Integrated Computer-Aided Engineering 29 (2022) 153–173
DOI 10.3233/ICA-210673
IOS Press

Lightweight encryption for short-range wireless biometric authentication systems in Industry 4.0

Borja Bordel[a], Ramón Alcarria[b,*] and Tomás Robles[a]
[a]*Department of Information Systems, Universidad Politécnica de Madrid, Madrid, Spain*
[b]*Department of Geospatial Engineering, Universidad Politécnica de Madrid, Madrid, Spain*

Abstract. Most recent solutions for users' authentication in Industry 4.0 scenarios are based on unique biological characteristics that are captured from users and recognized using artificial intelligence and machine learning technologies. These biometric applications tend to be computationally heavy, so to monitor users in an unobtrusive manner, sensing and processing modules are physically separated and connected through point-to-point wireless communication technologies. However, in this approach, sensors are very resource constrained, and common cryptographic techniques to protect private users' information while traveling in the radio channel cannot be implemented because their computational cost. Thus, new security solutions for those biometric authentication systems in their short-range wireless communications are needed. Therefore, in this paper, we propose a new cryptographic approach addressing this scenario. The proposed solution employs lightweight operations to create a secure symmetric encryption solution. This cipher includes a pseudo-random number generator based, also, on simple computationally low-cost operations in order to create the secret key. In order to preserve and provide good security properties, the key generation and the encryption processes are fed with a chaotic number sequence obtained through the numerical integration of a new four-order hyperchaotic dynamic. An experimental analysis and a performance evaluation are provided in the experimental section, showing the good behavior of the described solution.

Keywords: Cryptography, chaos, numerical methods, XOR, Industry 4.0, wireless security, privacy, lightweight algorithms

1. Introduction

Industry 4.0 [1] represents an innovative technological approach, where Cyber-Physical Systems [2], Artificial Intelligence [35,72], automatic decision systems [23], optimization solutions [18], exponential technologies [74], robotics [73] and circular business models are building the productive fabric [3,4].

In Industry 4.0, production activities and workplaces are user-centric [5] and personalized [7]. Therefore, Industry 4.0 systems must implement user authentication and identification solutions [8]. However, most traditional mechanisms for user authentication (based on explicit interactions through plastic cards, keyboards and readers) are considered massive, intrusive, and too pollutants to be part of Industry 4.0 solutions. Then, to address this situation, in the last years, authentication technologies based on personal biometric characteristics have been reported [10]. Nevertheless, recognition algorithms tend to be computationally heavy, and large processing servers are needed. If those servers are placed into the user's living environment, their well-being could be reduced. Thus, an edge computing architecture [11] is employed in most recent authentication systems. In that scheme, sensors capturing people information [13] are placed close to final users, and heavy processing algorithms are deployed in large hidden gateways or servers [12]. This solution, on the other hand, opens new problems to be addressed [14]. In par-

*Corresponding author: Ramón Alcarria, Department of Geospatial Engineering, Universidad Politécnica de Madrid, Madrid, Spain. E-mail: ramon.alcarria@upm.es.

ISSN 1069-2509/$35.00

ticular, employed microcontrollers are typically very resource constrained, and they cannot implement common privacy preservation solutions (such as standard cryptographic algorithms) because of their computational cost [15]. Actually, as sensors are not connected to the global Internet, the risks are lower than in other scenarios [16], but biometric information is a very critical and valuable data, and clear communications should not be distributed in a public radio medium. Even in those short ranges, attackers could try to collect data in an unauthorized manner.

Therefore, the objective of this paper is to describe a new lightweight encryption mechanism for short-range wireless communications in Industry 4.0 biometric systems.

Hereinafter, the term "lightweight" refers to mechanisms with a very reduced computational cost or power [6]. In our work, we are looking for an algorithm requiring a very limited computation time. However, Industry 4.0 systems need lightweight algorithms in all senses [9]. Thus, we are also evaluating the memory consumption of the proposed encryption algorithm, looking for a solution with a reduced memory usage.

The proposed solution is focused on enabling a good quality privacy preservation technique, by using only simple binary operations. The cipher includes only lightweight functions, including permutations and rotations; and a XOR gate combining the private information with a secret pseudorandom key. The secret key is obtained through a pseudorandom number generator (PRNG), where only binary operations are employed. In order to guarantee the resulting scheme is strong enough and presents good characteristics in terms of entropy, key space, key sensitivity, etc., the PRNG and the cipher are fed with a chaotic number sequence, generated by numerically integrating a hyperchaotic four-order dynamic. This dynamic introduces important properties such as the sensitivity to the initial conditions. Besides, as an integer dynamic is employed, the computational cost remains acceptable contrary to other approaches based on fractional chaos.

The remainder of the paper is organized as follows. Section 2 presents the state of the art on lightweight encryption mechanisms. Section 3 presents the proposed solution, including all the considered elements and modules. Section 4 describes the experimental evaluation; and Section 5 concludes the paper.

2. State of the art

Proposals about lightweight encryption mechanisms may be classified into two basic groups [17]: on the one hand, cryptographic primitives and, on the other hand, application-specific technologies. Cryptographic primitives are generic algorithms or mathematical functions that can be integrated into different encryption schemes for different scenarios. Application-specific solutions are vertical security technologies specifically designed to adapt the characteristics of certain scenarios.

The following subsections analyze works on each one of these two groups.

2.1. Lightweight cryptographic primitives

Cryptographic primitives may be based on four basic implementation technologies [19]: block ciphers, stream ciphers, hash functions and hardware cryptosystems.

- Block ciphers. A first group of lightweight block ciphers try to improve the performance of DES (Data Encryption Standard) and AES (Advanced Encryption Standard) algorithm through different strategies. A large collection of AES-like lightweight ciphers following a Substitution-Permutation Network (SPN) structure have been reported. From solutions where only small differences are applied, such as AES-128 [20], to more innovative approaches such as KLEIN [21], SKYNNY [24] or LED [22]. Other lightweight encryption mechanisms such as PRINCE [25] or Hummingbird2 [26] introduce larger differences and, even, modify the main core algorithm. Other AES-like lightweight encryption schemes simplify the key management, so the global number of operations and their complexity are also reduced. Solutions such as PRESENT [27], TWINE [28] or mCRYPTON [29] employ keys whose length is greatly below 128 bits of traditional AES. Moreover, although apparently using large keys, some AES-like ciphers such as TEA [30] chop the original key into small subkeys. On the other hand, schemes such as PRIDE [31], RECTANGLE [32] or Neokeon [33] replace traditional complex operations in AES algorithm for simple binary operations which reduce the global computational cost. Finally, although because of the intrinsic insecurity of DES they are less common, some lightweight ciphers based on DES may be also found. DESL [34] is probably the most known.

A second relevant group of lightweight block ciphers are those based on Feistel networks. Feistel functions may be ARX-based (only additions, rotations and XOR operations are allowed) or may be general func-

Table 1
State of the art on cryptographic primitives

	Cipher	Structure	Comments/problems
BLOCK CIPHERS	AES-128	SPN	They are still computationally costly (memory and processing time)
	KLEIN	AES-like	Most of them are not secure anymore
	LED		Problems to operate at real-time or at high-speed in Industry 4.0 scenarios
	SKYNNY		
	PRINCE		
	Hummingbird2		
	PRESENT		
	TWINE		
	mCRYPTON		
	TEA	SPN	
	PRIDE	S-Box	
	RECTANGLE		
	Neokeon		
	DESL		
	SIMON	ARX-based	Poor security performance when implemented alone
	RC5	Feistel	Meet the requirements of Industry 4.0 scenarios
	SPECK	network	Poor key structure
	XTEA		
	KASUMI	General Feis-	
	MISTY	tel	
	RoadRunneR	network	
STREAM CIPHER	TRIVIUM	–	Meet the requirements of Industry 4.0
	MICKEY		Most of them are unsecure
	GRAIN		
HASH	PHOTON	–	Collision probability very high
	QUARK		
	SPOGENT		

tions. Encryption mechanism such as SIMON [39], RC5 [40], SPECK [39] or XTEA [41] belong to the first group. While technologies such as KASUMI [42] or MISTY [43] are based on the second approach. Solutions such as RoadRunneR [44] have been studied in Industry 4.0 scenarios, but they show a poor key structure to guarantee its lightweight properties.

All these previous block encryption schemes, however, present two common problems. First, most of them are not completely secure anymore. And, secondly, current block ciphers still consume large amount of resources. Even solutions with the lowest key length and employing only bitwise operations have reported important memory and processing time consumptions [15].

- Stream ciphers. Lightweight stream ciphers are sparse, and, although they perfectly meet the characteristics of resource-constrained devices (some of them are even focused on these scenarios, such as TRIVIUM [45]), most of them are already broken because of their simple structure and low key length (designed to operate at a very high speed). MICKEY [46] and GRAIN [47] ciphers are examples of this situation.
- Hash functions. Around 2010, there was a great interest to create fast, lightweight hash functions supporting asymmetric encryption schemes in IoT devices. However, reported solutions such as PHOTON [48], QUARK [49] or SPOGENT [50] reduced the output size, so the probability of collision dramatically increased. Thus, most of these schemes are unsecure in typical practical applications.
- Hardware cryptosystems. Hardware supported cryptographic mechanisms have received a lot of attention in the last years. ASIC (application-specific integrated circuit) and FPGA (field-programmable gate array) are employed to build computationally low-cost cryptographic functions [51]. The main practical problem of cryptographic hardware is its sparse flexibility and lack of commercial platforms, what increases the maintenance and replacement costs.

Table 1 summarizes all the information about lightweight cryptographic primitives.

2.2. *Lightweight application-specific security solutions*

For the best of our knowledge, no lightweight encryption technology for Industry 4.0 systems has been reported. Although different lightweight schemes for Industry 4.0 scenarios [52] have been described, the focus of our work is totally different. Currently, lightweight

security technologies are being applied to three basic scenarios: cloud computing systems, Internet of things (IoT) deployments and sensor networks.

- Cloud scenarios. These solutions tend to be focused on reaching computationally fast algorithms, taking profit of the large resources that are available in the cloud. In that way, attribute-based encryption schemes [53] and parametric encryption solutions [38] may be found, where keys are generated by complex procedures from attribute descriptions between both devices communicating [54]. Moreover, as cloud systems are communicating through the global Internet, solutions based on proxies that consume a little amount of private information and re-encrypt and transmit the packets [55], all this in a very fast and efficient manner, have been also reported. On the other hand, specific lightweight encryption technologies for mobile devices accessing to cloud services may be found. Solutions such as the Very Lightweight Proxy Re-Encryption (VL-PRE) [56] or the Lightweight Homomorphic Encryption (LHE) [57] where the key generation process (based on asymmetric mechanisms such as RSA [70]) is improved to reduce its computational cost. Furthermore, lightweight encryption systems whose purpose is reducing the power consumption may be also found [58].
- IoT systems. Lightweight encryption schemes for IoT deployments are typically based on reduced symmetric and/or asymmetric ciphers. Solutions consisting of simple symmetric algorithms where bits are mixed according to a random sequence [15] and simplified Elliptic Curve Cryptography (ECC) mechanisms [59] may be found. The main problem of these proposals is to balance between a high security level and a fast encryption delay. Some hybrid mechanisms have been proposed, combining lightweight symmetric and asymmetric techniques [17,60], but they are hard to adapt to Industry 4.0 scenarios.
- Sensor networks. Schemes for wireless sensor networks [62], Smart Homes [61] and 5G networks [63] may be found. The main drawback of these works is the practical impossibility to adapt application-specific technologies to new scenarios in an efficient manner. Among solutions for sensor networks, there is a group of mechanisms that are very relevant for our works: chaos-based solutions. Chaos-based cryptography employs schemes such as masking [64] or modulation [65] to build solutions, such as watermarking mechanisms [67]. These schemes, however, are vulnerable; and more complex dynamics [66] and fractional chaos [36] have been investigated. Even optical chaos has been employed at physical level [14,68]. Any case, these approaches are still weak as, among other things, transmitter and receptor must be coupled and synchronized [37], and they force the user to send the private key as a stream in a parallel channel, which can be captured by attackers. To address this problem, in our proposal, chaos is not employed as main encryption system, but as a way to increase the entropy and sensibility of the cipher.

