EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Design and FPGA implementation of a wireless hyperchaotic communication system for secure real-time image transmission

Said Sadoudi[1], Camel Tanougast[2*], Mohamed Salah Azzaz[1] and Abbas Dandache[2]

Abstract

In this paper, we propose and demonstrate experimentally a new wireless digital encryption hyperchaotic communication system based on radio frequency (RF) communication protocols for secure real-time data or image transmission. A reconfigurable hardware architecture is developed to ensure the interconnection between two field programmable gate array development platforms through XBee RF modules. To ensure the synchronization and encryption of data between the transmitter and the receiver, a feedback masking hyperchaotic synchronization technique based on a dynamic feedback modulation has been implemented to digitally synchronize the encrypter hyperchaotic systems. The obtained experimental results show the relevance of the idea of combining XBee (Zigbee or Wireless Fidelity) protocol, known for its high noise immunity, to secure hyperchaotic communications. In fact, we have recovered the information data or image correctly after real-time encrypted data or image transmission tests at a maximum distance (indoor range) of more than 30 m and with maximum digital modulation rate of 625,000 baud allowing a wireless encrypted video transmission rate of 25 images per second with a spatial resolution of 128 × 128 pixels. The obtained performance of the communication system is suitable for secure data or image transmissions in wireless sensor networks.

Introduction

Over the past decades, the confidentiality of multimedia communications such as audio, images, and video has become increasingly important since communications of digital products over the network (wired/wireless) occur more frequently [1,2]. Therefore, the need for secure data and transmission is increasing dramatically and defined by the required levels of security depending on the purpose of communication. To meet these requirements, a wide variety of cryptographic algorithms have been proposed.

In this context, the main challenge of stream cipher cryptography relates to the generation of long unpredictable key sequences. More precisely, the sequence has to be random, its period must be large, and the various patterns of a given length must be uniformly distributed over the sequence.

Traditional ciphers like DES, 3DES, IDEA, RSA, or AES are less efficient for real-time secure multimedia data encryption systems and exhibit some drawbacks and weakness in the high stream data encryption [3,4]. Indeed, the increase and availability of a high-power computation machine allow a force brute attack against these ciphers. Moreover, for some applications which require a high-level computation and where a large computational time and high computing power are needed (for example, encryption of large digital images), these cryptosystems suffer from low-level efficiency [5]. Consequently, these encryption schemes are not suitable for many high-speed applications due to their slow speed in real-time processing and some other issues such as in the handling of various data formatting.

Over the recent years, considerable researches have been taken to develop new chaotic or hyperchaotic systems and for their promising applications in real-time encryption and communication [6-8]. In fact, it has been shown that chaotic systems are good candidates for designing cryptosystems with desired properties [9].

*Correspondence: camel.tanougast@univ-lorraine.fr
[2]Équipe ASEC - Laboratoire Conception, Optimisation et Modélisation des Systèmes, Université de Lorraine, Metz, France
Full list of author information is available at the end of the article

The most prominent is sensitivity dependence on initial conditions and system parameters, and unpredictable trajectories.

