. Yu et al. "Chaos-Based Engineering Applications with a 6D Memristive Multistable Hyperchaotic System and a 2D SF-SIMM Hyperchaotic Map", Complexity, Vol. 2021, No. 1, 2021, pp. 6683284-6683305.

[25] B. A. Mezatio, M. T. Motchongom, B. R. W. Tekam, R. Kengne, R. Tchitnga, A. Fomethe, "A novel memristive 6D hyperchaotic autonomous system with hidden extreme multistability", Chaos, Solitons & Fractals, Vol. 120, 2019, pp. 100-115.