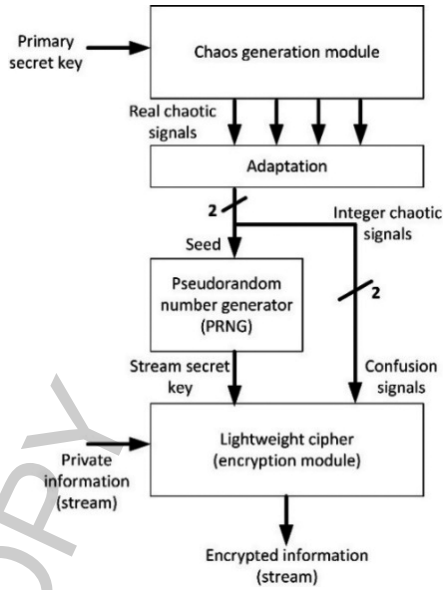


Fig. 1. Scheme for the proposed cryptosystem.

3. Proposed cryptosystem

The proposed cryptosystem (see Fig. 1) includes three basic modules: the chaos generation module, the pseudo-random number generator (PRNG) and the encryption module.

The chaos generation module is software component producing a self-maintained (autonomous, no extra energy input is needed) numerical oscillating trajectory, based on a four-order chaotic dynamic. It generates four different chaotic signals, whose divergence rate may reach very high values (depending on the configuration). These signals are obtained through numerical procedures and take values from the set of real numbers. Thus, these signals must suffer an adaptation process to

transform them into a chaotic sequence of positive integer numbers, as private information and PRNG operate only over the set of natural numbers.

Although chaos presents a sensible and complex behavior, it is a deterministic signal. Then, if directly employed to encrypt private information, statistical analyses and similar techniques could discover the underlying structure. To avoid this problem, a random signal must be employed as key in the cipher. To cover this function, a PRNG fed with a random seed is producing a pseudo-random numerical flow. The proposed PRNG, known as Trifork, only employed bitwise operations to create a high quality pseudo-random signal. The PRNG must be fed with a seed, which is calculated and updated at real-time from the chaotic signals and the previous random number sequences. Thus, the randomness of the final number sequence employed as secret key is even increased, guaranteeing its does not follow any pattern and it cannot be easily replicated by seed variation techniques.

Finally, the secret key from the PRNG and the chaotic signals are introduced in a cipher, which follows a hybrid approach between stream and block ciphers. Private information is divided into small cells, so the proposed encryption schemes may employ block encryption techniques, but macroscopically, it acts as a stream cipher at real time. The proposed cipher integrates only parametric bitwise operations and simple matrix manipulations, but enriched with chaotic signals so the sensibility, confusion and diffusion properties are largely increased compared to previously reported lightweight ciphers. To even increase more the induced confusion and diffusion in the encrypted messages, the cipher implements a feedback loop providing a complex bit mixture. Next subsections are describing all details about each one of these modules.

3.1. Chaos generation module and adaptation module

Traditional chaotic dynamics have been reported to be vulnerable [66]. Systems such as the Lorenz dynamic present structural problems as variables are highly coupled and divergence is limited to only one dimension. Therefore, dynamics with a more complex behavior and higher order are being studied. In this manner, we are considering hyperchaotic dynamics [66], which show a wide catalogue of trajectories as main core of this chaos generation module.

The proposed dynamics is defined by four continuous ordinary differential Eq. (1), where four different continuous real time-dependent variables $\{x(t), y(t), z(t), w(t)\}$ are integrated. Hereinafter, the explicit temporal dependence is omitted. It is important to note that the proposed dynamics only includes simple smooth mathematical operations, contrary to traditional hyperchaotic systems that include complex functions.

$$\begin{aligned} \dot{x} &= d\,(y-x) + 2w \\ \dot{y} &= 5x + cy - 4xz \\ \dot{z} &= xy - 3z \\ \dot{w} &= -bw - a(x-y) \end{aligned} \tag{1}$$

As resource-constrained devices in Industry 4.0 work with a limited word size (usually less than 16 bits), we look for employing only unsigned arithmetic and data formats, so we can take advantage of all bits (no bit for sign is required) and improve the numerical precision in calculations. As a consequence, we are considering a, b, c, d are real positive parameters. Hereinafter they are named as "bifurcation parameters". Three equilibrium points are then observed Eqs (2–4). Being c a positive number these three points always exist.

$$E_0 = (0, 0, 0, 0) \tag{2}$$

$$E_1 = \left(\sqrt{\frac{3}{4}(5+c)}, \sqrt{\frac{3}{4}(5+c)}, \frac{5+c}{4}, 0\right) \tag{3}$$

$$E_2 = \left(-\sqrt{\frac{3}{4}(5+c)}, -\sqrt{\frac{3}{4}(5+c)}, \frac{5+c}{4}, 0\right) \tag{4}$$

In order to determine the parameters space of this dynamics, we must consider that the system must be globally stable Eq. (5), although unstable in some directions of the phase space , so the volume $V\,(t)$ occupied by the trajectory in the phase space must reduce as time passes. This condition Eq. (6) induces a relation between parameters b, c, d which cannot be broken Eq. (7). Hereinafter,

$$\overrightarrow{F} = \begin{pmatrix} d\,(y-x) + 2w \\ 5x + cy - 4xz \\ xy - 3z \\ -bw - a(x-y) \end{pmatrix}$$

represents the dynamics in vector format

$$\frac{1}{V}\left(\frac{dV\,(t)}{dt}\right) < 0 \tag{5}$$

$$\begin{aligned} &\frac{1}{V}\left(\frac{dV\,(t)}{dt}\right) = div\,\left(\overrightarrow{F}\right) \\ &= \left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} + \frac{\partial \dot{z}}{\partial z} + \frac{\partial \dot{w}}{\partial w}\right) = c - (d + 3 + b) \end{aligned} \tag{6}$$

$$d > c - (b + 3) \tag{7}$$

Now, the dynamics may be linearized through the Jacobian matrix Eq. (8), evaluated in a generic equilibrium point $E_g = (x^*, y^*, z^*,\ w^*)$.

$$J\left(E_g\right) = \begin{pmatrix} -d & d & 0 & 2 \\ 5-4z^* & c & -4x^* & 0 \\ y^* & x^* & -3 & 0 \\ -a & a & 0 & -b \end{pmatrix} \tag{8}$$

From this linearized system, three characteristic equations are deducted Eqs (9–11).

$$\begin{aligned} &P_\lambda\left(E_0\right) = \lambda^4 + (b+3+d-c)\,\lambda^3 + \\ &(2a+3b+bd-bc-2d-3c-dc)\,\lambda^2 + \\ &(-3cb-3dc-2db-bcd-15d-2ac-4a)\,\lambda + \\ &(-30a-3dbc-15bd-6ac) = 0 \end{aligned} \tag{9}$$

$$\begin{aligned} &P_\lambda\left(E_1\right) = \lambda^4 + (b+d-c+3)\,\lambda^3 + \\ &(-cb+3d+15+db+2a+3b)\,\lambda^2 + \\ &(3bd+15b+6dc+6a+30d)\,\lambda + \\ &(60a+12ac+6dbc+30db) = 0 \end{aligned} \tag{10}$$

$$\begin{aligned} &P_\lambda\left(E_2\right) = \lambda^4 + (b+d-c+3)\,\lambda^3 + \\ &(-cb+3d+15+db+2a+3b)\,\lambda^2 + \\ &(3bd+15b+6dc+6a+30d)\,\lambda + \\ &(60a+12ac+6dbc+30db) = 0 \end{aligned} \tag{11}$$

A numerical evaluation of the four eigenvalues associated to each characteristic equation shows that, for any valid combination of the bifurcation parameters, there is always, at least, one unstable equilibrium point. This is a very important result, as it guarantees the chaos generation module is generating an oscillating signal for many possible parameter configurations. This guarantees the PRNG seed does not follow a simple pattern and the cipher has a high entropy. Although any other ranges could be selected, in this paper we are assuming the bifurcation parameters vary in specific bounded intervals Eq. (12).

$$\begin{aligned} a &\in [5, 55] \\ b &\in [1, 3] \\ c &\in [2, 4] \\ d &\in [1, 12] \end{aligned} \tag{12}$$

In those regions, the bifurcation diagrams (see Fig. 2) show a large catalogue of structures, including regular and chaotic trajectories. The corresponding chaotic attractors, besides, can be considered self-existent as

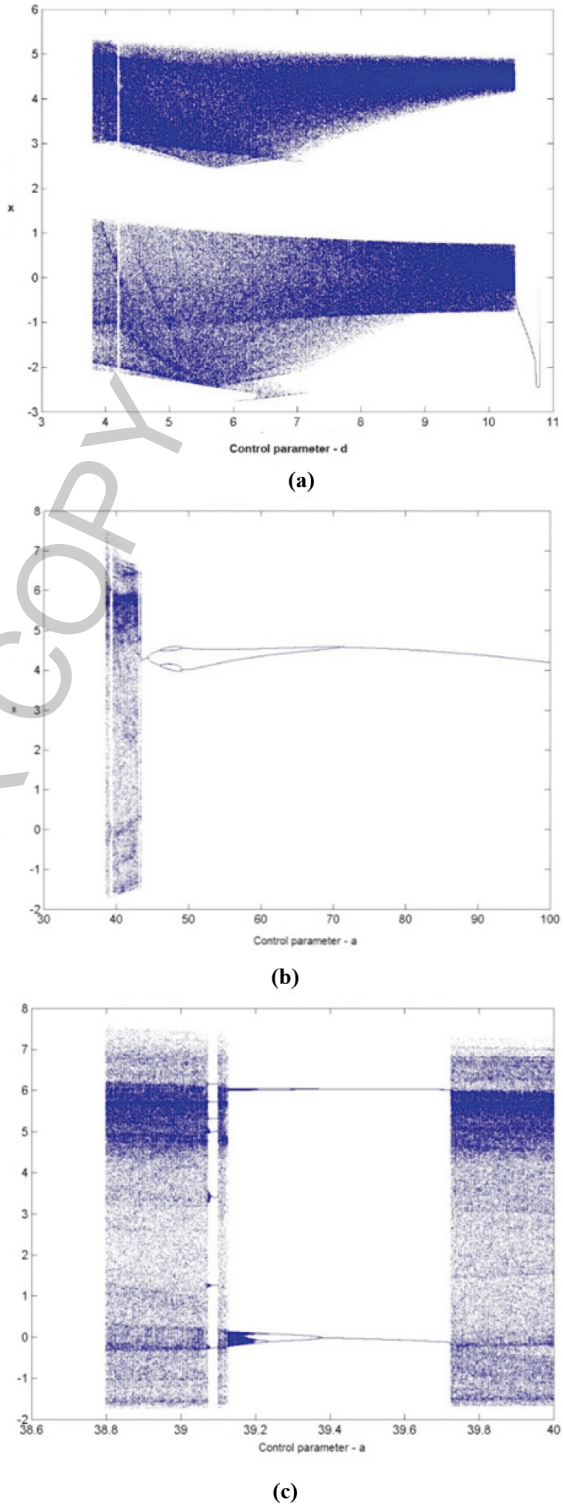
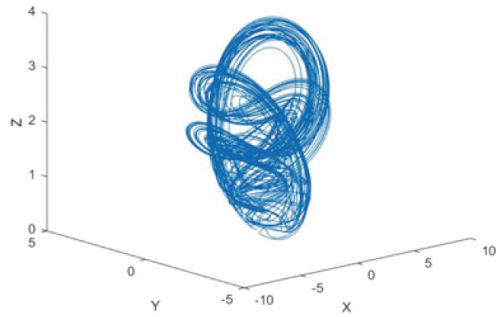
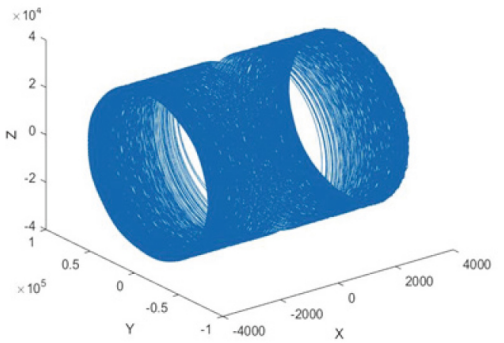


Fig. 2. Bifurcation diagrams. (a) $\{a = 20, b = 2, c = 3\}$ (b–c) $\{b = 2, c = 3, d = 2\}$.



