Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **7**, Number **1**, February **2022**

Design of synchronized multimode random key generators based on chaos-based peak value coding and its application to secure communication

Jun-Juh Yan*

Department of Electronic Engineering,National
Chin-Yi University of Technology
Taichung 41107, Taiwan
*Corresponding author: jjyan@ncut.edu.tw

Wen-Yuan Chen

Department of Electronic Engineering,National
Chin-Yi University of Technology
Taichung 41107, Taiwan
cwy@ncut.edu.tw

En-Rri Chang

Department of Electronic Engineering,National
Chin-Yi University of Technology
Taichung 41107, Taiwan
andys920605@gmail.com

Received August 2021 Revised October 2021

Abstract. *In this paper, we propose a novel secure communication design based on synchronized multimode random key generators with chaos-based peak value coding (PVC). First, the sliding mode control is introduced to solve the synchronization problem between the master and slave chaotic systems. Then, integrating the randomness and butterfly effect of the chaotic responses with the SHA3-256 (Secure Hash Algorithm 3) algorithm, a dynamic fixed-length random key generator is implemented. Since the peak values of the chaos responses cannot be predicted, the peak coding sequence will be random and unpredictable. Therefore, the quality of random keys can be further promoted. According to the NIST random number test and analysis, the quality of the random keys generated by the multimode structure is better than that of a single chaotic system. Finally, to illustrate the performance and feasibility of this proposed research, we apply synchronized multimode random key generators to a secure communication system for manufacturing machines to ensure the security of information transmission.*
Keywords: Chaotic system; Synchronization; Chaos-based peak value coding(PVC); Secure Hash Algorithm 3; Secure communication

1. **Introduction.** Due to the advancement of information and communication technology, the transmission of data has become very popular and important. Inevitably, the transmitted data will include personal privacy and industrial confidential information. Therefore, how to protect the security of the transmitted data, especially in the big data application, has become a very important issue [1, 2]. To solve this problem, this paper aims to propose a design methodology of novel multimode chaotic random key generators which can provide high quality of random and unpredictable dynamical keys. The

dynamic keys are generated by the combination of the synchronized multimode chaotic dynamic states and SHA3-256 to greatly improve its security and make it difficult to crack. In traditional cryptography, symmetric-key algorithms include DES (Data Encryption Standard), AES (Advanced Encryption Standard), RC5 (Rivest Cipher5), etc.. They all rely on the complexity of encryption algorithms and fixed keys to complete the encryption and decryption of the data [3]. As the increase in computing speed of current computers and the maturity of the quantum computers, quantum computing will easily be realized by using Shor or Grover's quantum computing algorithms [4, 5], and the complex mathematical problems in the traditional cryptography will be easy solved by quantum computers. In other words, all key cryptosystems will be at risk if quantum computers have enough quantum bits in the future [6]. In addition, the key in symmetric encryption also has the storage problem. If the private key is stolen by hackers, the cipher text will be easily cracked. Therefore, in this paper, we propose the design of keys dynamically updated to solve this problem. As well known, chaotic system is a complex nonlinear system that produces a certain non-periodic and random-like motion. Because chaotic motions are dynamic and unpredictable, a chaotic system can generate a large number of random signals in a very short period of time. These chaos signals show random-like behavior and sensitivity to initial values due to the butterfly effect, strange attractors and other characteristics in chaotic systems. Therefore, chaotic systems have been widely applied to data encryption [7-11]. In the past research literature [12, 13], this feature has been successfully applied to symmetric encryption, and the master and slave chaotic systems are established at the transmitter and receiver, respectively, and high-quality dynamically updated keys can be designed to solve the shortcomings of key storage and distribution in the traditional symmetric encryption. However, the encryption and decryption system designed through the chaos random responses must be with the same initial values. But when the state responses are disturbed and resulted in a very small difference, due to the butterfly effect, such design methods will fail. To cope with this problem, it is necessary to rely on chaotic synchronization control technology. Through the synchronization controller, the butterfly effect between the master and slave chaotic systems can be effectively suppressed, and the state trajectories of the master and slave chaotic systems can be forced to synchronize and always generate the same random key to complete the symmetric encryption and decryption process. In [14-18], the researchers of chaotic cryptography have introduced the synchronization controller to improve the stability of encryption systems. But the structure is simple. When the mathematical model and the synchronization controller are identified by intercepting the synchronization signal, the security might be cracked by a brute-force attack. Therefore, in this paper, we aim to propose a multimode chaotic system architecture. Under this design, due to the multimode design and the random effect of the peak code, the seeds for switching selected modes can be dynamically updated to make brute force attacks infeasible. Furthermore, since seeds of the peak coding sequences are random and unpredictable, the quality of random numbers can be also promoted. According to the NIST random number test and analysis, the quality of the random keys generated by the multimode chaotic system is better than that of a single chaotic system.

In order to complete the design of this multimode chaotic random key generator and the secure communication in the machine network, we firstly propose the sliding mode controller to achieve the synchronization of the master-slave chaotic systems. With the pioneering research of Pecora and Carroll [19], many effective control methods have been proposed for chaos synchronization, such as adaptive control, fuzzy control, sliding mode control and H∞ control, etc. Because the sliding mode control method is insensitive to system parameters and external disturbances and has good robustness, this control

approach will be used [20]. In the proposed multimode structure, the synchronized Henon -Map chaotic system [21] and Lorenz-Stenflo chaotic system [22] are implemented. We introduce multiple synchronized Henon -Map systems with different initial conditions and combine with SHA3-256 [23] to generate a dynamic fixed length (256 bits) keys for processing data encryption and decryption. Since the Lorenz-Stenflo system can generate random peaks [24], we will collect unpredictable peaks and design a novel peak value encoding controller (PVCC). By using this PVCC, we realize a multimode chaotic random key generator and then apply it to secure communication. The experimental results illustrate that the PVCC with unpredictable switching time sequence can be successfully realized and applied to the secure communication systems with multimode structure. In order to further verify the randomness quality of the dynamic key generated by our multimode structure, we use the NIST SP 800-22 published by the National Institute of Standards and Technology [25] to test the key sequence. The quantitative score in NIST test are all passed and superior than those only with a single chaotic system. Finally, through the experimental results in the circuit implementation with microcontroller chip, it also shows the success and superiority of this design.

This paper was organized as follows. Section 2 formulated chaos synchronization and the design of dynamic chaos-based random key generator. The synchronization controller was proposed by using discrete sliding mode control. Numerical simulations were given to illustrate the derived results. Section 3 introduced the structure of peak value coding controller (PVCC) and the synchronization design of master-slave PVCCs. The performance of PVCC-based multimode random key generators is analyzed. The secure communication of machinery networks was implemented and verified in Section 4. Finally, conclusions are presented in Section 5.

