Nikolaos F Karagiorgos\textsuperscript{1}, Stavros G Stavrinides\textsuperscript{2}, Carol de Benito\textsuperscript{3,4}, Spyridon Nikolaidis\textsuperscript{5}, and Rodrigo Picos\textsuperscript{3,4}

\textsuperscript{1}Physics Department, Aristotle University of Thessaloniki, Greece
\textsuperscript{2}Physics Department, International Hellenic University, Greece
\textsuperscript{3}University of Balearic Islands
\textsuperscript{4}Balearic Islands Health Institute (IdISBa)
\textsuperscript{5}Physics Department, Aristotle University of Thessaloniki, Greece

January 22, 2024

Abstract

The two main issues in the area of the Internet of Things are low resources consumption and secure data transmission. Conjugating both is fairly hard on ensuring security, leading to great efforts in research. The standard cryptography methods currently proposed are based on simplifications of standard protocols, but are still demanding on resources. Chaotic encryption is a way to reduce this burden, while keeping an equivalent level of security. In this paper, we propose, for the first time, a purely digital scheme for chaotic secure communications, able to be implemented in hardware or software without occupying most of the available resources. Next to the delivered analysis of the system, the experimental demonstration on both FPGA and ESP32 Arduino platforms of chaotic synchronization between transmitter and receiver, included examples of applied encrypted communication in the case of consecutive picture and text transmissions.
Unconventional Security for the IoT: Hardware and Software Implementation of a Digital Chaotic Encrypted Communication Scheme

Nikolaos F. Karagiorgos, Stavros G. Stavrinides, Senior Member, IEEE., Carol de Benito Member, IEEE, Spyridon Nikolaidis, Senior Member, IEEE, and Rodrigo Picos, Senior Member, IEEE

Abstract—The two main issues in the area of the Internet of Things are low resources consumption and secure data transmission. Conjugating both is fairly hard on ensuring security, leading to great efforts in research. The standard cryptography methods currently proposed are based on simplifications of standard protocols, but are still demanding on resources. Chaotic encryption is a way to reduce this burden, while keeping an equivalent level of security. In this paper, we propose, for the first time, a purely digital scheme for chaotic secure communications, able to be implemented in hardware or software without occupying most of the available resources. Next to the delivered analysis of the system, the experimental demonstration on both FPGA and ESP32 Arduino platforms of chaotic synchronization between transmitter and receiver, included examples of applied encrypted communication in the case of consecutive picture and text transmissions.

Index Terms—Secure communication, IoT, Chaotic encryption, Nonlinear circuits, Chaotic synchronization.

I. INTRODUCTION

SECURE communications have been one of the most important issues from the very beginning of the internet. In a technological environment increasingly dependent on information transmission the enormously increasing importance of security is undeniable, especially when one takes into account the applications in the frame of the Internet of Things (IoT) and the relevant boom in the need for data circulation [1]–[4].

The widespread adoption of the IoT, has led to more distributed, interconnected computing, as well as system architectures, further increasing the demand of data transfer. Within the IoT framework, it is evident that security should be a very important factor in the development of any application (smart homes [5], autonomous vehicles [6], health monitoring [7], etc.), especially when the relevant smart devices are remotely accessible and controllable. As the exchanged data may be very sensitive, without sufficient security, an intruder might take control over specific parts of the system and may, for example, compromise the owner’s privacy by spying through sensors or even cause physical injuries and in general sabotage the operation by exploiting some of the actuators. Thus, one could safely claim that since the IoT ecosystems enter all areas of everyday life, security issues can have privacy and safety implications, but the limitations imposed by hardware demand as simple and lightweight cybersecurity implementations as possible [1].

As far as lightweight implementations of secure data transmission are considered, it is mentioned the USA National Institute of Standards and Technology (NIST) launched in 2018 a call for the definition of requirements, selection process and the evaluation criteria of algorithms for constrained environments, where the performance of current NIST cryptographic standards is not acceptable [1]. These algorithms were evaluated both in hardware [8] and software [9] implementations. As a result of the process, the ASCON algorithm [10] was selected to be proposed as a standard for lightweight encryption [11]. However, all the applicants (the winner included) in this contest proposed version of secure communications protocols that were simply based on standard techniques, requiring private keys and other operations (like predefined s-boxes versions), which may be burdensome, among other due to the length of the required keys. At the same time, simplifying existing algorithmic approaches, apparently lead to reduced resistance in attacks.

To this direction, chaotic communication systems emerge as simpler in circuit design or software implementation, and thus, in power or other resources consumption, without compromising their resilience in attacks. Their value resides to the complexity of their behavior and not the complication of their design [12].

Chaotic communication systems are exploiting the unique property of chaotic synchronization [13]. According to this, a chaotic oscillator generates a chaotic signal, which is deterministic; thus, the knowledge of this procedure by an authorized receiver allows replication (synchronization) of this chaotic signal. Then recovering the information message is achieved by removing the chaotic carrier [14]–[16]. As a result, the confidentiality of the encryption technique is based on the difficulty to synchronize the receiver with the transmitter, by reproducing the chaotic carrier signal, if an intruder does not know the particular dynamical system used. It should be mentioned that chaotic sequences are not easy to predict, especially beyond the prediction horizon determined by the system’s inverse Lyapunov exponent [17], [18]. Consequently, security could be considered as an inherent property of such
systems and this is due to two key features of chaos, namely the emerging “noise-like” form of the time series and the crucial dependence on initial conditions and the system’s parameters. Both features allow for low probability of information detection and interception [19], [20]. Breaking the encryption means that the eavesdropper must be able to either synchronize another circuit to that of the transmitter, or solve the equations of the receiving system, assuming that he knows the system’s architecture. This is something difficult to achieve and it seems that it fills the gap created by current lightweight cryptographic schemes that are based on simplified versions of existing standard techniques, like those mentioned above.