(a)



(b)

Fig. 3. Most representative attractor. (a) $\{a = 40, d = 2, b = 2, c = 3\}$ (b) $\{a = 10, d = 2, b = 2, c = 3\}$.

Table 2
Lyapunov exponents for the proposed chaotic dynamic

Topology $b = 2, c = 3$	Lyapunov exponents	Kaplan-Yorke dimension
$a = 20, d = 7$	$(0.62, 0, -4.7, -4.91)$	2.12
$a = 10, d = 2$	$(12.37, 0, -0.036, -14.71)$	3.83
$a = 40, d = 2$	$(1.11, 0, -2.3, -2.7)$	2.47

their basin of attraction is vast and an equivalent structure is obtained for almost every possible set of initial conditions.

For those ranges of the bifurcation parameters, Lyapunov exponents and Kaplan-York dimension (see Table 2) reach extremely high values, showing the relevant complexity and sensibility to initial conditions and small variation in the bifurcation parameters of the product chaotic signals. Figure 3 shows the most representative attractors in that area.

In order to implement a software chaos generation module, the proposed continuous differential dynamics must be transformed into a sequence of simple mathematical numerical operations (i.e. additions, subtractions, multiplications, etc.). This is especially relevant to meet the requirements of Industry 4.0 resource-constrained devices. To do that, we propose to evaluate the chaotic trajectories using a Runge-Kutta numerical method, which has been proved to solve chaotic differential problems with a good precision and low error.

Although some previous works have successfully integrated chaotic trajectories using four-order numerical methods [14], in order to increase the entropy and security of our cipher and guarantee a good representation and calculation of hyperchaotic trajectories, we propose a more complex scheme. Specifically, we are adapting the Huta's formula [71] to the proposed dynamic. Huta's traditional formula is defined for unidimensional problems, so we are adapting this definition to vectorial functions and trajectories $\overrightarrow{r}(t)$. Besides, Huta considers h the time step. However, the resulting numerical sequence is not a continuous trajectory but a discrete chaotic signal. This discrete signal may be understood as a sampled sequence from the original continuous trajectory, with a sampling period T_s depending on the time step Eq. (13).

$$\begin{aligned} \overrightarrow{r}_n &= \overrightarrow{r}[n] = \overrightarrow{r}(n \cdot T_s) \\ T_s &= h \end{aligned} \tag{13}$$

In those conditions, the Huta's formula is formally identical to a sixth order eight-stage Runge-Kutta method. On the other hand, the Huta numerical method tends to fluctuate between very small and very high numerical values, which is not an adequate behavior for precision-limited devices as they could overflow (in Industry 4.0 many devices show an 8-bit architecture, for example). Then, the operating range of this numerical method may be modified by adapting some coefficients [69] in the original Huta's proposal. The resulting numerical approximation can be directly applied to the dynamics, in order to calculate hyperchaotic trajectories $\overrightarrow{\phi}$ using simple operations Eq. (13).

This Runge-Kutta method defines an initial value problem which requires a four-dimensional vector of initial conditions $\overrightarrow{\phi_0}$ Eq. (15) to be posed, so the problem can be solved. Besides, vector function $\overrightarrow{F}$ has embedded the four bifurcation parameters from the original dynamics. As a result, as the Runge-Kutta method is deterministic, the chaotic trajectories are defined by a set of eight parameters (four initial conditions and four bifurcation parameters). This set is the primary secret key ω_0 Eq. (16) of the proposed cipher.

This primary key starts the encryption process by triggering the generation of the chaotic signals that feed the PRNG and the encryption module. Nevertheless,

this key is not directly involved in the encryption process and, as the chaotic dynamics presents a high entropy, it cannot be deducted from the encrypted messages nor the chaotic signals (see Section 4).

$$\overrightarrow{\phi_{n+1}} = \begin{pmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \\ w_{n+1} \end{pmatrix} = \overrightarrow{\phi_n} + \frac{1}{840}$$
$$\left[41\left(\overrightarrow{k_0}+\overrightarrow{k_7}\right)+216\left(\overrightarrow{k_2}+\overrightarrow{k_6}\right)+\right.$$
$$\left.27\left(\overrightarrow{k_3}+\overrightarrow{k_5}\right)+272\overrightarrow{k_4}\right]$$
$$\overrightarrow{k_0} = h\overrightarrow{F}\left(\overrightarrow{\phi_n}\right)$$
$$\overrightarrow{k_1} = h\overrightarrow{F}\left(\overrightarrow{\phi_n}+\frac{1}{9}\overrightarrow{k_0}\right)$$
$$\overrightarrow{k_2} = h\overrightarrow{F}\left(\overrightarrow{\phi_n}+\frac{1}{24}\left[\overrightarrow{k_0}+3\overrightarrow{k_1}\right]\right)$$
$$\overrightarrow{k_3} = h\overrightarrow{F}\left(\overrightarrow{\phi_n}+\frac{1}{6}\left[\overrightarrow{k_0}-3\overrightarrow{k_1}+4\overrightarrow{k_2}\right]\right)$$
$$\overrightarrow{k_4} = h\overrightarrow{F}\left(\overrightarrow{\phi_n}+\frac{1}{8}\left[-5\overrightarrow{k_0}+27\overrightarrow{k_1}-24\overrightarrow{k_2}+6\right.\right.$$
$$\left.\left.\overrightarrow{k_3}\right]\right) \tag{14}$$
$$\overrightarrow{k_5} = h\overrightarrow{F}\left(\overrightarrow{\phi_n}+\frac{1}{9}\left[221\overrightarrow{k_0}-981\overrightarrow{k_1}+867\overrightarrow{k_2}\right.\right.$$
$$\left.\left.-102\overrightarrow{k_3}+\overrightarrow{k_4}\right]\right)$$
$$\overrightarrow{k_6} = h\overrightarrow{F}\left(\overrightarrow{\phi_n}+\frac{1}{48}\left[-183\overrightarrow{k_0}+678\overrightarrow{k_1}\right.\right.$$
$$\left.\left.-472\overrightarrow{k_2}-66\overrightarrow{k_3}+80\overrightarrow{k_4}+3\overrightarrow{k_5}\right]\right)$$
$$\overrightarrow{k_7} = h\overrightarrow{F}\left(\overrightarrow{\phi_n}+\frac{1}{82}\left[716\overrightarrow{k_0}-2079\overrightarrow{k_1}+\right.\right.$$
$$1002\overrightarrow{k_2}+834\overrightarrow{k_3}-454\overrightarrow{k_4}-9\overrightarrow{k_5}$$
$$\left.\left.+72\overrightarrow{k_6}\right]\right)$$
$$\overrightarrow{\phi_0} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ w_0 \end{pmatrix} \tag{15}$$
$$\Omega_0 = \{x_0, y_0, z_0, w_0, a, b, c, d\} \tag{16}$$

This primary key must be shared between both devices to be communicated. This process can be done using many existing solutions and protocols [14,15]. As this operation will be occasional, its impact in the long-term computational cost will be negligible.

With this approach, four real chaotic signals are generated, in a way that meets the Industry 4.0 characteristics. First, as can be seen Eq. (13), this six-order method only requires eight numerical substitutions to generate a new sample; while a traditional four-order Runge-Kutta method needs twelve. Then, the proposed approach is computationally less complex. But, second, at the same time, it is much more accurate. In fact, the error in this modified Huta's formula can be approximated by the error in a seven-point Newton Cotes formula [69] Eq. (17); while in a standard Runge-Kutta method the error is in the order $o\left(h^4\right)$.

$$\frac{9}{1400}\cdot\left(\frac{h}{6}\right)^9\cdot\overbrace{F}^{\overrightarrow{\cdots\cdots}}\left(\overrightarrow{\phi_n}\right) \tag{17}$$

These real functions, however, are not compatible with integer symbols in PRNG and encryption schemes. Thus, they must be adapted. In other words, data flows taking positive and negative values must be mapped to only take positive values. This operation will be performed through a simple algebraic function f_{v_i} Eq. (18), being s_{min} and s_{max} the minimum and maximum integer numbers accepted in the chaotic signals (may be different for each signal v_i). These values will be selected according to the application scenario. The resulting integer chaotic trajectory $\overrightarrow{\phi^*}$ Eq. (19) will be then injected in the following modules.

$$f_{v_i}(s) = \left\lceil (s - s_{min})\cdot\frac{(s_{max}-s_{min})}{(max\{v_i\}-min\{v_i\})}+s_{min}\right\rceil \tag{18}$$

$$\overrightarrow{\phi^*} = \begin{pmatrix} x^* \\ y^* \\ z^* \\ w^* \end{pmatrix} \tag{19}$$

3.2. Pseudorandom number generator

At this point, the four integer chaotic signals $\overrightarrow{\phi^*}$ are divided into two different groups. Each group ϕ_i^* includes two signals. Although these groups may be freely configured, in this initial proposal we are grouping signals in a sequential manner Eq. (20).

$$\begin{aligned} \Phi_1^* &= \{x^*, y^*\} \\ \Phi_2^* &= \{z^*, w^*\} \end{aligned} \tag{20}$$

Group Φ_2^* is directly introduced into the encryption module (see Section 3.1), while set Φ_1^* is employed to

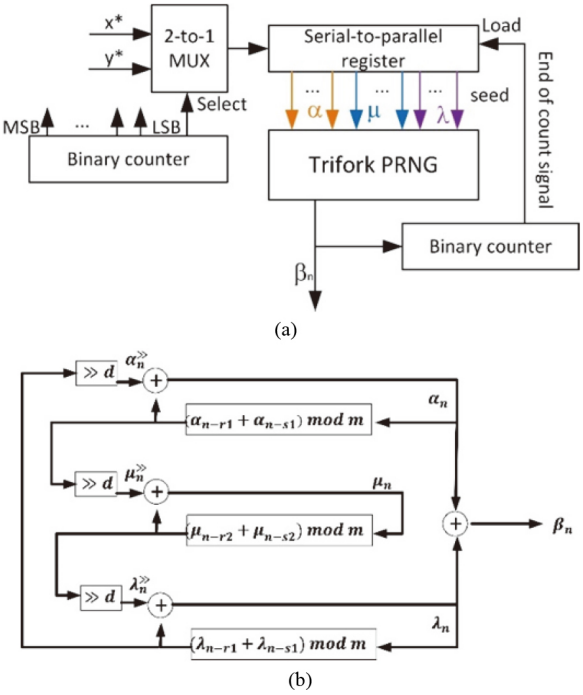


Fig. 4. (a) Proposed scheme for the PRNG (b) internal configuration of Trifork PRNG.

create and periodically refresh the seed of a lightweight pseudo-random number generator (PRNG). Figure 4 shows the proposed scheme. This PRNG will later generate the operational key Φ_1 for the encryption module. Thus, the PRNG must be designed to guarantee the statistical and cryptographic properties required to a robust secret key (see Section 4).