2. **Design of synchronized dynamic chaos-based key generators.** In order to solve the problem of communication security caused by traditional static fixed keys, this research adopts chaotic synchronization technology to not only prevent key information from being exposed to public transmission channels but also provide dynamic keys for prompting the information security. Due to the chaos synchronization integrated with SHA256 algorithm, it can simultaneously provide the same and high-security dynamic random keys at both transmitter and receiver. To complete the design of chaos synchronization in our multimode architecture, we firstly introduce the master and slave Henon-Map chaotic system to discuss, of course, the technology developed can be extended and applied to different chaotic systems. Equations (1) and (2) are the mathematical models of the master and slave Henon-Maps. Following the design approach in [14], we have

Master hyperchaotic Henon-Map:

$$\begin{aligned} x_1(k+1) &= 1.76 - x_2^2(k) - 0.1x_3(k) \\ x_2(k+1) &= x_1(k) \\ x_3(k+1) &= x_2(k) \end{aligned} \tag{1}$$

Slave hyperchaotic Henon-Map:

$$\begin{aligned} y_1(k+1) &= 1.76 - y_2^2(k) - 0.1y_3(k) + u(k) \\ y_2(k+1) &= y_1(k) \\ y_3(k+1) &= y_2(k) \end{aligned} \tag{2}$$

, where x_i and $y_i, i = 1, 2, 3$ are states of the master and slave Henon-Maps, respectively. The control input $u(k)$ in (2) is the controller designed later to guarantee the synchronization between master and slave systems. To discuss the synchronization controller design,

the error state is defined as

$$e_i(k) = y_i(k) - x_i(k), i = 1, 2, 3 \tag{3}$$

then we have the error dynamics as:

$$\begin{aligned} e_1(k+1) &= x_2^2(k) - y_2^2(k) - 0.1e_3(k) + u(k) \\ e_2(k+1) &= e_1(k) \\ e_3(k+1) &= e_2(k) \end{aligned} \tag{4}$$

To complete the synchronization controller design, we introduce the sliding mode control which includes the following steps **Step1:** we propose the switching function s(k) as

$$s(k) = e_1(k) + c_1e_2(k) + c_2e_3(k) \tag{5}$$

c_1, c_2 are designed parameters. Obviously, if we can design a controller $u(k)$ (the controller will be detailed in Step 2 below) to guarantee that $s(k) = 0$, it yields

$$e_1(k) = -c_1e_2(k) - c_2e_3(k) \tag{6}$$

By substituting $e_1(k) = -c_1e_2(k) - c_2e_3(k)$ into (4), we have

$$\begin{aligned} e_1(k+1) &= x_2^2(k) - y_2^2(k) - 0.1e_3(k) + u(k) \\ E(k+1) &= \begin{bmatrix} e_2(k+1) \\ e_3(k+1) \end{bmatrix} = \begin{bmatrix} -c_1 & -c_2 \\ 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} e_2(k) \\ e_3(k) \end{bmatrix} = AE(k) \end{aligned} \tag{7}$$

From (7), if c_1, c_2 are well selected such that eigenvalues of matrix A will be limited to the identity circle, i.e., i.e,$|\lambda_i(A)| < 1$, then $E(k) = [(e_2(k) \quad e_3(k))]^T$ can converge to zero. Furthermore, $e_1(k)$ will also converge to zero since $s(k) = 0$.

Step2: We continue to design the synchronization controller to guarantee $s(k) = 0$. Form (4) and (5), we have

$$\begin{aligned} & s(k+1) - s(k) \\ &= \underbrace{x_2^2(k) - y_2^2(k) - 0.1e_3(k) + c_1e_1(k) + c_2e_2(k) - e_1(k) - c_1e_2(k) - c_2e_3(k)}_{f(k)} + u(k) \end{aligned} \tag{8}$$

Let

$$u(k) = -f(k) + \alpha s(k) \tag{9}$$

, it yields

$$s(k+1) - s(k) = \alpha s(k) \tag{10}$$

From (10), $s(k+1) = (\alpha+1)s(k)$ can be derived. If we choose an appropriate α such that $|\alpha+1| < 1$, the error dynamic system will smoothly enter the sliding mode with $s(k) = 0$, and from the above Step 1, the errors $e_i(k), i = 1, 2, 3$ can surely converge to zero and the synchronization design is completed.

In the actual application of secure communication, to reduce the exposure of information in the public channel, the designed synchronization controller (9) is decomposed into $u(k) = F(u_m, u_s) = u_m + u_s$, where

$$u_m = x_2^2 + (1 - c1)x_1 + (c_1 - c_2)x_2 + (c_2 + 0.1)x_3 - a(x_1 + c_1x_2 + c_2x_3) \tag{11}$$

$$u_s = -y_2^2 + (c_1 - 1)y_1 + (c_2 - c_1)y_2 - (c_2 + 0.1)y_3 + a(y_1 + c_1y_2 + c_2y_3) \tag{12}$$

Consequently, numerical simulation results are performed to verify the synchronization design. For simulation, the initial conditions of master and slave systems are selected as $x_1 = 1.19, x_2 = 1.03, x_3 = 0.34, y_1 = 0.12, y_2 = 0.11, y_3 = 0.29$ and the parameters are selected as $c_1 = -0.5, c_2 = 0.06, \alpha = -0.5$,then $|\lambda_i(A)| = (0.3, 0.2)$. Obviously, the conditions $|\lambda_i(A)| < 1$ and $|\alpha+1| < 1$ are satisfied and the master and slave Henon-Maps

can be synchronized. Figure 1 shows the dynamic error responses $e_i(k), i = 1, 2, 3$. As expected, under the control of $u(k)$, the system is completely synchronized after $k > 10$.