Since secure communication is based on chaotic circuits operating in synchrony, the resulting system must be endowed with two additional attributes:

- **Synchronization stability.** This means that the system must be immune against small deviations of the matching between transmitter and receiver, without allowing easy synchronization of even non-identical circuits, thus, damaging security quality [13], [19].
- **Synchronization robustness.** Meaning that it has to be resilient up to a certain noise levels (internal or induced). Additionally, the system must be able to shield itself from intentional attacks by malevolent signals, which are not only trying to eavesdrop but also to destroy communication [13], [19]–[21].

The synchronized nonlinear circuits proposed until now are in most cases analog ones and very rarely mixed-signal ones. Even in the latter case, the design of the chaotic oscillating sub-circuit is achieved by means of an analog approach [22]. In this paper a purely digital, lightweight, chaotic-synchronized scheme for encrypted communication, suitable for IoT applications, is presented. The proposed scheme is implemented by two nonlinear chaotic circuits, i.e., a non-autonomous oscillator driven by the information to be encrypted and a properly designed circuit able to decode the encrypted information by utilizing chaotic synchronization. Next to the system’s analysis, a hardware (on an FPGA) and a software (on an Arduino) implementation are presented, confirming the proof of concept as far as encryption is considered. The experimental demonstration includes examples of applied encrypted communication in the case of consecutive picture and text transmissions. When compared with standard proposals of lightweight encryption ([1], [8]–[11]), the hereby presented approach demonstrates merits that allows it to be considered for applications in the frame of the IoT, without any compromise in the overall security.

**II. DIGITAL CHAOTIC COMMUNICATION SCHEME**

The design of the digital chaotic communication scheme presented in this paper is somehow a derivative scheme, based on a mixed-signal communication setup, initially presented in [23] and thoroughly studied in [12]. This setup is taking advantage of chaotic synchronization for modulating, encrypting, transmitting and retrieving information; thus establishing it as one demonstrating intrinsic security features, based on an unconventional approach, i.e., chaotic encryption. In the presented hereby approach, a fully digital communication system is implemented, by designing the digital circuits of a chaotic encoder/transmitter (its design, realization and study has been reported in [24]) and a decoder/receiver suitable for demodulating the chaotic-modulated information. The overall chaotic communication scheme appears in Fig. 1. All the circuits comprising this system and its overall operation are presented below.

**A. The Encoder - Transmitter**

As already mentioned, a digital circuit having the ability of encoding by modulating and transmitting information over a chaotic carrier has been proposed and studied in [24]. This circuit is a non-autonomous, nonlinear, digital oscillator able to operate in a chaotic mode. A block diagram reenacting its architecture appears in the upper half of Fig. 1. It disposes a digital input $M(n)$ and an output that demonstrates a digital chaotic signal $F(n)$, in the sense of a digital signal whose pulse duration exhibits a chaotic distribution. This chaotic digital carrier-signal contains the encrypted information.

The oscillator’s feedback path, involving the shifted Heaviside function $H(V_0 - V_2)$ and the XOR gate, is forming an important element for the circuit’s operation, that of the nonlinearity, essential for demonstrating a chaotic behavior [17]. In this case, $V_0$ is set to the positive value of 0.35. The XOR gate arranges a path utilized for inserting the information as a digital signal $M(t)$, further modulating the transmitter’s chaotic output $F(n)$. This feedback path is a requisite, since information signal $M(t)$ is providing the scheme with an external stimulus, essential for non-autonomous oscillating systems to exhibit a chaotic behavior.

![Fig. 1. The block diagram of the chaotic synchronized encoding-decoding scheme. The top part encodes the message M(n) producing the chaotic signal F(n), that is fed into the decoder (bottom), providing the recovered signal M’(n).](image-url)
This non-autonomous oscillatory loop configuration involves an IIR filter, appearing in Fig. 2, which is driven by the amplitude-bounded output signal \( c_1 F(n-1) + c_2 F(n-2) \). As pointed out in [24], it is obvious that any instability that could cause the responses \( V_1 \) and \( V_2 \) to escape to infinity, would originate from this IIR structure. Since in a dynamical system, its deterministic chaotic evolution is by definition bounded in amplitude, any system variable blowing up to infinity is improper and its behavior is unwelcome. Consequently, the IIR filter design should ensure that its poles would lie within the unit cycle. For this reason a stability analysis necessary for ensuring the filter’s bounded behavior, has taken place in [24] and it defined the proper sampling period value to be \( T_S = 3.2 \mu s \). This value led to a clock frequency \( 1/T_S = 312.5 \) kHz, further ensuring the stability of the overall oscillation.

The difference equations describing the operation of this oscillator are the following [24]:

\[
\begin{align*}
V_1(n) &= \alpha_{11} V_1(n-1) + \alpha_{12} V_1(n-2) + \\
&+ \alpha_{21} V_2(n-1) + \alpha_{22} V_2(n-2) + \\
&+ c_1 F(n-1) + c_2 F(n-2) \\
V_2(n) &= b_{11} V_1(n-1) + b_{12} V_1(n-2) + \\
&+ b_{21} V_2(n-1) + b_{22} V_2(n-2) \\
F(n) &= M(n) \oplus H[V_0 - V_2(n)]
\end{align*}
\]

(1)