Among all lightweight PRNG one of the most efficient in computational terms is the Linear Feedback Shift Register (LFSR), where k-bit samples $\{_n\}$ from a register bank with R positions are mixed through the exclusive-or addition (XOR, hereinafter). Parameters r and s are integer numbers freely selected, while meeting a simple contour condition Eq. (21). The result is a recursive Eq. (22) generating a sequence of pseudorandom numbers, but where the repetition period is basically dependent on R. Because of the correlations between the values produced as output in LFSR, they can be broken using simple attacks.

$$r, s \geqslant R \tag{21}$$

$$\beta_n = \beta_{n-r} \oplus \beta_{n-s} \tag{22}$$

Generating, then, long pseudo-random sequences needs big registers (and memory space), which are not always available in Industry 4.0. Two basic strategies have been reported in the literature to address this problem. On the one hand, the introduction of additional control parameters produces the Tausworthe generators Eq. (23).

$$\beta_n = \lambda_{Q-1}\beta_{n-1} \oplus \lambda_{Q-2}\beta_{n-1} \oplus \ldots \oplus \lambda_0\beta_{n-Q} \tag{23}$$

This PRNG is a feedback scheme including a set λ_i of Q binary parameters, a number sequence $\{\beta_n\}$ represented as binary variables and a simple XOR operator. On the other hand, the generalization of LFSR, using other arithmetic operations such as modular additions, produces Lagged Fibonacci Generators (LFGs). LFG Eqs (24)–(25) is defined by two integer parameters $r > s > 0$ known as lags, an arithmetical (binary) operation $\circ$, the base for the modular arithmetic m, and a r-dimensional initialization vector (IV) or seed $\{\beta_n,\ n = 0, \ldots, r-1\}$. In Industry 4.0, in order to take advantage, as much as possible, of the controllers' resources, m is chosen as the higher value for an N-bit architecture Eq. (26).

$$LF\,[r, s, m, \circ; \{\beta_n,\ n = 0, \ldots, r-1\}] \tag{24}$$

$$\beta_n = \beta_{n-r} \circ \beta_{n-s} \quad n \geqslant r \tag{25}$$

$$m = 2^N \tag{26}$$

Tausworthe generators produce longer sequences but, however, may cause microcontrollers overflow because of natural multiplications. On the contrary, LFGs perfectly meet the requirements of Industry 4.0 devices, but sequences present poorer cryptographic properties [15]. Then, in this work, we propose a hybrid technique, where control parameters and modular arithmetic are employed at the same time. To solve the overflow risk, multiplications are replaced by binary left-shift and right-shift operations (from a numerical perspective the result is quite similar), which are lighter and cannot overflow the controller. To improve the cryptographic properties of LFG, the calculation procedure Eq. (25) is complicated by adding additional variables to increase the entropy of the final secret key.

To do that, we are representing the LFG as a trinomial Eq. (27) over the Galois Field of two elements, $GF(2)$. Through this mathematical formalization, we can easily calculate the number of samples p (period) that are truly random before the PRNG is captured by a periodical sequence Eq. (28).

$$x^r + x^s + 1 \tag{27}$$

$$p = 2^{N-1}\,(2^r - 1) \tag{28}$$

However, although the output sequence only repeats after p samples, the probability of each one of these

samples may not be uniform, and the PRNG could be attacked using statistical techniques. Actually, in traditional LFG not every output bit has the same behavior, so some samples are more probable than others. Many output samples are only different in the most significant bit (MSB), whose period is equal to the global LFG period Eq. (29), while the remaining bits are already captured in a cyclic behavior (so most samples in the output sequence can take, at the end, only two different values, according to the MSB). In general, the $k-th$ bit shows a much smaller period Eq. (30), so the least significant bit (LSB) has a quite short period Eq. (31).

$$p_{MSB} = 2^{N-1}\left(2^r - 1\right) \tag{29}$$

$$p_k = 2^{k-1}\left(2^r - 1\right) \tag{30}$$

$$p_{LSB} = \left(2^r - 1\right) \tag{31}$$

This standard LFG configuration is weak against modern statistical attacks. Then, a very efficient manner to increase entropy in LFG is to perturbate LSB in the samples to increase its period and make the probability of samples more uniform. These perturbations may take the form of any arithmetic operation, but if additional internal samples are obtained, the computational cost of the global PRNG will go up. To avoid this problem, perturbations are introduced by manipulating the bits inside each sample. The resulting scheme Eq. (32) is known as Perturbed Lagged Fibonacci Generators (PLFGs), where d is an integer control parameter to introduce the desired perturbations.

$$LF\left[r, s, m, d, \circ; \{\beta_n,\ n = 0, \ldots, r-1\}\right] \tag{32}$$

In this case, in order to reduce the PRNG computational cost, we are putting together the multiplications introduced by Tausworthe generators and perturbation from PLFG Eq. (33). The result is a lightweight PRNG where n represents time, d is a constant integer Eq. (34) selected to guarantee the stability of the whole PRNG (specially in limited-precision devices) and symbols $\ll$ and $\gg$ represent the left-shift and right-shift operations.

$$\beta_n = \left(\left(\beta_{n-r} \oplus \beta_{n-s}^{\gg}\right) + \left(\beta_{n-s} \oplus \beta_{n-r}^{\ll}\right)\right) \; mod\, m$$

$$\beta_{n-s}^{\gg} = \left(\beta_{n-s} \gg d\right)$$

$$\beta_{n-r}^{\ll} = \left(\beta_{n-r} \ll d\right) \tag{33}$$

$$2 \leqslant d \leqslant 0.7N \tag{34}$$

This expression Eq. (33) perturbates LSB in every sample, while (at the same time) they represent an arithmetic multiplication inherited from Tausworthe generators Eq. (35).

$$\left(\beta_{n-s} \gg d\right) = \left\lfloor \frac{\beta_{n-s}}{2^d} \right\rfloor$$

$$\left(\beta_{n-r} \ll d\right) = \left(\beta_{n-r} \cdot 2^d\right) mod\, m \tag{35}$$

At this point, the proposed PRNG Eq. (33) includes three important innovations that adapt this technology to Industry 4.0 scenarios: $m - mod$ arithmetical additions increasing the complexity of normal PLFG but avoiding overflow problems; bit-shift (left-shift and right shift) introducing perturbations to the LSB at the same time they represent arithmetical multiplications, two XOR binary operations that increase the period length. This scheme improves dramatically the randomness and repetition period of the PRNG. However, Industry 4.0 devices may be operating continuously for months, so even this improved scheme may not be enough, and even more advanced solutions are needed.

As a response, we create complex PRNG by interconnecting (as branches) different PLFG Eq. (33). Specifically, in this work we are using a three-branch scheme, named Trifork (see Fig. 4). The three branches are connected as follows: the final global samples are obtained through a XOR operation applied to branches, the branches will be totally hidden for external users and the final number of samples depends on several parameters, so statistical attacks cannot infer neither the internal system parameters, the current or past system state nor the secret keys. With this approach, Trifork PRNG Eq. (36) produces random sequences much longer than the ones obtained from conventional PLFG, but with a lower number of operations (and computational cost) than three independent PLFG or one complex PLFG including all operation in only one expression.

Experimentally, it has been proved that values around $d = N/2$ generate sequences with higher randomness.

$$\alpha_n = \left(\left(\alpha_{n-r1} + \alpha_{n-s1}\right)\ mod\, m\right) \oplus \lambda_n^{\gg}$$

$$\mu_n = \left(\left(\mu_{n-r2} + \mu_{n-s2}\right)\ mod\, m\right) \oplus \alpha_n^{\gg}$$

$$\lambda_n = \left(\left(\lambda_{n-r3} + \lambda_{n-s3}\right)\ mod\, m\right) \oplus \mu_n^{\gg}$$

$$\alpha_n^{\gg} = \left(\left(\alpha_{n-r1} + \alpha_{n-s1}\right)\ mod\, m\right) \gg d$$

$$\mu_n^{\gg} = \left(\left(\mu_{n-r2} + \mu_{n-s2}\right)\ mod\, m\right) \gg d$$

$$\lambda_n^{\gg} = \left(\left(\lambda_{n-r3} + \lambda_{n-s3}\right)\ mod\, m\right) \gg d$$

$$\beta_n = \alpha_n \oplus \lambda_n \tag{36}$$

Despite all the previous designs, Trifork needs to be initialized at random, and the randomness of the seed is a key factor conditioning the final behavior of the entire PRNG. Therefore, the seed is not introduced

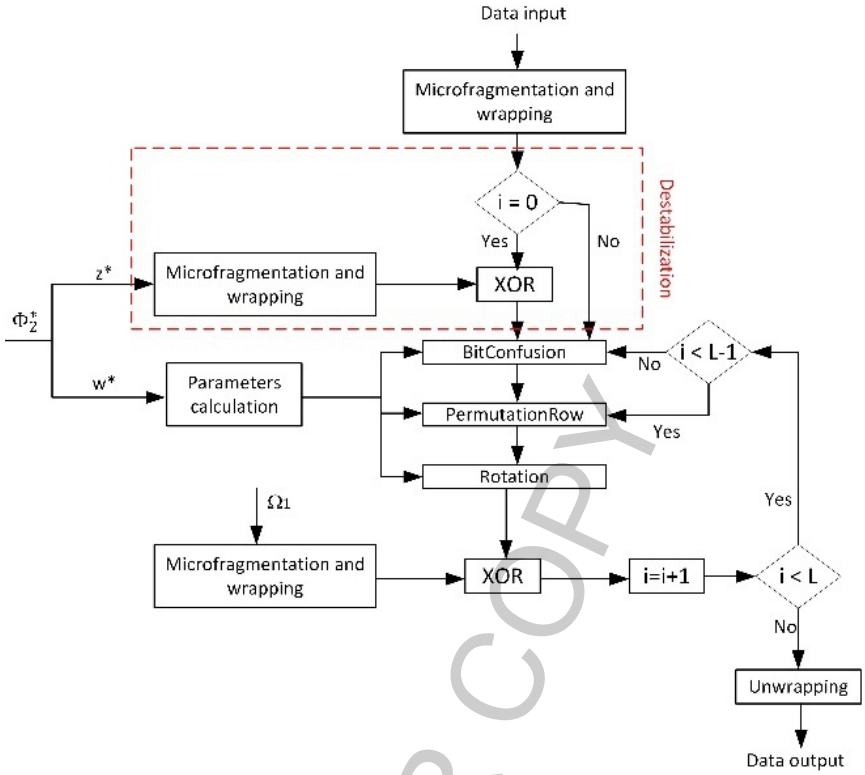


Fig. 5. Proposed encryption module.

by users, but obtained from chaotic signals Φ_1^*. These chaotic signals, although not totally random, have been proved to present a very high entropy [15] (see Section 3.1), enough to guarantee a good performance of the PRNG [58].

As can be seen in Fig. 4, chaotic signals are introduced in a serial-to-parallel register, mixing samples in an alternative way from both signals in Φ_1^* Eq. (37).

$$seed\,[i] = \begin{cases} x^*\,[i] \text{ if } i \text{ is even} \\ y^*\,[i] \text{ if } i \text{ is odd} \end{cases}$$
$$i = 1, \ldots, max\,\{r_1, s_1\} + max\,\{r_2, s_2\} + max\,\{r_3, s_3\} \quad (37)$$

This alternance is controlled through a simple 2-to-1 multiplexer and a cyclic binary counter (with any length), where the LSB is employed to control the multiplexer.