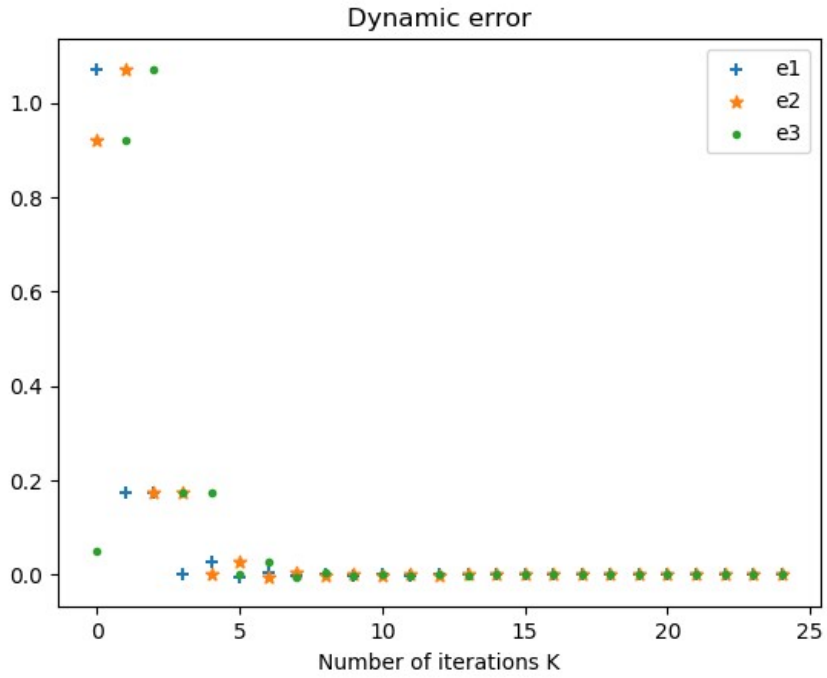

FIGURE 1. The dynamic error responses

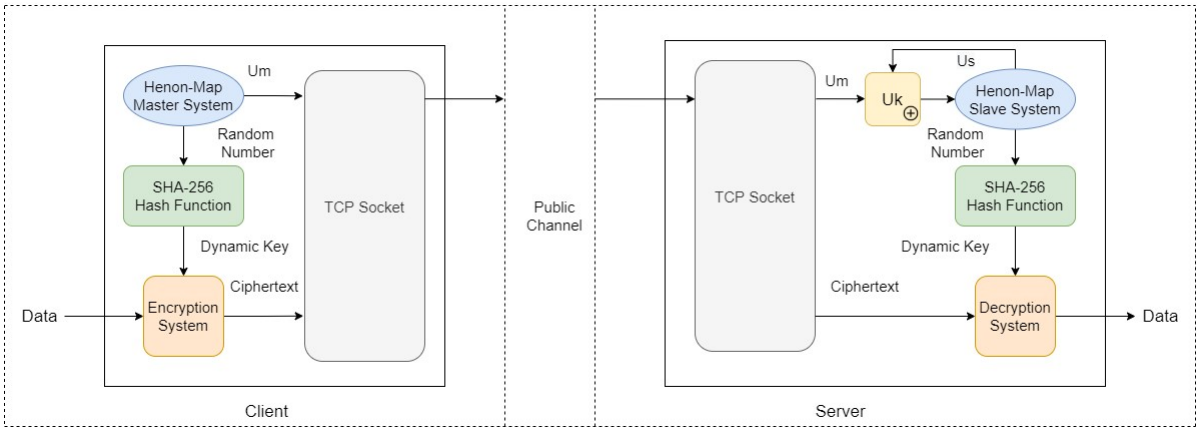

FIGURE 2. Synchronized chaos-based dynamic key generators and data encryption and decryption

Figure 2 describes the design of synchronized chaos-based dynamic key generators and the data encryption and decryption process. From Figure 2, we have placed the master and slave chaotic systems on the client and server, respectively. Both the master and slave Henon-Maps will generate random state responses and u_m, u_s, respectively. Then the server combines the received synchronization signal u_m with the slave synchronization signal u_s to obtain the synchronization controller $u(k)$. Consequently, as discussed above, the master and slave chaotic systems can be forced to synchronize with each other. After synchronization, the synchronized chaotic random responses are inputted to the SHA-256 hash algorithms on the client and server, respectively, as shown in Figure 3. Due

to the butterfly effect of chaos and the avalanche effect of SHA-256 hash algorithms, synchronized random and unpredictable dynamic keys can be obtained simultaneously at the client and server. XOR is used in the encryption and decryption mechanism, as shown in Figure 4. Because of the data transmission in the network environment, we use TCP socket as the transmission protocol. We put the cipher-text and the synchronization signal u_m in the data packet so that the data packet can be exchanged smoothly on the client and server.

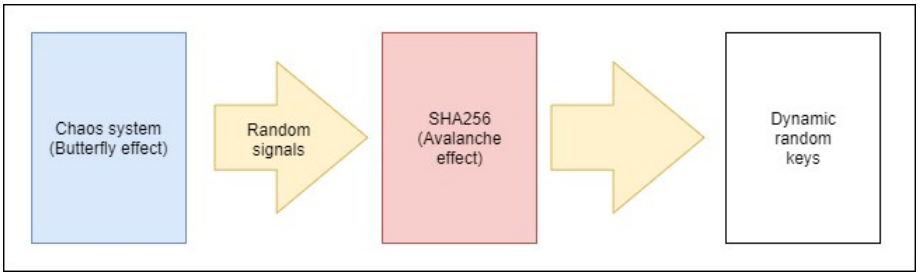

FIGURE 3. The dynamic key generator.

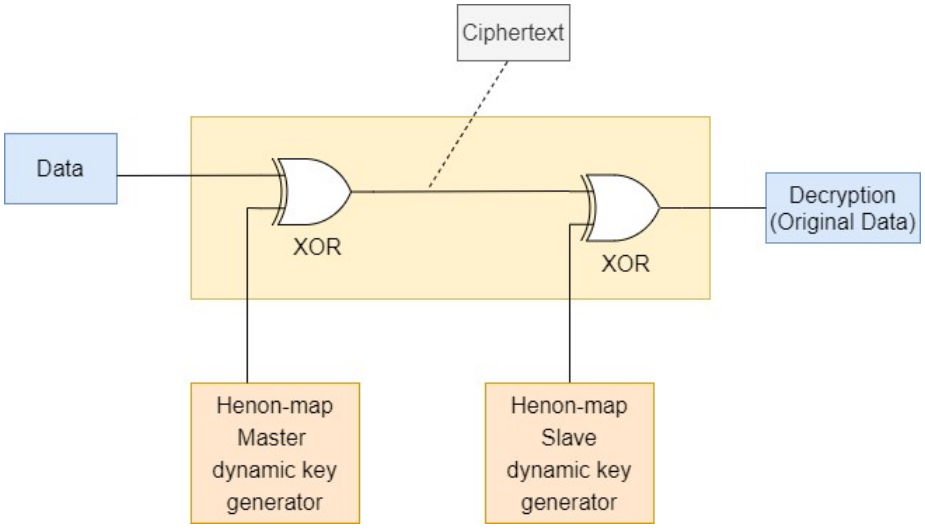

FIGURE 4. The encryption and decryption mechanism.

3. **Design of synchronized multimode random number generators with chaotic peak value coding.** To further improve the randomness quality of the dynamic keys, we propose a multimode architecture as shown in Figure 5. In Figure 5, we build a plurality of chaotic key generators and design a peak value coding controller (PVCC) to randomly switch the dynamic keys generated by different chaotic key generators. And by NIST test analysis, it can conclude the randomness quality of dynamic keys generated with the proposed multimode architecture is greatly improved.

Chaotic peak value coding is based on the random trajectory of chaotic systems. Due to the characteristics of sensitivity to initial values (butterfly effect), it can generate random signals with strange attractors that do not diverge or converge. Therefore, we utilize this random property to design switching rules to achieve unpredictable switching timing and increase the difficulty of cracking. Before formulating the chaotic peak value coding, we introduce the synchronization of the Lorenz-Stenflo chaotic systems for producing the same random peak coding sequences, so that the client and server can have the

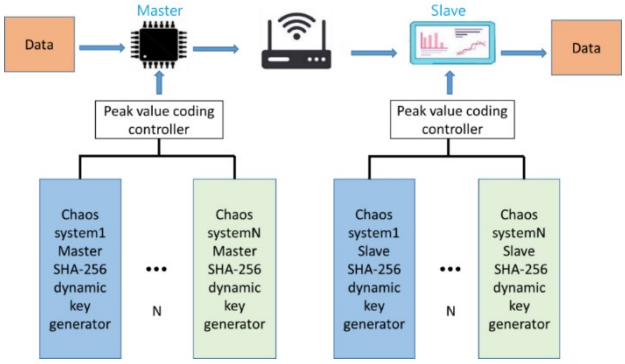

FIGURE 5. Examples of seven expressions

same switching timing to select the corresponding dynamic key generators. The following describes the synchronization controller design of the master-slave Lorenz-Stenflo chaotic systems [22].