The values for the coefficients of the above equations (1) have been calculated in accordance to the sampling period (clock) \( T_S \) value that ensures boundedness and stability, necessary for the operation of the oscillator. They were calculated according to the procedure in [24], and are provided in Table I. Notice that those values are dimensionless, since the system is a digital one. Almost all parameter values came up to be irrational, forcing us to use approximations in the real circuit implementation. Those approximations had to be compatible with the arithmetic available on the FPGA utilized, as explained and presented in the section describing the implementation of the system.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_{11} )</td>
<td>421/425</td>
<td>( b_{11} )</td>
<td>-12/25</td>
</tr>
<tr>
<td>( \alpha_{12} )</td>
<td>4/1275</td>
<td>( b_{12} )</td>
<td>4/25</td>
</tr>
<tr>
<td>( \alpha_{21} )</td>
<td>12/25</td>
<td>( b_{21} )</td>
<td>421/425</td>
</tr>
<tr>
<td>( \alpha_{22} )</td>
<td>-4/25</td>
<td>( b_{22} )</td>
<td>4/1275</td>
</tr>
<tr>
<td>( c_1 )</td>
<td>16/15</td>
<td>( c_2 )</td>
<td>-16/45</td>
</tr>
</tbody>
</table>

TABLE I

The calculated values of all the parameters in equation set 1. The calculation resulted from the IIR stability analysis [24].

B. The Decoder - Receiver

The design of the receiver topology was aiming to create a circuit implementing the following three features: (i) to demonstrate a chaotic operation, thus being a nonlinear circuit topology, (ii) to synchronize its chaotic operation to that of the transmitter, and (iii) to successfully demodulate/decode information signal out of the transmitted chaotic carrier [16].

The block diagram of the proposed and implemented receiver circuit appears in the lower half of Fig. 1. Though the proposed topology looks very much alike the transmitter’s, it is not the same, since the receiver circuit has no feedback path. Conceptually, it has two parts: i) the upper part, which forms a driven local oscillator; and ii) the lower part, which involves the demodulating-decoding logic, essential for retrieving the initial information.

The driven local oscillator is a circuit with identical topology than the corresponding non-autonomous oscillatory loop configuration of the transmitter, presented above. It involves an identical IIR filter, appearing in Fig. 2, which is fed with the input signal \( F(n) \). Assuming that this driving signal is the same signal with the one driving the corresponding local oscillator in the transmitter circuit, then it is expected that the receiver’s oscillator would operate in synchrony with the one in the transmitter circuit, in a chaotic mode though. Thus, their outputs \( V_1 - V_2 \) and \( V_1' \) and \( V_2' \) would get the same values simultaneously. At the same time, the required IIR filter stability is ensured in the case of the receiver circuit topology, as well. It is apparent that this way the receiver’s dynamics are fully driven by inserting the proper input signal, a fact that is beneficial to establishing both a chaotic behavior, as well as synchronization.

The difference equations describing the receiver’s local oscillator driven operation are the following:

\[
\begin{align*}
V_1'(n) &= \alpha_{11}' V_1'(n-1) + \alpha_{12}' V_1'(n-2) + \\
&+ \alpha_{21}' V_2'(n-1) + \alpha_{22}' V_2'(n-2) + \\
&+ c_1 F(n-1) + c_2 F(n-2) \\
V_2'(n) &= b_{11}' V_1'(n-1) + b_{12}' V_1'(n-2) + \\
&+ b_{21}' V_2'(n-1) + b_{22}' V_2'(n-2)
\end{align*}
\]

(2)

All the coefficients involved in equation set (2) have the same values than their namesakes in Table I, thus \( \alpha_{ij}' = \alpha_{ij} \),

---

![Fig. 2. Block diagram of the IIR filter, essential part of the circuit topology of the non-autonomous, digital chaotic oscillator. Its stability analysis defines the proper clock frequency.](image-url)
within the system, for any initial difference \( \Delta \), the unilateral z operation by driving it with the transmitter’s output signal way that it is fully synchronized to the transmitter’s chaotic inclusions on its ability for synchronized operation. Defining chaotic synchronized communication scheme and draw con-

The transmitter output is a digital chaotic signal \( F(n) \) created by an XOR procedure between the information signal \( \alpha \) and \( \beta \), as well.

C. The Overall Secure Communication Scheme

Referring to the overall operation of the proposed digital, secure, chaotic communication scheme, appearing in Fig. 1, this is governed by the set of equations in (1) and (2), which describe the encoder/transmitter and the decoder/receiver dynamics, respectively. As mentioned, the parameter values for which the system demonstrates stable dynamics, are listed in Table 1 and they have to be the same for both the transmitter and the receiver, i.e., \( \alpha_{ij} = \alpha_{ji}, b_{ij} = b_{ji} \) and \( c_{ij} = c_{ji} \). The transmitter output is a digital chaotic signal \( F(n) \), created by an XOR procedure between the information signal \( M \) and another locally produced digital, chaotic signal \( H(n, V_0) \). At the same time, the receiver circuit is designed in such a way that it is fully synchronized to the transmitter’s chaotic operation by driving it with the transmitter’s output signal \( F(n) \).

Introducing error variables into equation sets (1)-(2), one may obtain the resulting error dynamics for the proposed chaotic synchronized communication scheme and draw conclusions on its ability for synchronized operation. Defining the error variables, in this case:

\[
\begin{align*}
\Delta V_1(n) &= V_1(n) - V_1'(n) \\
\Delta V_2(n) &= V_2(n) - V_2'(n)
\end{align*}
\]

the obtained equations governing the error dynamics are:

\[
\begin{align*}
\Delta V_1(n) &= \alpha_1 \Delta V_1(n-1) + \alpha_2 \Delta V_1(n-2) + \alpha_3 \Delta V_2(n-1) + \alpha_4 \Delta V_2(n-2) \\
\Delta V_2(n) &= b_1 \Delta V_1(n-1) + b_2 \Delta V_1(n-2) + b_3 \Delta V_2(n-1) + b_4 \Delta V_2(n-2)
\end{align*}
\]