As said before, Industry 4.0 devices may be operating for very long periods, and even complex PRNG such as Trifork may show a periodical behavior if infinite time is considered. To avoid that problem, the proposed PRNG includes a reset mechanism. The Trifork output is monitored by a cyclic counter, so each time a new sample is generated the count is increased. After reaching the maximum value, the seed is refreshed with the current content in the serial-to-parallel register. This seed will be totally different from the previous one, thanks to the chaotic signals, and (then) the random number sequence will start again, avoiding periodical behaviors as much as possible.

The final random sequence of integer numbers is employed as operational secret key at the encryption module, where private biometric information is finally protected.

3.3. Encryption module

Once the operational secret key Ω_1 is generated, the encryption scheme may work. This module will receive three different flows: the operation secret key Ω_1 and both chaotic signals in Φ_2^*. The proposed scheme will operate with data cells, so the encryption scheme may take advantage of stronger block encryption techniques, while the manipulation of small data cells (macroscopically) produces the same performance and user experience than stream ciphers. Figure 5 shows the proposed encryption module.

The proposed encryption module presents the following characteristics:

- The encryption process and the decryption process are identical. Thus, the operational secret key Ω_1 is symmetric.
- The proposed scheme is flexible, and a variable number of rounds may be considered. In this article we are describing two different approaches, but any other may be also valid.
- Input data, in this work, are considered to be already serialized and packed to be sent through the communication system. However, other information types may also be considered, if just the adequate microfragmentation and wrapping function is included.
- The cipher is byte oriented, to match the internal structure of the encryption module and the PRNG.

The raw input data stream $\mathfrak{I}_{raw}$ is introduced in a microfragmentation and wrapping module. In this module, the stream is divided into 64-byte microcells, which are wrapped to create 8×8 bytes matrices. A simple chopping sliding window *win[n]* Eq. (38) with length 64 bytes may be employed to create the microcells $\mathfrak{I}^n_{cell}$ Eq. (39).

$$win\,[n] = \begin{cases} 1 \text{ if } 0 \leqslant n < 63 \\ 0 \text{ otherwise} \end{cases} \tag{38}$$

$$\mathfrak{I}^k_{cell} = \mathfrak{I}_{raw}[n] \cdot win\,[n + 64k] \tag{39}$$

These microcells are then wrapped following a zigzag scheme Eq. (40) to create matrices $\mathfrak{I}_{matrix}$ which may be manipulated as a block.

$$\mathfrak{I}_{matrix} = \left(i^{matrix}_{j,k}\right)$$
$$i^{matrix}_{j,k} = \mathfrak{I}^n_{cell}[k - 1 + 8(j - 1)] \tag{40}$$

One of the key problems in block cipher is the plaintext attack. In biometric authentication solutions this is especially critical. Exposing the biometric system to a certain particular input, we can force the cipher to work with critical inputs (for example, a null matrix). In that case, the system may expose the internal structure or the private key. To avoid this problem, in the first encryption round, the input information is introduced in a destabilization phase.

In this phase, chaotic signal z^* in Φ^*_2 is chopped and wrapped identically as information data $\mathfrak{I}_{raw}$. The resulting 8×8 byte matrix z^*_{matrix} is combined through a XOR addition with $\mathfrak{I}_{matrix}$ matrix Eq. (41). The obtained cell $\mathfrak{I}_{des}$ is guaranteed to have an acceptable entropy, similar to a regular real biometric information. In that way we avoid critical situation and reduce the probability of a successful plaintext attack.

$$\mathfrak{I}_{des} = \mathfrak{I}_{matrix} \text{ XOR } z^*_{matrix} \tag{41}$$

Then, the second chaotic signal w^* is processed to obtain a set of parameters controlling three different functions: the bitConfusion function $bc\,(\cdot)$, the permutationRow function $pr\,(\cdot)$ and the rotation function $ro\,(\cdot)$.

The bitConfusion function $bc\,(\cdot)$ employs a number sequence to reorganize the matrix $\mathfrak{I}_{des}$ at bit level. In order to guarantee bit-oriented operations are performed in an efficient way, the programming language must be carefully selected. Low-level languages, even assembler languages, are, in general, the most suitable option. Otherwise, the efficiency at software level could be compromised. $\mathfrak{I}_{des}$ is a 64×64 bit matrix. Through the function $bc\,(\cdot)$ this matrix is mapped into a new matrix $\mathfrak{I}_{con}$ Eq. (43) according to a set of 4096 different parameter pairs (σ_k, σ_j). These parameters are obtained from the chaotic signal w^* using 64-module arithmetic Eq. (43). If the parameter calculation procedure generates two identical pairs, a new additional pair is obtained until the 4096 different pairs are computed. Data structures enabling an efficient comparison of vectors, such as hash tables, should be employed to achieve a lightweight implementation at software level. This mapping and bit-oriented confusion function is computationally heavier than other considered byte-oriented operations. Therefore, it is only performed twice per cell: in the first and in the last rounds.

$$\mathfrak{I}_{con} = \left(i^{con}_{k,j}\right) = bc\,(\mathfrak{I}_{des}) = bc\left(\left(i^{des}_{k,j}\right)\right) \tag{42}$$

being

$$i^{des}_{k,j} = i^{con}_{\sigma_k,\sigma_j}$$
$$(\sigma_k, \sigma_j) = \left(w^*_k,\ w^*_j\right)\ mod\ 64 \tag{43}$$

The permutationRow function $pr\,(\cdot)$ exchanges two rows in matrix $\mathfrak{I}_{con}$ at byte level. Two different parameters π_1 and π_2 in the range $[1, 8]$ are obtained Eq. (44) from signal w^* using 8-module arithmetic. If two identical numbers are obtained (non-valid result); the parameter calculation process is repeated until two valid number are produced. Again, efficient data structures must be employed to perform this vector comparison step. Then, the π_1-th and the π_2-th rows are permuted Eq. (45) resulting the matrix $\mathfrak{I}_{per}$.

$$\pi_i = w^*_i\ mod\ 8\ + 1 \tag{44}$$

$$\mathfrak{I}_{per} = (row^{per}_k) = pr\,(\mathfrak{I}_{con})$$
$$= pr\,((row^{con}_k))$$

being

$$row^{per}_{\pi_1} = row^{con}_{\pi_2}$$
$$row^{per}_{\pi_2} = row^{con}_{\pi_1} \tag{45}$$

Finally, the rotation function $ro(\cdot)$ moves bytes in matrix $\mathfrak{I}_{per}$ in a cyclic counterclockwise manner. This function receives a parameter representing the turn amplitude: 0°, 90°, 180° or 270°. This parameter is obtained from signal w^* using 4-module arithmetic Eq. (46). The rotated matrix $\mathfrak{I}_{rot}$ Eq. (47) guarantees any minimal change in any byte is later propagated to any other byte in the encrypted data.

$$_i = w_i^* \; mod \; 4 \tag{46}$$

$$\mathfrak{I}_{rot} = \left(i_{k,j}^{rot}\right) = ro\left(\mathfrak{I}_{per}\right) = ro\left(\left(i_{k,j}^{per}\right)\right)$$

being

$$\begin{cases} i_{k,j}^{rot} = i_{k,j}^{per} & if \; \rho_i = 0 \\ i_{k,j}^{rot} = i_{j,9-k}^{per} & if \; \rho_i = 1 \\ i_{k,j}^{rot} = i_{9-k,9-j}^{per} & if \; \rho_i = 2 \\ i_{k,j}^{rot} = i_{9-k,j}^{per} & if \; \rho_i = 3 \end{cases} \tag{47}$$

After this process, the matrix $\mathfrak{I}_{rot}$ is encrypted using the operation secret key Ω_1. The key is microfragmented and wrapped and the original raw data, obtaining a "cellular" key Ω_1^{cell}. In particular, the encrypted cell $\mathfrak{I}_{enc}$ is obtained as the XOR addition between $\mathfrak{I}_{rot}$ and Ω_1^{cell} Eq. (48). This function as main encryption mechanism may present a very strong or weak privacy level, depending on the characteristics of Ω_1. If cell Ω_1^{cell} is a truly random matrix, then, all possible values have the same probability, and this characteristic is transferred to the encrypted cell Eq. (49). On the other hand, because of the structure of XOR addition, given a value in the encrypted cell $\mathfrak{I}_{enc}$ any possible value in the rotated matrix $\mathfrak{I}_{rot}$ has the same probability of having produced that value Eq. (50). Then, considering the Shannon's information theory, the mutual information between the rotated and the encrypted cells represents the residual information that remains in the encrypted matrix $\mathfrak{I}_{enc}$ about the rotated one $\mathfrak{I}_{rot}$ Eq. (51). A simple calculation proves that this quantity is zero, showing the XOR operation is a valid privacy protection mechanism.

$$\mathfrak{I}_{enc} = \mathfrak{I}_{rot} \oplus \Omega_1^{cell} \tag{48}$$

$$P\left(i_{k,j}^{rot} = \xi_i\right) = P\left(\Omega_1^{cell,k,j} = \xi_i\right) = \varphi_1 = \frac{1}{2^N} \forall \xi_i \tag{49}$$

$$P\left(i_{k,j}^{rot} = \xi_j | i_{k,j}^{enc} = \xi_i\right) = \varphi_2 = \frac{1}{2^N} \forall \xi_i, \xi_j \tag{50}$$

$$I\left(\mathfrak{I}_{rot}; \mathfrak{I}_{enc}\right) = \sum_{i=0}^{2^N-1} \sum_{j=0}^{2^N-1} P\left(i_{k,j}^{rot} = \xi_j, i_{k,j}^{enc} = \xi_i\right) \cdot \log\left(\frac{P\left(i_{k,j}^{rot} = \xi_j, i_{k,j}^{enc} = \xi_i\right)}{P\left(i_{k,j}^{enc} = \xi_i\right)}\right) = 0 \tag{51}$$

It is important to note that sequence Ω_1 (the operation secret key) is a pseudo-random number sequence (generated by Trifork PRNG). The combination of permutationRow and rotation functions and XOR operation with the secret operation key is executed for L rounds. This scheme is lightweight while guarantees the secrecy of biometric information. Two different approaches are proposed to determine the value of L. In the first one, this value is fixed for all cells. In the second one, for each cell the value is different, taking values from samples in signal w^* Eq. (52).

$$L_i = w_i^* \tag{52}$$

Once the final encrypted cell $\mathfrak{I}_{enc}$ is obtained, after L rounds, it is unwrapped using the same zig-zag algorithm Eq. (40) to generate the original data stream.

3.4. Decryption procedure

In order to decrypt the encrypted matrix $\mathfrak{I}_{enc}$, it is essential the chaotic dynamics in the receptor and the transmitter are synchronized [37]. Different strategies may be followed to achieve this objective [8,64]. When both dynamics are synchronized, we can guarantee the same cipher configuration is applied to both, the receptor and the transmitter.

Then, the XOR encryption is easily reversed, as it is enough to re-apply the same operation using the same secret key (53).

$$\mathfrak{I}_{rot} = \mathfrak{I}_{enc} \oplus \Omega_1^{cell} \tag{53}$$

Besides, as both chaotic dynamics are synchronized, the rotation and confusion steps can be also undone. In this case, permutations and rotations must be undone in the exact opposite order in which they were applied. Thus, indexes $_i$, π_i and (σ_k, σ_j) must be calculated for all L rounds before triggering the decryption process, as they must be applied in the inverse order they are obtained. When all these parameters are available, it is enough to run the same algorithm employed during encryption to extract the raw information in the receptor.