Master hyperchaotic Lorenz-Stenflo system:

$$\begin{aligned}
\dot{x}_1(t) &= -ax_1(t) + ax_2(t) + \lambda x_3(t) \\
\dot{x}_2(t) &= -dx_1(t) - x_2(t) - x_1(t)\, x_4(t) \\
\dot{x}_3(t) &= -cx_1(t) - x_3(t) \\
\dot{x}_4(t) &= x_1(t)\, x_2(t) - bx_4(t)
\end{aligned} \tag{13}$$

Slave hyperchaotic Lorenz-Stenflo system:

$$\begin{aligned}
\dot{y}_1(t) &= -ay_1(t) + ay_2(t) + \lambda y_3(t) \\
\dot{y}_2(t) &= -dy_1(t) - y_2(t) - y_1(t)\, y_4(t) \\
\dot{y}_3(t) &= -cy_1(t) - y_3(t) \\
\dot{y}_4(t) &= y_1(t)\, y_2(t) - by_4(t)
\end{aligned} \tag{14}$$

According to [26], the continuous-time 4D LS hyper-chaotic systems (13) and (14) can be discretized as:

$$x_d(k+1) = Gx_d(k) + H \begin{bmatrix} -x_{d1}(k)\, x_{d4}(k) \\ x_{d1}(k)\, x_{d2}(k) \end{bmatrix} \tag{15}$$

$$y_d(k+1) = Gy_d(k) + H \begin{bmatrix} -y_{d1}(k)\, y_{d4}(k) \\ y_{d1}(k)\, y_{d2}(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u(k), \tag{16}$$

where $G \in R^{(4\times4)}$, $H \in R^{(4\times2)}$, $x_d(k) \in R^4$ and $y_d(k) \in R^4$ are the discrete states defined as

$$x_d(t) = \begin{bmatrix} x_{d1}(t) \\ x_{d2}(t) \\ x_{d3}(t) \\ x_{d4}(t) \end{bmatrix}, y_d(t) = \begin{bmatrix} y_{d1}(t) \\ y_{d2}(t) \\ y_{d3}(t) \\ y_{d4}(t) \end{bmatrix} \tag{17}$$

Substituting parameters with $a = 11.0, b = 2.9, c = 5.0, d = 23.0, \lambda = 1.9, T = 0.01$, it yields

$$G = \begin{bmatrix} 0.9071 & 0.1041 & 0.0180 & 0 \\ 0.2176 & 1.0022 & 0.0021 & 0 \\ -0.0473 & -0.0026 & 0.9896 & 0 \\ 0 & 0 & 0 & 0.9714 \end{bmatrix}, H = \begin{bmatrix} 0 & 0 \\ 0.0100 & 0 \\ 0 & 0 \\ 0 & 0.0099 \end{bmatrix} \tag{18}$$

T is the sampling time. Define the error states

$$e_{di}(k) = y_{di}(k) - x_{di}(k), i = 1, 2, 3, 4, \tag{19}$$

we have the error dynamics as

$$e_{d1}(k+1) = 0.9071e_{d1}(k) + 0.1041e_{d2}(k) + 0.0180e_{d3}(k) \tag{20}$$

$$e_{d2}(k+1) = 0.2176e_{d1}(k) + 1.0022e_{d2}(k) + 0.0100(x_{d1}(k)x_{d4}(k) - y_{d1}(k)y_{d4}(k)) + u(k) \tag{21}$$

$$e_{d3}(k+1) = -0.0473e_{d1}(k) + 0.9896e_{d3}(k) \tag{22}$$

$$e_{d4}(k+1) = 0.9714e_{d4}(k) + 0.0099(-x_{d1}(k)x_{d2}(k) + y_{d1}(k)y_{d2}(k)) \tag{23}$$

In (21), the controller $u(k)$ is designed as

$$u(k) = -0.2176e_{d1}(k) - 0.0100(x_{d1}(k)x_{d4}(k) - y_{d1}(k)y_{d4}(k)) - r \times e_d2(k) \tag{24}$$

We obtain

$$e_{d2}(k+1) = (1.0022 - r)e_{d2}(k) \tag{25}$$

Obviously, when r is specified to satisfy the condition of $|(1.0022 - r)| < 1$, the error state $e_{d2}(k)$ will converge to zero. Furthermore, when $e_{d2}(k)$ converges to zero, (20) and (22) can be rewritten in the matrix form as:

$$e_{d1,d3}(k+1) = \begin{bmatrix} 0.9071 & 0.0180 \\ -0.0473 & 0.9896 \end{bmatrix} \begin{bmatrix} e_{d1}(k) \\ e_{d3}(k) \end{bmatrix} = A \begin{bmatrix} e_{d1}(k) \\ e_{d3}(k) \end{bmatrix} \tag{26}$$

By surveying the eigenvalues of matrix A, we have $\lambda_i(A) = (0.9192, 0.9775)$ and it ensures that $e_{d1}(k)$ and $e_{d3}(k)$ will converge to zero.
Finally, when $e_{d1}(k) = e_{d2}(k) = e_{d3}(k) = 0$, according to (23),we have

$$e_{d4}(k+1) = 0.9714e_{d4}(k) \tag{27}$$

(27) means that e_{d4} will also converges to zero.
Based on the above discussion, we get a conclusion that under the control of $u(k)$ in (24), the error states $e_{di}(k), i = 1, 2, 3, 4$ will converge to zero, which means that the master-slave Lorenz-Stenflo chaotic systems (15) (16) can be synchronized. For the security of system realization, we also decompose the controller $u(k)$ (24) into the form of $u(k) = F(u_m, u_s)$, where

$$u_m = 0.2176x_{d1}(k) - 0.0100x_{d1}(k)x_{d4}(k) + r \times x_{d2}(k) \tag{28}$$

$$u_s = -0.2176y_{d1}(k) + 0.0100y_{d1}(k)y_{d4}(k) - r \times y_{d2}(k) \tag{29}$$

For simulation, let $r = 0.9$ and the initial conditions of master and slave systems are selected as $x_{d1} = 18.0, x_{d2} = -1.0, x_{d3} = 6.0, x_{d4} = -2.0, y_{d1} = -1.0, y_{d2} = -3.0, y_{d3} = 2.0$ and $y_{d4} = -3.0$ Figure 6 shows the dynamic error responses $e_{di}(k), i = 1, 2, 3, 4$ As expected, under the control of $u(k)$, the system is completely synchronized.