To study the evolution of the errors as these are propagating within the system, for any initial difference \( \Delta V_1(0), \Delta V_2(0) \) the unilateral z transform is employed. The characteristic polynomial of the above equations has the form:

\[
1 - (a_{11} + b_{21})z^{-1} - (a_{12} + b_{22} - a_{21}b_{11} - a_{11}b_{22})z^{-2} - (a_{22}b_{11} + a_{21}b_{12} - a_{12}b_{21} - a_{11}b_{22})z^{-3} + (a_{12}b_{22} + a_{22}b_{12})z^{-4} = 0
\]

The magnitude of all four roots of this polynomial has been calculated and they appear to possess a value less than 1. Consequently, any initial difference would decrease exponentially to zero, \( \Delta V_1(n) \rightarrow 0, \Delta V_2(n) \rightarrow 0 \). Therefore, an asymptotically stable synchronization is expected to take place, leading to the obvious situation of robust synchronization, \( V_1(n) \rightarrow V_1'(n), V_2(n) \rightarrow V_2'(n) \), and by extension to the synchronous behavior of the two nonlinear element-functions \( H(V_0 - V_2) \rightarrow H(V_0 - V_2') \).

Considering the chaotic synchronized communication capability of the proposed scheme, a well-known and unique feature that the XOR digital operators exhibit, is utilized. This feature is based on the sum-mod2 property: \( X \oplus X = 0 \). This way the transmitted original, chaotic-encoded information signal is decoded-demodulated in the receiver. As it is obvious from the lower part of Fig. 1, receiver’s output signal \( M(n) \) is related to transmitter’s output by the following relation:

\[
M(n) = F(n) \oplus H(V_0 - V_2(n)) = M(n) \oplus H(V_0 - V_2(n)) \oplus H(V_0 - V_2'(n))
\]

Then, if we set both Heaviside functions to be equal, we obtain the desired result:

\[
M(n) = M(n)
\]

It is apparent from this relation that the error free information retrieval at the receiver, is dependent on the quality of synchronization between the two nonlinear elements realized by the shifted Heaviside functions in both the transmitter and the receiver.

III. IMPLEMENTATION

To move beyond a simple proof of concept and test experimentally the proposed secure, digital, chaotic-synchronized communication scheme, this was implemented in two different platforms: i) a FPGA, and ii) an Arduino development board.

In the case of the FPGA implementation, we used two FPGAs; one was encoding and transmitting an encrypted black and white picture (due to RAM limitations) and then receive, and the second FPGA was decoding and retrieving the original picture. Successful synchronized transmission had previously taken place with colored pictures (larger files), as well.

The Arduino system used also two separate boards, with the same functionality than the two FPGAs. Both of them were connected to a PC running RedHat, which was used to send and receive a text.

A. FPGA Implementation

In order to proceed with the realization of the described system in the FPGAs, we initially had to make the proper choice for the suitable arithmetic scheme to be leveraged. The arithmetic adopted in this realization was the Q6.10 fixed-point signed arithmetic. This means that all parameter and signal values were 16-bit signed binary numbers; with the six most significant digits representing the integer part; actually the most significant bit of the six is devoted to the sign. The proposed arithmetic achieved the required accuracy for the
coefficients (Table I), while it ensured the defined chaotic but stable operation of the circuits.

The Q6.10 arithmetic was preferred to others, after trying a variety of arithmetic schemes with different number lengths in simulation environment (MATLAB-Simulink), for being the most inexpensive in terms of gate usage on the FPGA chip. At the same time, this approach resulted to an implementation, which was using the least word length, while it exhibited a good quality of chaotic behavior and synchronization robustness, compared to other implementations using larger number of bits (24 and 32 bits).

Each sub-circuit in Fig. 1, namely the encoder/transmitter and the decoder/receiver, both incorporating the IIR filter appearing in Fig. 2, have been directly implemented onto reconfigurable hardware. We opted for realizing the proposed overall communication system, in two different FPGA circuits. One FPGA was used to realize the transmitter architecture, while the other one the receiver circuit. This methodology was preferred to the one implementing transmitter and receiver architectures in one FPGA, since this way a more realistic approach of the scheme’s implementation and operation is adopted, in terms of possible problems that could emerge in clocking and signal transmission. The platforms used were two identical Xilinx SPARTAN3E FPGAs. It should be noted that in both FPGAs no multiply-accumulate (MAC) unit was used. Instead, shifting and adding operations were preferred, resulting to a reduced number of the required slices.

Regarding the transmitting module, the extra circuits needed for storing the information of the picture and pre-processing it, appear in Fig. 3(a). It is apparent that next to the encoding chaotic oscillator, the FPGA’s build-in RAM for storing the information and a unit implementing the RAM address management, are necessary. In this example the information signal applied to the encoder was pulse-width modulated (PWM), therefore the corresponding modulating unit was also included.

As already mentioned the build-in RAM-block, with a size of 16384 bits, was utilized for storing the information data to be transmitted. This could have the form of any multimedia signal or typical data in general; in this example the information had the form of a picture, which was then forwarded to the pulse-width modulator, before sent to the chaotic encoder. The Control Logic Unit, was a Finite State Machine (FSM) coordinating the communication of the RAM-block with the PWM modulator (which was implemented by another FSM, as well), by providing the address values, essential for the RAM-block to properly communicate with the other circuits of the system. It should be mentioned that the size of the RAM in the specific FPGA was 16384 bits.

Since the transmitted signal (due to the encryption procedure) was a serial chaotic PWM one and the connection between the two different FPGAs was a simple wired connection, we had to use a suitable for the situation and effective technical standard for avoiding clocking synchronization problems. Therefore, the low-voltage differential signaling (LVDS) scheme was adopted [25]. This ensured the proper, noise-immune transmission of the chaotic-encrypted signal $F(n)$ over cable, while clocking synchronization would also be ensured.