4. Evaluation: Methods and methodology

In order to evaluate and test the proposed technology, we conducted an experimental validation, based on simulation scenarios and tools. This experimental

Table 3
Configuration parameters

Parameter	Value	Comments
a, b, c, d	10, 2, 3, 2	Hyperchaotic regimen
N	32	32-bit architecture
d	16	Recommended value for bit shift in Trifork

phase consisted of two phases: the first phase includes a security analysis of the proposed technique, while the second part considers a performance analysis (focused on the encryption delay).

All the experiments were supported by a simulation scenario describing a complex biometric system including three different devices: cameras for facial recognition, fingerprint readers and iris readers. This system represents a large infrastructure, for example an airport, where different biometric techniques are employed for different purposes (such as access control or criminal identification). Twenty-five devices belonging to each type are considered in all simulations. Each different device model is created according to commercial solutions, to simulate a realistic data stream (regarding both factors, data format and communication protocol). The fingerprint reader model was created according to R307 fingerprint Arduino module. Iris scanner was coded to simulate the behavior of IriMagic 100BK platform. And, finally, facial recognition is simulated to be performed by ESP-EYE embedded cameras.

Each subsystem was connected with a different authentication sever, where biometric information is decrypted and processed. Other effects such as packet losses or electromagnetic interferences are not considered in this experimental section.

To perform the experiments, the simulation scenario was implemented and executed using MATLAB 2017a software. All simulations were performed using a Linux architecture (Ubuntu 20.04 LTS) with the following hardware characteristics: Dell R540 Rack 2U, 96 GB RAM, two processors Intel Xeon Silver 4114 2.2G, HD 2TB SATA 7,2K rpm.

All simulations represented an operation time of seventy-two (72) hours. Each simulation was repeated twelve times, and final results were obtained as the average of all partial results. In order to perform all the described simulations, the system was configured using the parameters described in Table 3. Parameter L is selected as control parameter.

4.1. Methodology for the security analysis

In order to formally analyze the privacy and security level reached by the proposed technology, three different approaches may be done: the Kerckhoff's approach, based on analyzing the theoretical characteristics of the secret key; the Shannon's theory, where the statistical relation between information in the raw and the encrypted streams is numerically calculated through different indicators; and the Diffie-Hellman's approach, based on studying the resilience of the proposed solution against some key attacks. In this work, we are including a first study including all three perspectives.

The Kerckhoff's approach considers no security mechanism can be secret for an indefinite time, so the security level of any solution, at the end, depends on the properties of the key. In particular, three different indicators are analyzed: the key sensitivity, the key space and the resilience against the known-plaintext attack and the chosen-plaintext attack.

The key sensitivity represents how different are two encrypted messages when protected using similar keys. Strong security mechanisms generate totally different encrypted messages even if very similar keys are employed. The number (or percentage) of bits changing the final encrypted message (ΔK), for each bit differing in the secret key is a good representation of the key sensitivity. Bits to be modified are randomly selected. To calculate key sensitivity, four hundred key configurations homogenously distributed across the key space were randomly selected. For every key, the key sensitivity was measured for different values of ΔK. Results are the obtained as the average for all considered keys. The size of the key space is the number of all possible keys. As the size of the key space grows, safer will be the systems protected by our solution.

Finally, the resilience against the known-plaintext attack and the chosen-plaintext attack represents how different are two encrypted messages when two similar clear messages are employed. As when talking about the key sensitivity, the number (or percentage) of bits changing the final encrypted message (ΔX), for each bit differing in original message is a good representation of the resilience. Bits to be modified are randomly selected. To calculate resilience, eight hundred random messages were considered. For every random message and value of ΔX, the resilience is measured. Results are obtained as the average value for all random messages.

On the other hand, the Shannon's security analysis is based on statistical tests. These tests study how much information from the private original biometric streams is present in the encrypted messages. Many different indicators and tests may be employed to evaluate this value, but in this work, we are using six of them: the statistical

correlation between encrypted and original messages, the entropy of encrypted messages and the mutual information between encrypted and original messages, the histogram variance (strong cryptosystems generate encrypted messages uniformly distributed), the Number of Byte Change Rate (NBCR) and the Unified Average Changing Intensity (UACI) evaluating how different (in bits) are the original and the encrypted messages (in percentage and per bit, respectively), the sequence test that evaluates how random are the encrypted messages and, finally, the NIST PRNG suite analyzing how random are the secret operation keys.

Eight hundred (800) random messages, with a length of twelve (12) kilobytes were generated and encrypted using the proposed scheme. Raw and encrypted messages were introduced in standard libraries for correlation, entropy, and mutual information calculation. The final result is obtained as the average of all these 800 measures. In order to calculate the histogram variance, encrypted messages are formatted as integer numbers, and they are introduced in a standard library for histogram calculation and manipulation. The same process is performed to introduce encrypted messages in the sequence test and evaluate their randomness. Finally, the NBCR and UACI are calculated through the XOR operation of every pair of raw and encrypted messages. The number of bits set to the unit in the result is then measured (percentage is easily calculated considering the message length). The final results are obtained as the average value of all these 800 measures.

Finally, the Diffie-Hellman's scheme analyzes how resilient the proposed solution is against two basic cyber-attacks: the Known Message Attack (KMA) and the Encrypted Only Attack (EOA). In KMA, attackers have access to encrypted and original messages; while in EOA they only have access to encrypted messages. In strong crypto solutions, secret keys must not be revealed in any case.

4.2. Performance analysis: methodology

The proposed solution must show a performance compatible with Industry 4.0 requirements and scenarios. In particular, biometric information must flow at real time and the resource consumption must be low enough to allow deploying the proposed technology in small microcontrollers (typical in Industry 4.0 applications). In order to analyze these factors, two different experiments were carried out.

The first experiment evaluates the encryption delay introduced by the proposed solution in biometric authentication systems. The experiments consider different values for L parameter (non-chaotic configurations). The second experiment was focused on analyzing the memory consumption in real devices by the proposed algorithm. Results from this second experiment are compared to the state of the art.

For both experiments, a real ESP-EYE device was employed to code and deploy the proposed security mechanism (face recognition). That device was operating for 72 hours, and results about resource consumption and processing delay were collected and processed using MATLAB software. Real Industry 4.0 scenarios could integrate other devices such as fingerprint readers or iris scans. However, we decided to perform our experiments using only cameras because they are the most demanding devices, as they generate the highest data bitrate among all biometric devices and support the most complex data processing algorithms. Thus, if proposed solution is lightweight enough to operate with cameras, it is expected to work with other biometric devices.

5. Results and discussion

First, we are showing and discussing the results for the formal security analysis. We are first studying the key space. For a N-bit PRNG the maximum number we can obtain is 2^N. Besides, each matrix Ω_1^{cell} includes 64 numbers, and L different matrices are employed in the encryption process. Using the combinatory theory, we can easily calculate the size of the key space ks (54). As can be seen, the size of the key space may be freely increased by considering a higher number of bits in the PRNG of a higher number of rounds in the encryption process.

$$ks = 2^{64 \cdot L \cdot N} \tag{54}$$

Thus, we can always find a configuration with a key space large enough for any given application.

Figure 6 shows the key sensitivity for different system configurations. As can be seen, the growing rate in all cases is exponential, although it goes up as the number of rounds is increased, showing that schemes with higher round numbers are safer. Besides, as can be seen, differences between schemes with ten or more rounds are very small, including the scheme where L takes a variable chaotic value. This last option, in fact, is the one presenting the higher sensitivity, although it must be analyzed (in each scenario) if the added complexity to the security scheme when considering this approach

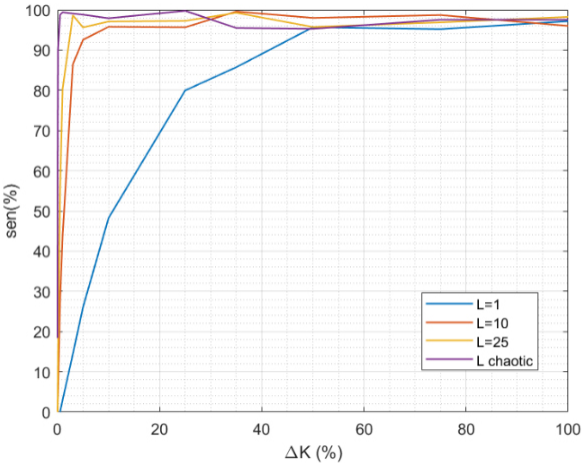


Fig. 6. Key sensitivity: Results.

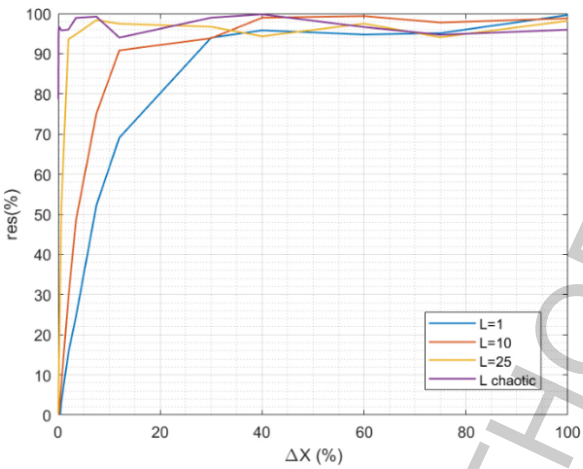


Fig. 7. Resilience against known-plaintext attack and chosen-plaintext attack: results.

is acceptable regarding the improvement obtained in return.

Any case, for schemes with $L = 10$ we already get a good behavior, so the sensitivity is around 100% for small variations in the key structure around 2%. These values are similar, or even better, than other previously reported technologies [15].

Figure 7 shows system resilience against known-plaintext and chosen-plaintext attacks. As can be seen, in this case the evolution is also exponential. The results are pretty similar to Fig. 6, as both the secret key and the raw message follow similar paths. Both flows are divided into cells and combined through the XOR operation which is a commutative operator. However, raw messages are randomized using additional rotation and confusion steps, so the resilience is expected to be higher than the key sensitivity given the same cipher configuration. Actually, as can be seen, given a difference of 20% in the secret key the key sensitivity is close to 70%, while for the same difference, the resilience is near 80%. Moreover, in this case, it is a clear improvement when using L as a variable and chaotic number of rounds. In this case, the resilience is almost 100% at any case. On the other hand, schemes with less than 25 rounds only reach that total resilience for messages differing, at least, around 25% of bits.

Table 4
Shannon's cryptoanalysis for the proposed solution

Parameter	Theoretical value	Experimental value
Information entropy	4096	4088.76
Mutual information	0	0.346
Correlation	0	−0.077
Histogram variance	0	4.01
NBCR	100	98.61
UACI	–	3078.14
Sequence test ($\alpha = 3$)	$\leqslant 3$	0.885

As a conclusion, the proposed system is secure, as it includes a clear configuration allowing a total resilience.

Now, we are analyzing the security analysis results according to Shannon's perspective. Table 4 shows the obtained results for the considered statistical indicators.

As can be seen, globally, the deviation of the real values from the ideal theoretical ones is around 3%. This amount is acceptable, and similar to other previously reported schemes [15]. In general, theoretical values are associated to totally random encrypted messages, so mutual information, correlation and variance must be null, and entropy equal to the number of bits in each cell. Deviation in entropy, mutual information and correlation may be considered negligible, and caused by pseudorandom flows. Variance shows a higher deviation that is caused by the underlying structure in PRNG and chaotic signal. This analysis is also supported by the sequence test, that shows a residual deterministic behavior in encrypted messages. This statistical test evaluates the significance level of the deterministic hypothesis (i.e. encrypted messages follow a predictable pattern). In this case, the hypothesis is rejected (results is below proposed value for α parameter), so encrypted messages do not follow any pattern. These results, any case, are coherent with the state of the art (as real random sequences cannot be generated through computational processes).