Chaotic peak value coding design:

After discussing the synchronization of chaotic systems used in the chaotic peak value coding, we now explain the rule of peak value coding. Since discrete Lorenz-Stenflo systems can generate random numbers, we record the peak values ($p_i, i = 1, 2, \ldots, \infty$) of the random signals. Then compare the peak values, if $p_{i+1} > p_i$, the peak coding sequence is stored as 1, otherwise, it is stored as 0. For example, as shown in Figure 7,

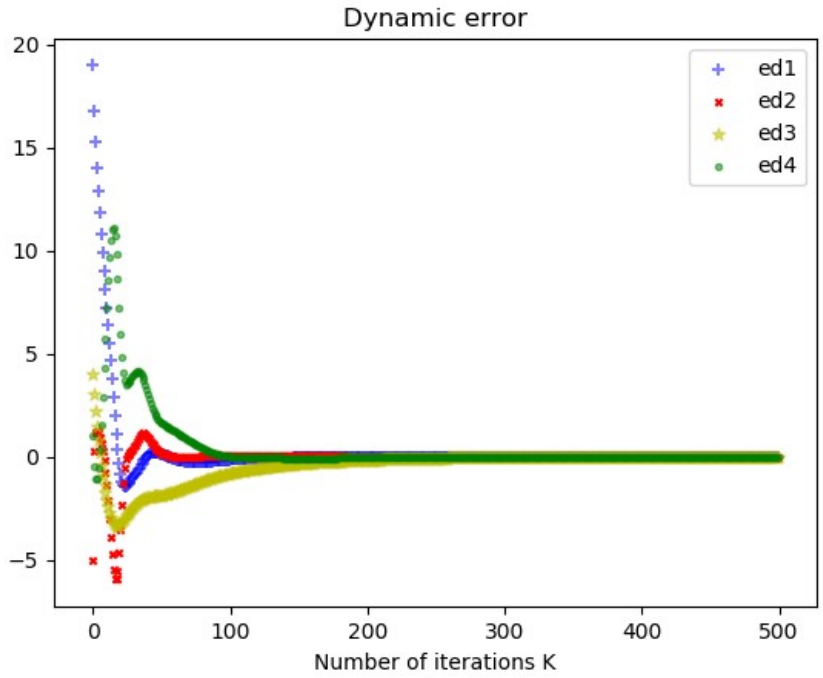

FIGURE 6. Error responses between master and slave discrete Lorenz-Stenflo systems

the peak coding sequence of [1,1,0] will be obtained. Then the collected dynamic and random peak coding sequence can be used to randomly control switching to different chaotic key generators to obtain the corresponding random keys. Figure 8 shows that the identification numbers of each chaotic key generator are in binary order. The length n of the dynamic peak coding sequence satisfies ($N \leq 2^n$) , and N is the number of key generators. Consequently, we explain the design of multimode random key generators based on peak value coding. In Figure 8, we set up N Henon-Map master dynamic key generators (Henon-Map master dynamic keys/ Henon-Map slave dynamic keys). When the Lorenz-Stenflo master-slave chaotic systems are synchronized, the same peak coding sequence will be simultaneously generated at both client and server. Then according to synchronized peak coding sequences, we can design a peak value coding controller (PVCC) to switch and select the corresponding dynamic keys to complete the multimode random key generator design. It is worth mentioning that such a peak coding design ensures that even the designer cannot predict the timing of the switching, so it has better security. Figure 9 shows the random chaotic signals used in the multimode architecture with 4 Henon-Map chaotic systems. The four colors correspond to the four chaotic random number generators. From Figure 9, it is observed if we removed the colors, one cannot distinguish how many chaotic systems are used in the multimode system. Furthermore, due to the dynamic and random switching timing, the security of the communication system will be promoted.

NIST analysis

In this section, we introduce the National Institute of Standards and Technology (NIST) test suite to evaluate the randomness of the dynamic keys with the proposed multimode structure with 4 chaotic random generators. This evaluation standard contains 15 items and the test result for every test item is called p-value. When $p - value > 0.01$, it means that the test item has passed. 7×10^6 bytes generated by single and multimode chaos