At the receiver, the encrypted information $M(n)$ in the chaotic modulated, digital signal $F(n)$ is decoded by the driven chaotic oscillator. This is achieved by exploiting the synchronization property theoretically proved in the previous section (see equation (3)), thus $M'(n) = M(n)$. Further on, the pulse-width demodulator provides the initial, raw, transmitted information, which is send to a shift register (SR), so that it could be monitored and stored by utilizing chipscope-pro.

The proposed implementation, as illustrated in Fig. 3, needs a proper clocking scheme. Within this scheme, three clock signals were implemented: the CH_CLK, DATA_CLK and
PWM_CLK. The clocking for the operation of the chaotic oscillator (encoder) had a period of \( T_{\text{CH_CLK}} = 3.2\mu s \). This value has been calculated according to the IIR filter stability analysis [24] and it corresponds to a frequency \( f_{\text{CH_CLK}} = f_0 = 312.5kHz \). As a result, the DATA_CLK, which is the clock needed for the operation of the FSM implementing the control unit that manages the RAM addressing procedure and its operation, had a period value of \( T_{\text{DATA_CLK}} = 51.2\mu s \). This value is calculated as \( f_{\text{DATA_CLK}} = f_{\text{CH_CLK}}/16 \), since that for every 16-bits encoded at the local chaotic oscillator, a new value should be transferred from the RAM to the PWM module. The PWM_CLK applied at the PWM modulator, had a period of \( T_{\text{PWM_CLK}} = 12.8\mu s \), i.e. 4 times the DATA_CLK period. The described clocking scheme appears in Fig. 5. Finally, regarding the PWM output, a typical scheme of 75% duty cycle for representing input-bit 1, at the input, and 25% for representing 0, was used. The same clocking scheme was also applied at the receiver module with the PWM_CLK applied at the PWM demodulator and DATA_CLK to the SR and Chipscope-Pro.

In order to implement the coefficient values into the FPGA, these must be represented using the Q6.10 fixed point arithmetic. Thus, the approximated values for all the coefficients in equation sets (1) and (2), appearing in the second column of Table II, were also calculated using the Q6.10 representation. Both values are very close, with a typical error less than 0.1%, except for two cases, where the error goes up to slightly less than 7%. In any case, experimental the experimental results shown in the next section, prove that the system is still behaving in the desired region under these conditions.

Assessing the FPGA implementation synthesis of the described circuits, for the utilized hereby SPARTAN3E (xc3s500e-5fg320), the consumed resources appear in Table III. It is apparent that utilization of the FPGA’s resources is extremely low, not only due to the proposed design but also due to the approach of encoding information by utilizing chaotic circuits.

### Table II

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Q6.10 Value</th>
<th>error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha_{11} )</td>
<td>0.9905882</td>
<td>0.9902343750</td>
<td>0.04</td>
</tr>
<tr>
<td>( \alpha_{12} )</td>
<td>0.0031372</td>
<td>0.0029296875</td>
<td>0.02</td>
</tr>
<tr>
<td>( \alpha_{21} )</td>
<td>0.4800000</td>
<td>0.4794921875</td>
<td>0.04</td>
</tr>
<tr>
<td>( \beta_{11} )</td>
<td>-0.1481480</td>
<td>-0.1474609375</td>
<td>0.02</td>
</tr>
<tr>
<td>( \beta_{12} )</td>
<td>0.1600000</td>
<td>0.159796875</td>
<td>0.04</td>
</tr>
<tr>
<td>( \beta_{21} )</td>
<td>0.9905882</td>
<td>0.9902343750</td>
<td>0.04</td>
</tr>
<tr>
<td>( \beta_{22} )</td>
<td>0.0031372</td>
<td>0.0029296875</td>
<td>0.02</td>
</tr>
<tr>
<td>( c_1 )</td>
<td>1.0666666</td>
<td>1.0664062500</td>
<td>0.02</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>-0.3555550</td>
<td>-0.3554687500</td>
<td>0.02</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Available</th>
<th>Used</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slices</td>
<td>4656</td>
<td>29</td>
<td>0.62%</td>
</tr>
<tr>
<td>Slice flip-flops</td>
<td>9312</td>
<td>1</td>
<td>0.01%</td>
</tr>
<tr>
<td>4-input LUTs</td>
<td>9312</td>
<td>56</td>
<td>0.60%</td>
</tr>
<tr>
<td>Bonded IOPs</td>
<td>232</td>
<td>35</td>
<td>15.08%</td>
</tr>
<tr>
<td>MULT18x18SIOs</td>
<td>20</td>
<td>5</td>
<td>25.00%</td>
</tr>
<tr>
<td>GCLKs</td>
<td>24</td>
<td>1</td>
<td>4.16%</td>
</tr>
</tbody>
</table>

### Table IV

<table>
<thead>
<tr>
<th>FPGA Library</th>
<th>LUTs</th>
<th>FFs</th>
<th>Slices</th>
<th>Multipliers</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper (*)</td>
<td>56</td>
<td>1</td>
<td>29</td>
<td>5</td>
</tr>
<tr>
<td>ASCON_GMU-v2</td>
<td>1790</td>
<td>974</td>
<td>513</td>
<td>-</td>
</tr>
<tr>
<td>Elephant-v5</td>
<td>2645</td>
<td>1502</td>
<td>759</td>
<td>-</td>
</tr>
<tr>
<td>GIFT-COFB_GMU-v6</td>
<td>2363</td>
<td>872</td>
<td>696</td>
<td>-</td>
</tr>
<tr>
<td>PHOTON-Beetle-v1</td>
<td>2065</td>
<td>729</td>
<td>620</td>
<td>-</td>
</tr>
<tr>
<td>Romulus-v5</td>
<td>887</td>
<td>422</td>
<td>246</td>
<td>-</td>
</tr>
<tr>
<td>TinyJAMBU-TJT-v3</td>
<td>576</td>
<td>432</td>
<td>215</td>
<td>-</td>
</tr>
<tr>
<td>Xoodyak_XF-v8</td>
<td>2443</td>
<td>559</td>
<td>618</td>
<td>-</td>
</tr>
</tbody>
</table>