The values obtained for NBCR and UACI are coherent with all previous discussions, showing that the proposed scheme can also be considered secure from Shannon's perspective. Specifically, NBCR refers how many bits the encrypted and raw messages have in com-

Table 5
NIST PRNG test suite: Results

Test	Score
Random excursions	121/122
Cumulative sums	199/200
Random excursions variant	120/122
FFT	196/200
Runs	196/200
Rank	200/200
Longest run	195/200
Block frequency	194/200
Approximate entropy	196/200
Non-overlapping template	199/200
Linear complexity	199/200
Serial	199/200
Frequency	197/200
Universal	193/200

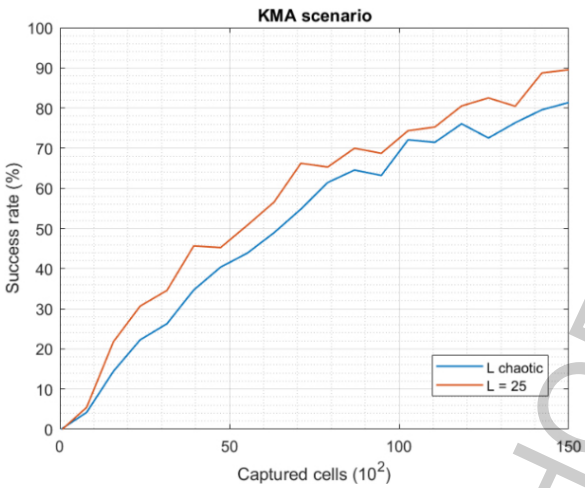


Fig. 8. Resilience against known message attack (KMA): Results.

mon (if both messages are totally different, NBCR takes value 100%). As can be see, the number of common bits is negligible.

The last test related to Shannon's view is the NIST PRNG test suite. Table 5 summarizes the obtained results. As can be seen, all tests were approved, even with a very high score. Thus, the operation secret key is good and random enough to ensure the security of the global proposed technology.

Finally, Figs 8 and 9 show the resilience of the proposed cryptosystem against the Known Message Attack (KMA) and the Encrypted Only Attack (EOA), following the Diffie Hellman's approach.

As can be seen, schemes where the number of rounds L is chaotic and variable present a better resilience than schemes where the number of rounds is fixed. This difference is maximum (around 10%) for a medium number of captured cells, while for small or large amounts the difference is smaller or null.

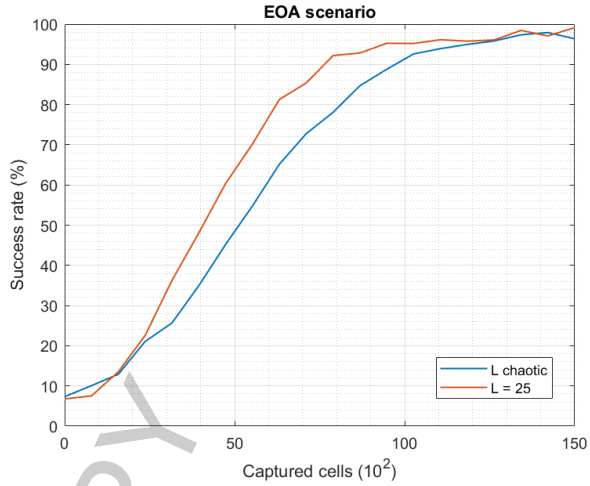


Fig. 9. Resilience against encrypted only attack (EOA): Results

In standard scenarios, the number of captured packets would tend to be small or medium, as large amounts are only possible in Man-in-the-Middle attacks, which are not possible in offline biometric information exchanges between sensors and servers in edge computing architecture.

In fact, the security level is usually defined as the required number of packets to reach a success rate of 90%. In this case, the security level against KMA is equal to $15 \cdot 10^3$ (higher for L chaotic) which is a very good value, compared to solutions in the state of the art. Regarding EOA, for a fixed number of rounds the security level is around $7.5 \cdot 10^3$, while it increases to $10 \cdot 10^3$ for L taking variable chaotic values. These results are also coherent, and even slightly better, than state-of-the-art mechanisms.

Once the security properties of the proposed mechanism are proved, the resource consumption of the technology must be analyzed, to study if it matches the Industry 4.0 requirements. Table 6 shows the memory and computational consumption of the proposed solution. RAM memory percentage refers the usage of the memory space for dynamic variables (compared to the available space in an ESP EYE device); while program space percentage refers the usage of the memory space for firmware in ESP EYE devices. As can be seen, even in resource-constrained devices, the memory usage caused by the proposed algorithm is globally around 10%, while the number of operations per encrypted cell is lower than other similar lightweight encryption schemes based on chaos [8]. In fact, the proposed scheme improves the resource consumption of similar proposals in around 25%.

Table 6
Resource consumption: Results

Encryption scheme	Use of RAM	Use of program space	Mathematical operations per cell
Proposal	12%	7%	409
[8]	19%	9%	521

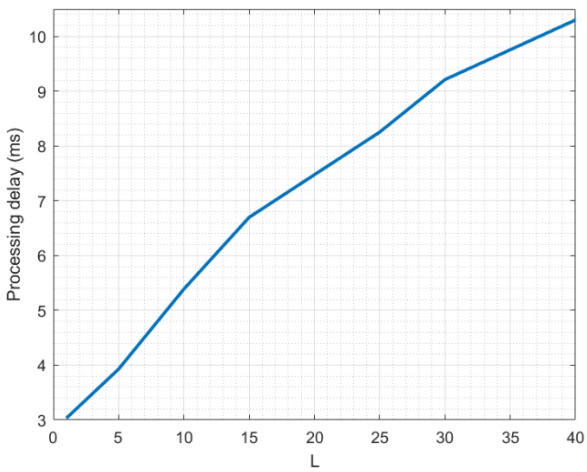


Fig. 10. Processing delay: Results.

Finally, Fig. 10 shows the processing delay caused by the proposed encryption solution, for different number of rounds in the encryption module (only non-chaotic configurations for L parameter). As can be seen, the temporal order is linear with respect to the number of rounds. Besides, even for large numbers (such as 40 rounds) the processing delay is in the range of tens of milliseconds. On the other hand, as seen in Figs 6 and 7, chaotic configurations for L parameter produce encryption schemes with a higher key sensitivity and resilience (to known-plaintext attack) than configurations with a fixed value. However, chaotic configurations introduce a variable computation delay, and the global jitter in the system is increased. Some Industry 4.0 systems (or applications) cannot tolerate this increasing jitter, so large values for L parameter can be then considered to get a similar behavior using non-chaotic configurations (although a higher processing delay is introduced). Any case, the encryption delay is much smaller than the one required by biometric algorithms, so we can conclude that the impact of the proposed solution is acceptable and compatible with Industry 4.0 applications.

6. Conclusions

In this paper, we propose a new lightweight secure symmetric encryption solution for Industry 4.0. This cipher includes a pseudo-random number generator based on simple computationally low-cost operations to create the secret key. To preserve and provide good security properties, the key generation and the encryption processes are fed with a chaotic number sequence obtained through the numerical integration of a new four-order hyperchaotic dynamic.

In general, we can conclude that the use of chaotic signal in encryption schemes improves their performance and security properties (key sensitivity and resilience), while the computation delay and jitter increase. Besides, with the proposed approach, based on simple binary operations, the resource consumption reduces up to 25% compared to the state-of-the-art mechanisms. This low-level approach requires the use of low-level programming languages and efficient data structures, so technological experts are essential to implement and deploy this new encryption system. In order to make easier its adoption in Industry 4.0 scenarios, prosumer mechanisms will be considered in future works.

Acknowledgments

This work is supported by the Ministry of Science, Innovation and Universities through the COGNOS project (PID2019-105484RB-I00) and by the European Commission by the Cities2030 project (H2020-FNR-2020-1. Grant no: 101000640).

References

[1] Lu Y. Industry 40: A survey on technologies, applications and open research issues. Journal of Industrial Information Integration. 2017; 6: 1-10.

[2] Bordel B, Alcarria R, Robles T, Martín, D. Cyber – physical systems: Extending pervasive sensing from control theory to the Internet of Things. Pervasive and mobile computing. 2017; 40: 156-184.

[3] Blunck E, Werthmann H. Industry 40 – an opportunity to realize sustainable manufacturing and its potential for a circular economy. In DIEM: Dubrovnik International Economic Meeting. 2017; 3(1): 644-666.

[4] Bordel B, Alcarria R, de Rivera DS, Robles T. Process execution in cyber-physical systems using cloud and cyber-physical internet services. The Journal of Supercomputing. 2018; 74(8): 4127-4169.

[5] Bordel B, Alcarria R, Hernández M, Robles T. People-as-a-Service dilemma: humanizing computing solutions in high-efficiency applications. In Multidisciplinary Digital Publishing Institute Proceedings. 2019; 31(1): 39.

[6] Benamara NK, Val-Calvo M, Alvarez-Sanchez JR, Díaz-Morcillo A, Ferrandez-Vicente JM, Fernández-Jover E, et al. Real-time facial expression recognition using smoothed deep

neural network ensemble. Integrated Computer-Aided Engineering. 2021; 28(1): 97-111.

[7] Bordel B, Iturrioz T, Alcarria R, Sanchez-Picot A. Provision of next-generation personalized cyber-physical services. In 2018 13th Iberian Conference on Information Systems and Technologies (CISTI), IEEE. June 2018, pp. 1-6.

[8] Bordel B, Alcarria R, Robles T, Iglesias, MS. Data authentication and anonymization in IoT scenarios and future 5G networks using chaotic digital watermarking. IEEE Access. 2021; 9: 22378-22398.

[9] Caraffini F, Neri F, Iacca G, Mol A. Parallel memetic structures. Information Sciences. 2013; 227: 60-82.

[10] Kumar P, Gaba GS. Biometric-based robust access control model for industrial Internet of Things applications. IoT Security: Advances in Authentication. 2020; 133–142.

[11] Robles T, Bordel B, Alcarria R, Sánchez-de-Rivera D. Enabling trustworthy personal data protection in eHealth and well-being services through privacy-by-design. International Journal of Distributed Sensor Networks. 2020; 16(5).

[12] Bordel B, Miguel C, Alcarria R, Robles T. A Hardware-Supported Algorithm for Self-Managed and Choreographed Task Execution in Sensor Networks. Sensors. 2018; 18(3): 812.

[13] Robles T, Bordel B, Alcarria R, de Andrés DM. Mobile Wireless Sensor Networks: Modeling and Analysis of Three-Dimensional Scenarios and Neighbor Discovery in Mobile Data Collection. Ad Hoc Sens Wirel Networks. 2017; 35(1-2): 67-104.

[14] Mareca P, Bordel B. A Chaotic Cryptographic Solution for Low-Range Wireless Communications in Industry 40.. In World Conference on Information Systems and Technologies, Springer, Cham, April 2019; pp. 134–144.

[15] Bordel B, Orúe AB, Alcarria R, Sánchez-De-Rivera D. An intra-slice security solution for emerging 5G networks based on pseudo-random number generators. IEEE Access. 2018; 6: 16149-16164.

[16] Bordel B, Alcarria R, Sánchez-de-Rivera D, Robles T. Protecting industry 40 systems against the malicious effects of cyber-physical attacks. In International Conference on Ubiquitous Computing and Ambient Intelligence, Springer, Cham, Nov. 2017; pp. 161–171.

[17] Singh S, Sharma PK, Moon SY, Park JH. Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. Journal of Ambient Intelligence and Humanized Computing. 2017; 1–18.

[18] Paz R, Pei E, Monzón M, Ortega F, Suárez L. Lightweight parametric design optimization for 4D printed parts. Integrated Computer-Aided Engineering. 2017; 24(3): 225-240.