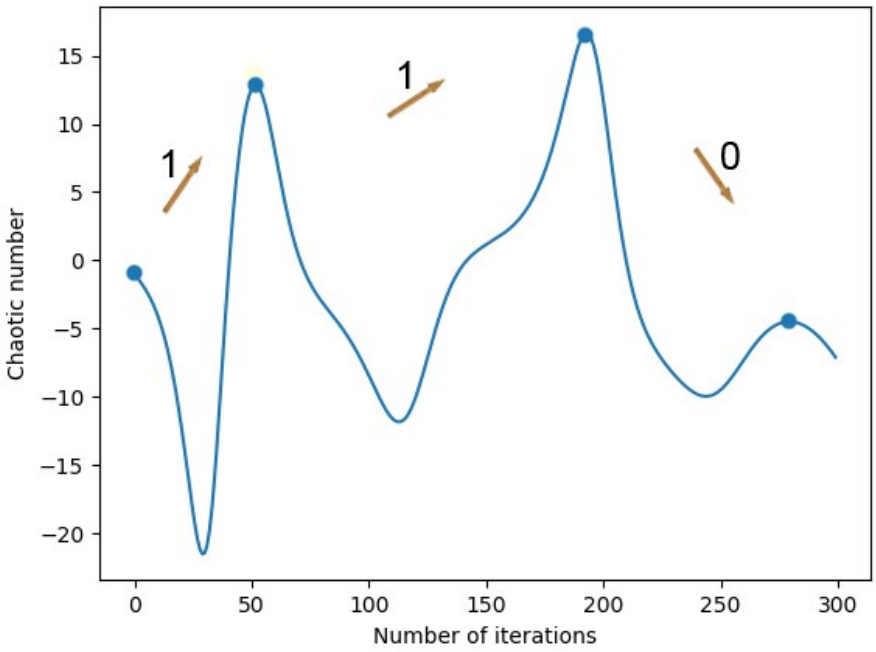

FIGURE 7. Chaotic peak value coding design

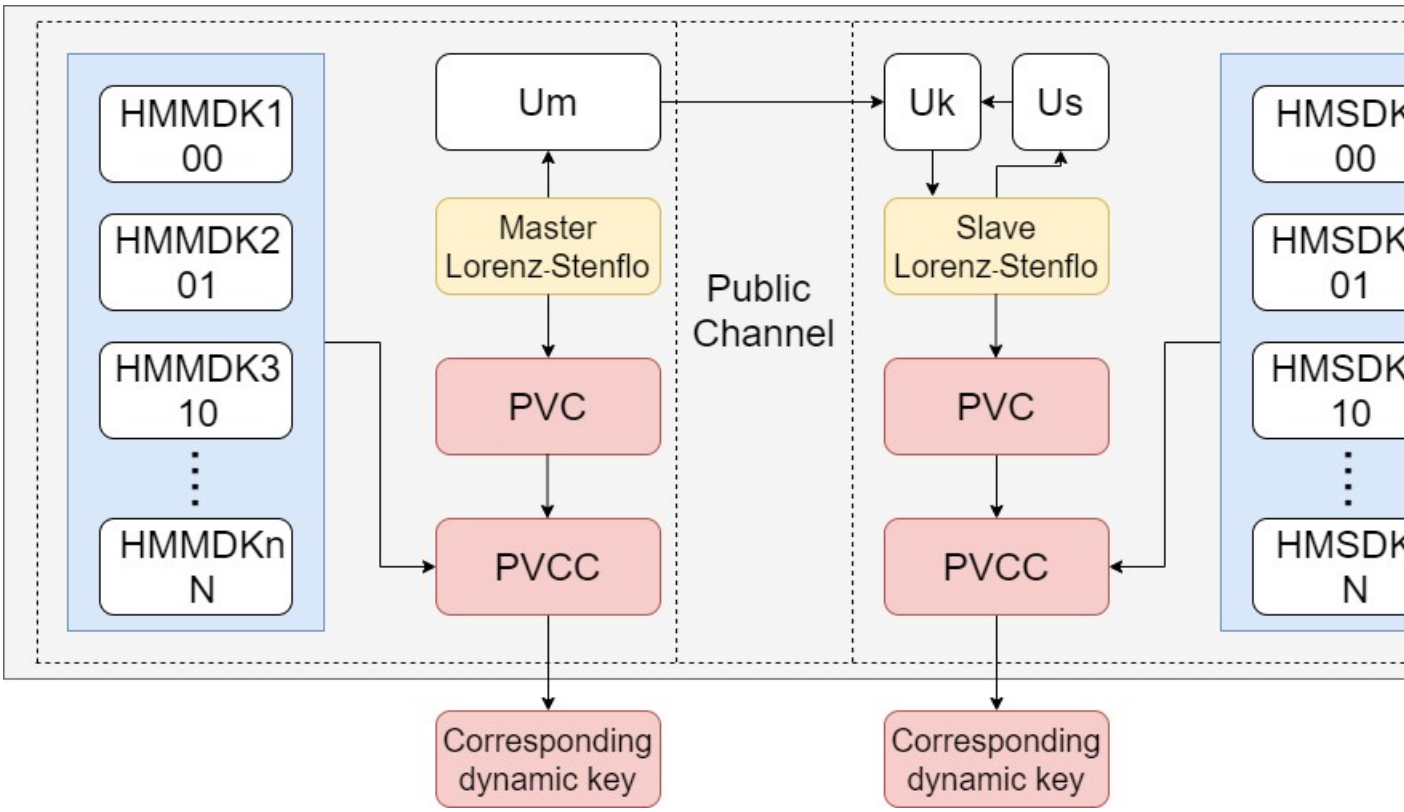

FIGURE 8. The multimode random key generators based on peak value coding.

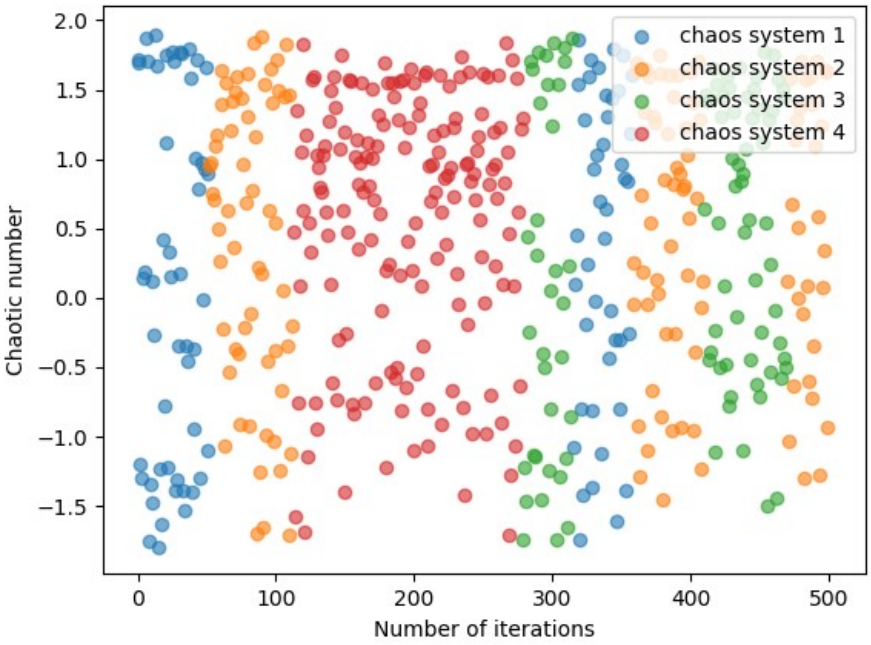

FIGURE 9. Random chaotic signals with the multimode architecture.

systems, respectively, are used and the tested performance results are shown in Table 1. According to the results in Table 1, it shows that the multimode structure is with better randomness performance than that with the single chaos system.

TABLE 1. NIST SP 800-22 test results

	Multimode Chaos System	Single Chaos System
Monobit_test	0.9465934412283477	0.5965516010095926
Frequency_within_block_test	0.8380339828765072	0.9142720506630712
Runs_test	0.983779533775317	0.1420592445387359
Longest_run_one_in_a_block_test	0.9018573433092044	0.46296704966231417
Binary_matrix_rank_test	0.5960553458849351	0.4221540426559427
Dft_test	0.8267653933661077	0.9010184720132587
Non_overlapping_template_matching_test	0.9921944472419604	0.6130991039720712
Overlapping_template_matching_test	0.9447919986387129	0.3773698319636067
Maurers_universal_test	0.679685222869556	0.9676108183616509
Linear_complexity_test	0.5217735995732476	0.7960057806898043
Serial_test	0.3497638056605061	0.4297170308254383
Approximate_entropy_test	0.8007450877593408	0.5731259585933529
Cumulative_sums_test	0.9317843302213593	0.3972832307664227
Random_excursion_test	0.04605987881966095	0.07221188845115749
Random_excursion_variant_test	0.05771562067463309	0.3016383653877924
Total	10.417599031899396	7.967084469554212

4. **Realization of multimode secure communication systems with chaotic peak value coding.** In this section, the proposed multimode architecture based on peak value

coding as shown in Figure 5 will apply to the intelligent machine field for establishing a secure communication system between the tool machines and the server. Figure 10 is the architecture of the proposed secure communication for the machine networks. First, the manufacturing processing data of tool machines are measured and stored by the microcontroller chips. Also in the microcontroller chips, the N master multimode chaos-based SHA256 key generators are built for encryption, and then the tool machine ID, synchronization control signal u_m, and ciphertext are transmitted to our server through a public channel. With the corresponding N master multimode chaos-based SHA256 key generators and the synchronized switching timing in the server, the received ciphertext can be decrypted. For realization and verifying our secure communication system design, we use