### Comparison

**Comparison between hardware resources used in this paper's proposal and other implementations, as described in [8]. (*) Notice that our proposal has been implemented on a Spartan3E, while the others used a Xilinx Artix-7, so the comparison is in only qualitative.**
The proposed implementation of our algorithm is compared in Table IV against implementations of those evaluated by NIST as the IoT encryption algorithm in FPGA [8]. Notice that our algorithm was implemented in a Spartan3E, while the implementations in [8] used a Xilinx Artix-7 and an Intel Arria-10. Since both of these last implementations are comparable (up to a factor of 3 in the number of elements), we have used only the Artix-7 figures. With those conditionants, we can still see that the number of required elements is much lower than in other, more complex, implementations of state-of-the-art algorithms. This is caused by the fact that our implementation deals with a smaller number of bits at the same time than any of the other implementations, thus reducing the required number of elements and, we expect, the required energy per bit. Notice that this is a task that has not been performed in this paper.

B. Arduino Implementation

As already mentioned, the Arduino implementation served as a proof of concept of implementing an unconventional encryption technique in an IoT platform. The code implementing the algorithms of the transmitting and receiving module, i.e. Equations 1 and 2, can be accessed and downloaded from https://github.com/rpicos-uib/chaotic_transmission_2022. The encryption/decryption scheme were each implemented in a different Arduino as is shown in Fig. 6, where the connections between them, through digital I/O ports, are also illustrated.

One of the Arduinos was configured as the transmitter, while the other was configured as the receiver. In order to analyze the process, both of them were communicating separately through a different serial port with a PC running Matlab under Linux.

Fig. 6. The Arduino implementation of the digital, chaotic-synchronized encryption scheme.

IV. DEMONSTRATION OF CHAOTIC SYNCHRONIZATION AND EVALUATION

The above described digital chaotic oscillator [24], suitable for encoding information $M(n)$, turns to be a very robust circuit. In particular, the encoding/transmitting digital topology implements a chaotic system with minimum embedding dimension $m_{\text{min}} = 5$ (consistent with a 4th order non-autonomous chaotic system), which has folded to a fractal dimension $v = 3.4$, while the rate of the system’s loss of information (Kolmogorov-Sinai entropy) is $K = 0.40 \text{bits/s}$ [24]. Additionally, it exhibits chaotic behavior in a quite broadband spectrum of excitation frequencies $f_{\text{exc}}$, as the maximal Lyapunov exponent spectrum in [24] shows.

A. FPGA Tests

An illustration of the chaotic behavior of the FPGA-realized encoding/transmitting system appears in Fig. 7. In Figs 7(a) and (b) two snapshots of the encoder’s IIR filter outputs $V_1$ and $V_2$ (the decimal equivalent values) are presented together with the corresponding phase portrait in Fig. 7(c), which is apparently a strange attractor. Notice that in this case a simple pulse series was driving the PWM stage, prior to the chaotic oscillators circuit. The expected chaotic behavior is apparent as already documented by the analysis provided in [24].

For demonstrating and evaluating the implemented communication system’s performance, the quality of the chaotic synchronization should be assessed. Towards this, the synchronized operation of the overall system implemented in the two different FPGAs (appearing in 1), was tested with a driving
pulse series. In Fig. 8(a) the driving frequency (upper green plot), representing the applied information signal $M(t)$ to be encoded, is presented, while the received and decoded signal $M'(t)$ appears in the lower part (blue plot). The transmitted chaotic, digital signal $F(t)$ appears in the middle (red plot) of Fig. 8(a). Synchronized operation is apparent when comparing the information signal (green plot) to the decoded one (blue plot).

Another interesting issue related to the synchronization robustness, led us to the investigation of the speed of achieving synchronization. In Fig. 8b a graph providing information on the question of how quickly the two subsystems are getting synchronized is presented. In this figure we consider the worst case scenario in their initial conditions, i.e., the longest time to synchronization. The vertical axis show the difference (in mV) between the output $V_2$ of the transmitter, and $V_2$ in the receiver, which are the points where the signal is entered and recovered, respectively. It is apparent from this graph that synchronization is achieved very quickly (about 500 clocks), and therefore the system’s robustness is confirmed.

As a next step transmission encoded, real data had to be verified. Therefore, the encoding and decoding of a colored picture was initially attempted within a simulation environment, namely a VHDL-testbench. This way the overall synchronized operation of the system was tested with real data. In Fig. 9(a) the transmitted (colored) picture is presented, while in Fig. 9(b) the received, decoded one appears. Checking the quality of chaotic synchronization the difference of these two pictures was calculated and it is presented in Fig. 9(c). In this figure the color-value of each pixel in the encoded picture was subtracted from corresponding one in the decoded picture; in the case of a zero result the color of the corresponding pixel was set black, otherwise it was set white. The emerging total
Fig. 11. The first paragraph from "El Quijote" utilized in the Arduino-based implementation of FPGA implementation of a chaotic, encrypted image transmission system, based on the digital chaotic communication scheme. (a) is the original text, (b) the transmitted characters, and (c) is the decoded text.

black picture in Fig. 9(c) clearly confirms full synchronization between the transmitting and the receiving system.

With the same procedure, the system’s encryption capability was briefly tested by simply interchanging coefficients’ values \( a_{ij} \leftrightarrow b_{ij} \) in the receiver’s equation 2. For the transmitted picture in 9(a), the received, decoded image appears in 9(d). Full desynchronization is evident, providing with a simple and abstract proof of the system’s encoding ability, from the point of view of cybersecurity.