[19] McKay K, Bassham L, Sönmez Turan M, Mouha N. NISTIR 8114, Report on lightweight cryptography. National Institute of Standards and Technology, March 2017.

[20] Iokibe K, Maeshima K, Kagotani H, Nogami Y, Toyota Y, Watanabe T. Analysis on equivalent current source of AES-128 circuit for HD power model verification. In 2014 International Symposium on Electromagnetic Compatibility, Tokyo, IEEE, May 2014, pp. 302–305.

[21] Gong Z, Nikova S, Law YW. KLEIN: a new family of lightweight block ciphers. In International Workshop on Radio Frequency Identification: Security and Privacy Issues, Springer, Berlin, Heidelberg, June 2011; pp. 1–18.

[22] Guo J, Peyrin T, Poschmann A, Robshaw M. The LED block cipher. In International workshop on cryptographic hardware and embedded systems, Springer, Berlin, Heidelberg, Sept. 2011; pp. 326-341.

[23] Rostami S, Neri F, Epitropakis M. Progressive preference articulation for decision making in multi-objective optimization problems. Integrated Computer-Aided Engineering. 2017; 24(4), 315–335.

[24] Beierle C, Jean J, Kölbl S, Leander G, Moradi A, Peyrin T, et al. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Annual International Cryptology Conference, Springer, Berlin, Heidelberg, Aug. 2016; pp. 123–153.

[25] Borghoff J, Canteaut A, Güneysu T, Kavun EB, Knezevic M, Knudsen LR, et al. PRINCE – a low-latency block cipher for pervasive computing applications. In International conference on the theory and application of cryptology and information security, Springer, Berlin, Heidelberg, Dec. 2012; pp. 208-225.

[26] Mohd BJ, Hayajneh T, Vasilakos AV. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. Journal of Network and Computer Applications. 2015; 58: 73-93.

[27] Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJ, et al. PRESENT: An ultra-lightweight block cipher. In International workshop on cryptographic hardware and embedded systems. Springer, Berlin, Heidelberg, Sept. 2007; pp. 450-466.

[28] Suzaki T, Minematsu K, Morioka S, Kobayashi, E. Twine: A lightweight, versatile block cipher. In ECRYPT Workshop on Lightweight Cryptography, Nov. 2011; pp. 146–169.

[29] Lim CH, Korkishko T. mCrypton – a lightweight block cipher for security of low-cost RFID tags and sensors. In International Workshop on Information Security Applications, Springer, Berlin, Heidelberg, Aug. 2005; pp. 243–258.

[30] Wheeler DJ, Needham RM. TEA, a tiny encryption algorithm. In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, Dec. 1994; pp. 363–366.

[31] Albrecht MR, Driessen B, Kavun EB, Leander G, Paar C, Yalçın T. Block ciphers – focus on the linear layer (feat. PRIDE). In Annual Cryptology Conference, Springer, Berlin, Heidelberg, Aug. 2014; pp. 57–76.

[32] Zhang W, Bao Z, Lin D, Rijmen V, Yang B, Verbauwhede I. RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms. Science China Information Sciences. 2015; 58(12): 1-15.

[33] Abdul-Latip SF, Reyhanitabar MR, Susilo W, Seberry J. On the security of NOEKEON against side channel cube attacks. In International Conference on Information Security Practice and Experience, Springer, Berlin, Heidelberg, May 2010; pp. 45–55.

[34] Leander G, Paar C, Poschmann A, Schramm K. New lightweight DES variants. In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, March 2007; pp. 196–210.

[35] Roda-Sanchez L, Olivares T, Garrido-Hidalgo C, de la Vara JL, Fernández-Caballero A. Human-robot interaction in industry 40 based on an internet of things real-time gesture control system. Integrated Computer-Aided Engineering. 2021; 28(2): 159-175.

[36] Ahmadlou M, Adeli H, Adeli A. Improved visibility graph fractality with application for the diagnosis of autism spectrum disorder. Physica A: Statistical Mechanics and its Applications. 2012; 391(20): 4720-4726.

[37] Ahmadlou M, Adeli H. Visibility graph similarity: A new measure of generalized synchronization in coupled dynamic systems. Physica D: Nonlinear Phenomena. 2012; 241(4): 326-332.

[38] Cai XT, Wang S, Lu X, Li WD, Liang YW. Parametric and adaptive encryption of feature-based computer-aided design

models for cloud-based collaboration. Integrated Computer-Aided Engineering. 2017; 24(2): 129-142.
[39] Beaulieu R, Shors D, Smith J, Treatman-Clark S, Weeks B, Wingers L. The SIMON and SPECK Families of Lightweight Block Ciphers. IACR Cryptol. ePrint Arch, 2013; 404.
[40] Rivest RL. The RC5 encryption algorithm. In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, Dec 1994; pp. 86–96.
[41] Yu J, Khan G, Yuan F. Xtea encryption based novel rfid security protocol. In 2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, May 2011, pp. 58–62.
[42] Antopolis S. Universal Mobile Telecommunications System (UMTS) LTE; 3G security. Lawful Interception architecture and functions, version, 10(0).
[43] Matsui M. New block encryption algorithm MISTY. In International Workshop on Fast Software Encryption, Springer, Berlin, Heidelberg, Jan. 1997; pp. 54–68.
[44] Baysal A, Şahin S. Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors. In Lightweight Cryptography for Security and Privacy, Springer, Cham, Sept. 2015; pp. 58–76.
[45] Kitsos P, Sklavos N, Provelengios G, Skodras AN. FPGA-based performance analysis of stream ciphers ZUC, Snow3g, Grain V1, Mickey V2, Trivium and E0. Microprocessors and Microsystems. 2013; 37(2): 235-245.
[46] Babbage S, Dodd M. The MICKEY stream ciphers. In New Stream Cipher Designs, Springer, Berlin, Heidelberg, 2008; pp. 191–209.
[47] Hell M, Johansson T, Meier W. Grain: A stream cipher for constrained environments. International Journal of Wireless and Mobile Computing. 2007; 2(1): 86-93.
[48] Guo J, Peyrin T, Poschmann A. The PHOTON family of lightweight hash functions. In Annual Cryptology Conference, Springer, Berlin, Heidelberg, Aug. 2011; pp. 222–239.
[49] Aumasson JP, Henzen L, Meier W, Naya-Plasencia M. Quark: A lightweight hash. In International Workshop on Cryptographic Hardware and Embedded Systems, Springer, Berlin, Heidelberg, 2010; pp. 1–15.
[50] Bogdanov A, Knežević M, Leander G, Toz D, Varıcı K, Verbauwhede I. SPONGENT: A lightweight hash function. In International Workshop on Cryptographic Hardware and Embedded Systems, Springer, Berlin, Heidelberg, Sept. 2011; pp. 312-325.
[51] Souissi R, Ben-Ammar M. An intelligent wireless sensor network temperature acquisition system with an FPGA. Wireless Sensor Network. 2014; 6(1): 1-7.
[52] Bordel B, Alcarria R, Martín D, Sánchez-de-Rivera D. An agent-based method for trust graph calculation in resource constrained environments. Integrated Computer-Aided Engineering. 2020; 27(1): 37-56.
[53] Huang Q, Yang Y, Shen M. Secure and efficient data collaboration with hierarchical attribute-based encryption in cloud computing. Future Generation Computer Systems. 2017; 72: 239-249.
[54] Naruse T, Mohri M, Shiraishi Y. Provably secure attribute-based encryption with attribute revocation and grant function using proxy re-encryption and attribute key for updating. Human-centric Computing and Information Sciences. 2015; 5(1): 8.
[55] Liang K, Au MH, Liu JK, Susilo W, Wong DS, Yang G, et al. A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing. Future Generation Computer Systems. 2015; 52: 95-108.
[56] Fugkeaw S, Sato H. Improved lightweight proxy re-encryption for flexible and scalable mobile revocation management in cloud computing. In 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), IEEE, June 2016, pp. 894-899.
[57] Baharon MR, Shi Q, Llewellyn-Jones D. A new lightweight homomorphic encryption scheme for mobile cloud computing. In 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, October 2015, pp. 618–625.
[58] Garcia-Bosque M, Pérez-Resa A, Sánchez-Azqueta C, Aldea C, Celma S. Chaos-based bitwise dynamical pseudorandom number generator on FPGA. IEEE Transactions on Instrumentation and Measurement. 2018; 68(1): 291-293.
[59] Yao X, Chen Z, Tian Y. A lightweight attribute-based encryption scheme for the Internet of Things. Future Generation Computer Systems. 2015; 49: 104-112.
[60] Aruna S, Usha G, Madhavan P, Kumar MR. Lightweight Cryptography Algorithms for IoT Resource-Starving Devices. Role of Edge Analytics in Sustainable Smart City Development: Challenges and Solutions. 2020; pp. 139–169.
[61] Al Salami S, Baek J, Salah K, Damiani E. Lightweight encryption for smart home. In 2016 11th International Conference on Availability, Reliability and Security (ARES), IEEE, August 2016, pp. 382–388.
[62] Baskar C, Balasubramaniyan C, Manivannan D. Establishment of light weight cryptography for resource constraint environment using FPGA. Procedia Computer Science. 2016; 78: 165-171.
[63] Mareca P, Bordel B. An intra-slice chaotic-based security solution for privacy preservation in future 5G systems. In World Conference on Information Systems and Technologies, Springer, Cham, March 2018; pp. 144–154.
[64] Mareca P, Bordel B. Robust hardware-supported chaotic cryptosystems for streaming commutations among reduced computing power nodes. Analog Integrated Circuits and Signal Processing. 2019; 98(1): 11-26.
[65] Liu W, Sun K, Zhu C. A fast image encryption algorithm based on chaotic map. Optics and Lasers in Engineering. 2016; 84: 26-36.
[66] Mareca M, Bordel B. Improving the complexity of the Lorenz dynamics. Complexity, 2017; Article ID 3204073.
[67] Tao H, Zain JM, Ahmed MM, Abdalla AN, Jing W. A wavelet-based particle swarm optimization algorithm for digital image watermarking. Integrated Computer-Aided Engineering. 2012; 19(1): 81-91.
[68] Bordel Sánchez B, Alcarria R, Robles T, Jara A. Protecting Physical Communications in 5G C-RAN Architectures through Resonant Mechanisms in Optical Media. Sensors. 2020; 20: 4104.
[69] Sarafyan D. Improved sixth-order Runge-Kutta formulas and approximate continuous solution of ordinary differential equations. Journal of Mathematical Analysis and Applications. 1972; 40(2): 436-445.
[70] Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM. 1978; 21(2): 120-126.
[71] Huta A. Une amélioration de la méthode de Runge-Kutta-Nystrôm pour la résolution numérique des équations différentielles du premier ordre. Acta Math. Univ. Comenian. 1956; 1: 201-224.

[72] Benamara NK, Val-Calvo M, Álvarez Sánchez JR, Díaz-Morcillo A, Ferrández de Vicente JM, Fernández-Jover E, Stambouli TB. Real-time facial expression recognition using smoothed deep neural network ensemble. Integr. Comput. Aided Eng. 2021; 28(1): 97-111.

[73] Pérez-Hurtado I, Martínez-del-Amor MA, Zhang G, Neri F, Pérez-Jiménez MJ. A membrane parallel rapidly-exploring random tree algorithm for robotic motion planning. Integr. Comput. Aided Eng. 2020; 27(2): 121-138.

[74] Siqueira H, Santana CJ, Jr., Macedo M, Figueiredo EMN, Gokhale A, Bastos Filho CJA. Simplified binary cat swarm optimization. Integr Comput Aided Eng. 2021; 28(1): 35-50.