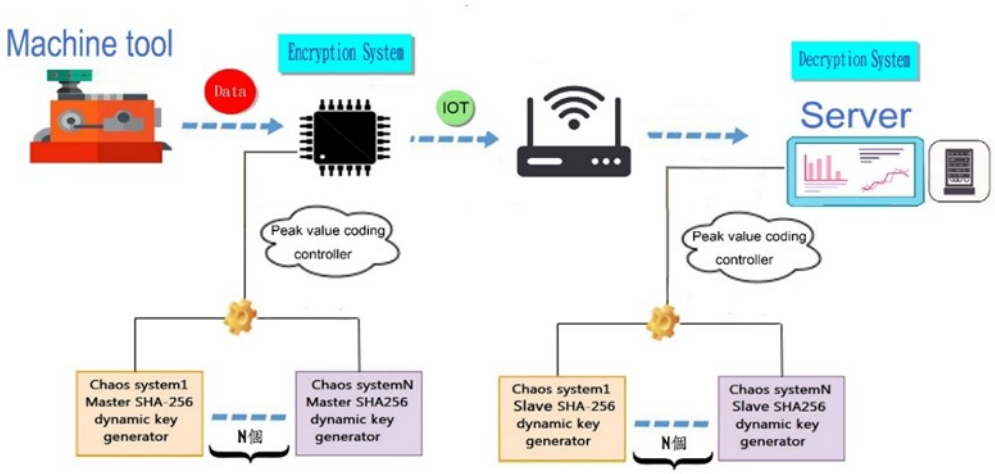

FIGURE 10. The architecture of the proposed secure communication for the machine networks.

two Raspberry Pi MCU to simulate the processing data generated by two machines, respectively, as shown in Figure 11. The data of machine A and B is then encrypted using the dynamic keys generated from the mentioned multimode structure and then retransmitted to the computer (server side) through TCP Socket. Figure 12 shows machine information of tool machine A and B including the switched chaos systems, the peak coding sequence, the dynamic keys, the manufacturing processing data. Figures 13 shows the verification of our realized system on the server. The displayed data includes the switched chaotic system, the peak coding sequence, the dynamic keys, the tool machine parameters, the received cipher-text, and the decrypted plaintext. Next, we explain the meaning of each display data. The switched chaotic system indicates which chaotic random key generator in the multimode architecture is used for the data encryption or decryption. The peak coding sequence describes the sequence collected by the peak coding. As shown in Figure 13, we observe the data information of machine A. the latest two bits of the peak coding sequence is 10 which means the corresponding multimode architecture is using chaotic system 3. When the same chaotic system is selected on the client and server, the same dynamic key can be used for encryption and decryption. The tool machine parameter is the data to be encrypted, which is the plaintext, and the received cipher text with hexadecimal formation is displayed. We can see that received cipher-text is successfully decrypted with the synchronized dynamic key and the manufacturing processing data can be restored back to plaintext as shown in Figure 11.

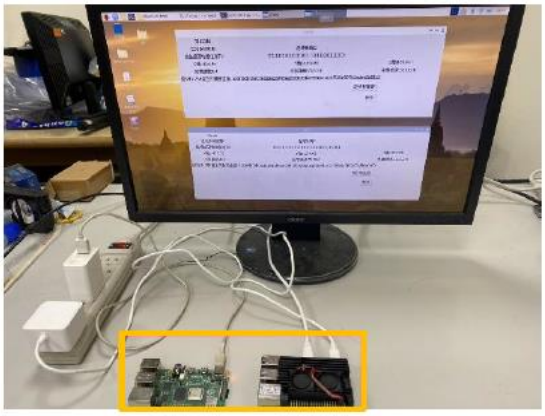

FIGURE 11. Two tool machines simulated with two Raspberry Pis.

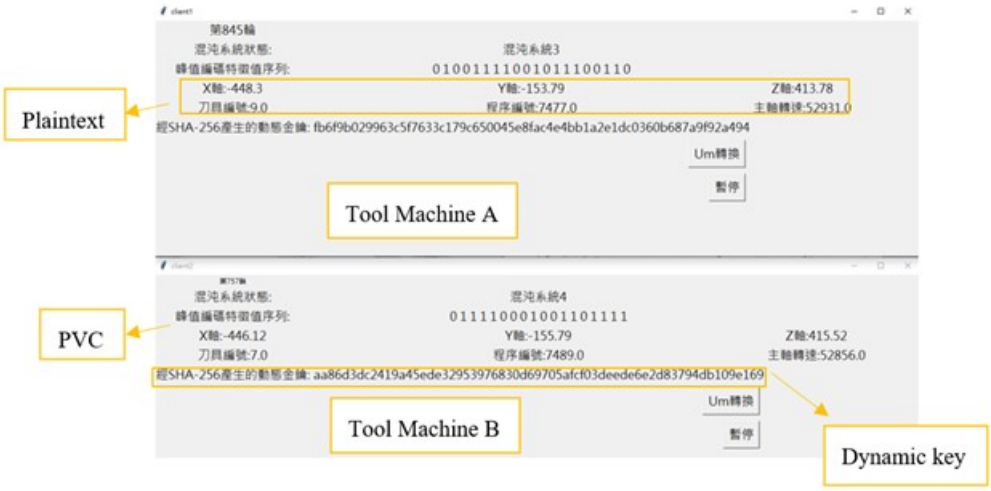

FIGURE 12. The information for tool machines A and B

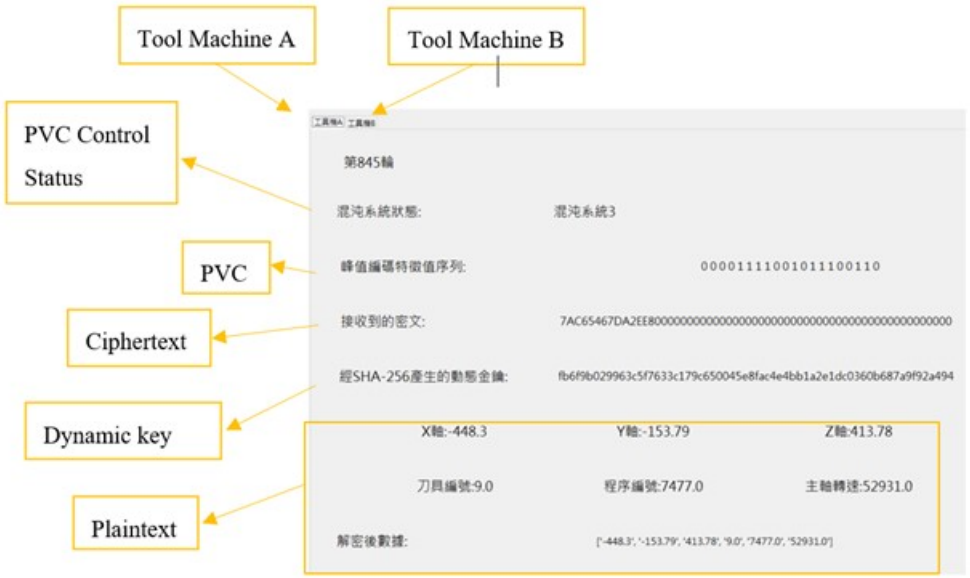

FIGURE 13. The corresponding information on the server

5. **Conclusion.** In this paper, a novel design of synchronized multimode random key generators has been proposed by integrating the chaos random property, chaos synchronization control and the avalanche effect of SHA3-256. An innovative chaotic peak coding sequence is derived to randomly select from the multimode dynamic keys. Such random seed of switching timing reduces the possibility of brute force attacks and spectrum analysis. Through the NIST SP 800-22 test and analysis, it was verified that the randomness of the keys generated by the multimode chaotic system is better than that of a single chaotic system. Finally, we apply this synchronized multi-mode random key generators to realize a secure communication system for two-to-one manufacturing machine system to ensure the security of information transmission between machines and server.