Finally, the system was experimentally tested with real data, as well. In order to be really realistic, the transmitting and receiving sub-systems were implemented in two different FPGAs. In this case, we opted for a simple black and wide picture, due to the limited memory size of the built-in RAM of the Xilinx SPARTAN3E FPGA; the available space for storing data was limited to 16384 bits. Therefore the transmitted picture was the black and white code appearing in 10(a), while the received figure appears in Fig. 10(b). Both pictures were identical, something that was further confirmed by their difference result illustrated in Fig. 10(c), since the total black outcome proved their exact and corresponding similarity.

B. Arduino Tests

In the case of the implementation of the studied communication system by programming two different arduinos, this was tested by sending a text in Spanish, just to have a larger character set than the one offered by English language, thus increasing complexity. The text corresponds to the first paragraphs of "El Quijote", the famous novel by Miguel de Cervantes, written in the XVI century. The original text is depicted in Fig. 11 (a). The resulting, according the proposed schema, encrypted text is shown in Fig. 11 (b), while in Fig. 11 (c) the decoded message appears. It is apparent that the decrypted message is exactly the same compared to the original one. Looking at the encrypted message, it can be seen that there are some values that are not encrypted, but the message is still unreadable. As a way to check the security of the encryption (see also the discussion in Section V), we have plotted the ascii codes of the encoded text versus those of the original/recovered text in Fig. 12. As is clearly seen, the proposed process spreads the input values into a very broad spectrum, thus showing its effectiveness. A plot showing the correlation between the original and the codified characters is provided in Fig 13 as a red line. This same plot shows also the sensitivity of the correct decodification under a variation of the parameters \( P_0 \) listed in Table II. From this picture, it is clear that even a \( \pm 1\% \) variation is enough to scramble the output.

On the other hand, we have also performed a comparison of the implementation of this algorithm against those evaluated by NIST as the IoT encryption algorithm [9]. Specifically, we have compared the used memory and the time needed to encode a byte between all the finalists and our proposal when using the ESP32 implementation. The results are shown in Table V. From this table, it is clear that the proposed algorithm needs less memory. This is due to the fact that the chaotic implementation is simpler than any of the other proposal, which are based on another paradigm, and require more operations to be performed, as well as many more internal variables and parameters. This difference is also shown in the time needed to encode a byte, which is also fairly lower for the same reasons.
Fig. 13. Sensitivity of the decoding process to the matching of the parameters in the encoder and the decoder. The horizontal axis are the codes of the original message, while the vertical axis correspond to the decoded values. The blue points are the result of a mismatched decoding, while the red points correspond to a perfectly matched decodification. The left picture corresponds to a set of parameters $P$ corresponding to the original parameters $P_0$ decreased in a 1%, while the right picture shows the effect of a 1% increase.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Memory (kB)</th>
<th>Speed ($\mu$s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>This paper</td>
<td>28.8</td>
<td>5.39</td>
</tr>
<tr>
<td>AES-GCM</td>
<td>33.9</td>
<td>67.75</td>
</tr>
<tr>
<td>ASCON</td>
<td>32.2</td>
<td>22.86</td>
</tr>
<tr>
<td>Elephant</td>
<td>33.7</td>
<td>3986.3</td>
</tr>
<tr>
<td>GIFT-COFB</td>
<td>33.6</td>
<td>55.61</td>
</tr>
<tr>
<td>GRAIN-128AED</td>
<td>33.0</td>
<td>119.72</td>
</tr>
<tr>
<td>ISAP</td>
<td>32.6</td>
<td>608.46</td>
</tr>
<tr>
<td>PHOTON-Beetle</td>
<td>33.4</td>
<td>185.76</td>
</tr>
<tr>
<td>Romulus</td>
<td>33.5</td>
<td>114.16</td>
</tr>
<tr>
<td>SPARKLE</td>
<td>32.5</td>
<td>62.2</td>
</tr>
<tr>
<td>TinyJAMBU</td>
<td>32.3</td>
<td>43.59</td>
</tr>
<tr>
<td>Xoodyak</td>
<td>32.6</td>
<td>39.18</td>
</tr>
</tbody>
</table>

TABLE V

Comparison between used memory and speed ($\mu$s per encrypted byte) in an ESP32 for different state-of-the-art IoT algorithms. The results for the other algorithms are extracted from the NIST evaluation [11] based on [9].

V. DISCUSSION AND CONCLUDING REMARKS

As already mentioned, implementing chaotic circuits in the digital domain is an issue out of the mainstream, but in the authors’ opinion this combination could lead to applications, which is now mature enough to get exploited by current trends in the technological landscape, and become beneficial in domains like the IoT. Combining the virtues of chaotic electronics like inherent information encoding, UWB transmission, low power (green electronics), with the standard advantages of a digital implementation, i.e., easy, cheap and mismatch free implementation, easy incorporation in existing designs and easy upgrade to fabrication process improvements, one can easily understand that a promising, lightweight-implemented cybersecurity approach is introduced, in general.

From this point of view a fully digital chaotic, secure, communication scheme is presented in this paper. It has to be noted that a digital implementation as presented in this paper has distinct advantages over the normal analog or mixed-signal ones typically found in the literature: it is easier to interface with an already existing digital circuit, it is immune to the fabrication process variations, and its parameters and initial conditions can be fixed exactly to a desired value. It is the derivative of an analog chaotic synchronized communication scheme [12]. The proposed scheme utilizes chaotic synchronization in a master-slave system. The spectrum of excitation frequencies in which the digital chaotic oscillator exhibits chaos, is very different and more broad than the analog prototype [12], [24]. The information (in a digital form) to be encoded and transmitted drives the chaotic synchronized circuits and it is then successfully decoded.

The presented hereby analysis of the circuital topologies comprising the chaotic-synchronized master-slave, encrypting-decrypting communication system, documented its ability to synchronize, its robustness, as well as the fidelity in information transmission, expected to demonstrate.

Towards the implementation of the system in an FPGA platform, the design was extended so that it would include a PWM modulating/demodulating stage, as well as all the relevant circuitry necessary to manage and control the successful information-data encryption and decryption process. It is worth mentioning that in this implementation an LVDS module, connecting the two different FPGAs’ transmitter and receiver circuits, was utilized so that no clocking issue would emerge.

The system was initially tested with a simple pulse series driving and its chaotic behavior was demonstrated and confirmed. Additionally, the chaotic synchronization property between transmitter and receiver was ratified, while studying the speed of getting the system synchronized, this was found to be rather quick and adequate for a real world application.

Testing the system with real information, encrypted transmission was successfully attempted in the case of a colored picture (Matlab), a QR code (FPGA) and text (Arduino). Additionally, we have also shown that a small mismatch between the parameters of the receiver and the transmitter causes the recovered and original information to be uncorrelated. Those tests proved the fidelity of chaotic synchronization and the efficiency of the proposed secure communication scheme. It is also noted that the proposed scheme has the property of easy and precise frequency scaling. Finally, we have compared in Table V our (non-optimized) implementation on an ESP32 against the results from the NIST standard IoT evaluation [9], and we have found that our implementation requires less memory usage, and offers also a better data transmission rate. A pending task is testing the system against side channel attacks.

It is also important to note that the presented scheme is immune to most side channel attacks. For instance, a timing attack doesn’t make sense, since the time needed to perform the encryption/decryption is independent of the key (values of the parameters) and the message. A differential power analysis would also find difficulties, since the interaction with the message is done bit by bit in a single gate, which can be very easily obfuscated. The main vulnerability would be a brute force attack, trying to extract the 5 parameters the $a, b, c$ constants depend on, as well as the 4 initial conditions defining the system [24]. In this specific implementation using $N = 10$ bits, that would mean a search space of $(2^N)^5 = 2^{50}$ possibilities. The attack can be carried out by searching the absolute minimum of the hamming distance [26], but it still
has to perform a search in a 9-dimensions space, with many possible local minima, thus requiring a lot of computational power. Notice that this kind of attack could be defeated by adding an additional (chaotic) system that would change the parameters dynamically after some also changing amount of information transmitted, thus making the brute force attack impractical.

Future work will concentrate on studying and testing the scheme’s robustness under noisy conditions, evaluate its encryption capability (possible improvements could be introduced as, for instance, use it in conjunction with a PUF as in [27]) and integrate it within a wireless platform.

REFERENCES

BIographies

Nikolaos F. Karagiorgos received the B.S. degree in physics from the University of Ioannina, Ioannina, Greece, in 2008, and the M.Sc. degree in electronic physics-radioelectrology from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2011. His interests include chaotic circuits and systems, digital design, DSP algorithms, CMOS cell modeling, and IoT applications.

Stavros G. Stavrinides (SM’04–AM’08–M’10–SM’13) is a Physicist with a MSc in Electronics and a PhD in Chaotic Electronics, all awarded by Aristotle University of Thessaloniki, Thessaloniki, Greece. He currently serves as a Professor of NonLinear Dynamics and Electronics at Physics Department, International Hellenic University, Greece. His research interests include, non-exhaustively, chaotic electronics and their applications (with emphasis on hardware-security), analog and mixed-signal electronic circuits, experimental chaotic synchronization, nonlinear time series analysis, complex networks, Physical Unclonable Functions (PUF) and memristors. Prof. Stavrinides has authored or co-authored more than 100 journal and conference papers and book chapters while he edited two books. Finally, he has contributed, as a researcher, in several national (Greek) and international (EU, NATO) funded projects.
Carola de Benito holds a M.S. in physics (1991) and a Ph.D. in Electronic Engineering (2012), both from the Universitat de les Illes Balears. Currently, she is an Associate Professor in the Electronic Technology Group of the Industrial and Construction Engineering Department, UIB. Her research interests include device and circuit modeling low-temperature CMOS design and memristor device and memristive system modeling. She has published more than 25 journal and conference papers and she has participated in several national and international projects.

Spyridon Nikolaidis received the Diploma and PhD degrees in electrical engineering from Patras University, Greece, in 1988 and 1994 respectively. Since September 1996 he has been with the Department of Physics of the Aristotle University of Thessaloniki, Greece where he is currently a full professor. His current research interests include: modeling the operations of basic CMOS structures, development of analytical expressions for the propagation delay and the power consumption of logic gates, design of high speed and low power digital circuit and embedded systems, modeling the power consumption of embedded processors. He is author and co-author in more than 200 scientific articles in international journals and conference proceedings He was member of the organization committee of three international conferences. He organizes the annual International Conference on Modern Circuit and System Technologies (MOCAST) as well as the 27th International Symposium on Power and Timing Modeling, Optimization and Simulation for 2017. He also contributes to a number of research projects funded by European Union and Greek Government where in many of that he has the scientific responsibility. He is a Senior member of IEEE.

Rodrigo Picos (SM’2006) received the M.S. and Ph.D. degrees from the Universitat de les Illes Balears (UIB), Palma (Illes Balears), Spain, in 1998 and 2006, respectively. He is currently a Professor with the Industrial Engineering and Construction Department at the University of Balearic Islands, and a member of the Balearic Islands Health Institute (IdISBa) Palma, Mallorca, Spain. His research interests include memristive systems and compact-device modeling, as well as analog circuit design and test. He has taught courses on Electron Device Modelling, Electronic Instrumentation, and basic Electronics. Dr. Picos has authored or co-authored more than 120 journal and conference papers and he has participated, as a researcher, in several national and international (EU) funded projects.