Author Contributions: All authors contributed to this manuscript. C.W. H. wrote the paper with the supervision from C.H.L. and J.J.Y.. G.H.H. was responsible for simulation program design of the robust sliding mode control. E.R.C and W.Y.C. were responsible for circuit realization.

Funding: This work was financially supported by the Ministry of Science and Technology, Taiwan, under MOST- 110-2221-E-167 -030 and MOST-110-2218-E-006 -014 -MBK.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

[1] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, A. Zanella, IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices. *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8182-8201, 2019.

[2] L. Bassi, Industry 4.0: Hope, hype or revolution?, *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*, pp. 1-6, 2017.

[3] R. Saha, G. Geetha, G. Kumar, T. Kim, RK-AES: an improved version of AES using a new key generation process with random keys. *Security and Communication Networks*, vol. 2018, 9802475, 2018.

[4] T. M. Fernández-Caramès and P. Fraga-Lamas, Towards post-quantum blockchain: a review on blockchain cryptography resistant to quantum computing attacks. *IEEE Access*. vol. 8, pp. 21091-21116, 2020.

[5] S. Gupta, K. Sau, J. Pramanick, S. Pyne, R. Ahamed, R. Biswas, Quantum computation of perfect time-eavesdropping in position-based quantum cryptography: quantum computing and eavesdropping over perfect key distribution. *2017 8th Annual Industrial Automation and Electromechanical Engineering Conference*. pp. 162-167, 2017.

[6] V. Mavroeidis, K. Vishi, M. D. Zych, A. Josang, The impact of quantum computing on present cryptography. *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 405-414, 2018.

[7] E. N. Lorenz, Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, vol. 20, pp. 130-141, 1963.

[8] A. A .M. Hany, Implementation of Chaotic Sequences on UWB Wireless Communication in Presence of NBI, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 12, no.2, pp. 83-92, 2021.

[9] C. Volos, I. Kyprianidis, I. Stouboulos, Image encryption process based on chaotic synchronization phenomena. *Journal of Signal Process*. vol. 93, no. 5, pp. 1328-1340, 2013.

[10] E. E. Mahmoud, M. Higazy, T. M. Al-Harthi, A new nine-dimensional chaotic Lorenz system with quaternion variables: complicated dynamics, electronic circuit design, anti-anticipating synchronization, and chaotic masking communication application. *Mathematics*. vol.7, no. 10, 877, 2019.

[11] Q. Wang, S. Yu, C. Li, J. Lu, X. Fang, C. Guyeux, J. Bahi, Theoretical design and FPGA-based implementation of higher-dimensional digital chaotic systems. *IEEE Transactions on Circuits and Systems I*. vol. 63, no. 3, pp. 401-412, 2016.

[12] J. He, J. Cai, Design of a new chaotic system based on Van Der Pol oscillator and its Encryption application. *Mathematics*. vol. 7, no. 8, 743, 2019.

[13] W. Chen, S. Luo, W. X. Zheng, Impulsive synchronization of Reaction–Diffusion neural networks with mixed delays and its application to image encryption. *IEEE Transactions on Neural Networks and Learning Systems*. vol. 27, no. 12, pp. 2696-2710, 2016.

[14] C. H. Lin, G. H. Hu, C. Y. Chan, J. J. Yan, Chaos-based synchronized dynamic keys and their application to image encryption with an improved AES algorithm. *Applied Sciences*. vol. 11, no. 3, 1329, 2021.

[15] S. Bendoukha, S. Abdelmalek, A. Ouannas, Secure communication systems based on the synchronization of chaotic systems. *Mathematics Applied to Engineering Modelling and Social Issues* pp. 281-311, 2019.

[16] C. Nwachioma, M. Ezuma, O.O. Medaiyese, FPGA prototyping of synchronized chaotic map for UAV secure communication. *2021 IEEE Aerospace Conference*. pp. 1-7, 2021.

[17] F. Capligins, A. Litvinenko, A. Aboltins, D. Kolosovs, FPGA implementation and study of synchronization of modified Chua's circuit-based chaotic oscillator for high-speed secure communications. *2020 IEEE 8th Workshop on Advances in Information, Electronic and Electrical Engineering*. pp. 1-6, 2021.

[18] S. Kassim, O. Megherbi, H. Hamiche, S. Djennoune, M. Bettayeb, Speech encryption based on the synchronization of fractional-order chaotic maps. *2019 IEEE International Symposium on Signal Processing and Information Technology*. pp. 1-6, 2019.

[19] L. M. Pecora, T. L. Carroll, Synchronization in chaotic systems. *Physical review letters*. vol.64, no. 8, pp. 821, 1990.

[20] J. J. Yan., C. Y. Chen, J.S.H. Tsai, Hybrid chaos control of continuous unified chaotic systems using discrete rippling sliding mode control. *Nonlinear Analysis: Hybrid Systems*. vol. 22, pp. 276-283, 2016.

[21] D. A. Miller, G. Grassi, A discrete generalized hyperchaotic Henon map circuit. *Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems*. pp. 328-331, 2001.

[22] C.H. Lin, G.H. Hu, J.J. Yan, Chaos suppression in uncertain generalized Lorenz–Stenflo systems via a single rippling controller with Input Nonlinearity. *Mathematics*. vol. 8, 327, 2020.

[23] M. Dworkin, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, Federal Inf. Process. Stds. (NIST FIPS), *National Institute of Standards and Technology, Gaithersburg, MD*, (online), https://doi.org/10.6028/NIST.FIPS.202 (Accessed June 21, 2021)

[24] J. Xavier, P. Rech, Regular and chaotic dynamics of the Lorenz-Stenflo system. *International Journal of Bifurcation and Chaos*. vol. 20, no. 1, pp. 145-152, 2010.

[25] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, S.A.Vo, Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. *2010, NIST Special Publication* 800-22, 2010.

[26] K. D. Young, V. I. Utkin, U. Ozguner, A control engineer's guide to sliding mode control. *IEEE Transactions on Control Systems Technology*. vol.7, no. 3, pp. 328-342, 1999.