Furthermore, chaos-based and other dynamical system-based algorithms have many important properties such as the pseudorandom properties, ergodicity and non-periodicity. These properties meet some requirements such as sensitivity to keys, diffusion, and mixing in the cryptographic context. Therefore, chaotic dynamics is expected to provide a fast and easy way for building superior performance cryptosystems, and the properties of chaotic maps such as sensitivity to initial conditions and random-like behavior have attracted the attention to develop data encryption algorithms suitable for secure multimedia communications. Until recently, chaotic communication has been a subject of major interest in the field of wireless communications. Many techniques based on chaos have been proposed such as additive chaos masking (ACM) [10], where the analog message signal is added to the output of the chaos generator within the transmitter. In [11], chaos shift keying is used where the binary message signal selects the carrier signal from two or more different chaotic attractors. Authors in [12] use chaotic modulation where the message information modulates a parameter of the chaotic generator. Chaos control methods [13,14] rely on the fact that small perturbations cause the symbolic dynamics of a chaotic system to track a prescribed symbol sequence. In [15], the receiver system is designed in an inverse manner to ensure the recovery of the encryption signal. An impulsive synchronization scheme [16] is employed to synchronize chaotic transmitters and receivers. However, all of these techniques do not provide a real and practical solution to the challenging issue of chaotic communication which is based on extreme sensitivity of chaotic synchronization to both the additive channel noise and parameter mismatches. Precisely, since chaos is sensitive to small variations of its initial conditions and parameters, it is very difficult to synchronize two chaotic systems in a communication scheme. Some proposed synchronization techniques have improved the robustness to parameter mismatches as reported in [16,17], where impulsive chaotic synchronization and an open-loop-closed-loop-based coupling scheme are proposed, respectively. Other authors proposed to improve the robustness of chaotic synchronization to channel noise [18], where a coupled lattice instead of coupled single maps is used to decrease the master-slave synchronization error. In [19], symbolic dynamics-based noise reduction and coding are proposed. Some research into equalization algorithms for chaotic communication systems are also proposed [20]. For other related results in the literature, see [21-23]. However, none of them were tested through a real channel under real transmission conditions. Digital synchronization can overcome the failed attempts to realize experimentally a performed chaotic communication system. In particular, when techniques exhibit any difference between the master/transmitter and slave/receiver systems, it is due to additive information or noise channel (disturbed chaotic dynamics) which breaks the symmetry between the two systems, leading to an accurate non-recovery of the transmitted information signal at the receiver. In [24], an original solution to the hard problem of chaotic synchronization high sensibility to channel noise has been proposed. This solution, based on a controlled digital regenerated chaotic signal at the receiver, has been tested and validated experimentally in a real channel noise environment through a realized wireless digital chaotic communication system based on zonal intercommunication global-standard, where battery life was long, which was economical to deploy and which exhibited efficient use of resources, known as the ZigBee protocol. However, this synchronization technique becomes sensible to high channel noise from a higher transmission rate of 115 kbps, limiting the use of the ZigBee and Wireless Fidelity (Wi-Fi) protocols which permit wireless transmissions up to 250 kbps and 65 Mbps [25,26], respectively. Consequently, no reliable commercial chaos-based communication system is used to date to the best of our knowledge. Therefore, there are still plentiful issues to be resolved before chaos-based systems can be put into practical use. To overcome these drawbacks, we propose in this paper a digital feedback hyperchaotic synchronization and suggest the use of advanced wireless communication technologies, characterized by high noise immunity, to exploit digital hyperchaotic modulation advantages for robust secure data transmissions. In this context, as results of the rapid growth of communication technologies, in terms of reliability and resistance to channel noise, an interesting communication protocol for wireless personal area networks (WPANs, i.e., ZigBee or ZigBee Pro Low-Rate-WPAN protocols) and wireless local area network (WLAN, i.e., Wi-Fi protocol WLAN) is developed. These protocols are identified by the IEEE 802.15.4 and IEEE 802.11 standards and known under the name ZigBee and Wi-Fi communication protocols, respectively [25]. These protocols are designed to communicate data through hostile Radio Frequency (RF) environments and to provide an easy-to-use wireless data solution characterized by secure, low-power, and reliable wireless network architectures. These properties are very attractive for resolving the problems of chaotic communications especially the high noise immunity property. Hence, our idea is to associate chaotic communication with the WLAN or WPAN communication protocols. However, this association needs a numerical generation of the chaotic behavior since the XBee protocol is based on digital communications. In the hardware area, advanced modern digital signal processing devices, such as field programmable gate array (FPGA), have been widely used to generate numerically the chaotic dynamics or the encryption keys [27-31]. The advantage of these techniques is that the parameter mismatch problem does not exist contrary to the analog techniques. In addition, they offer a large possible integration of chaotic systems in the most recent digital communication technologies such as the ZigBee communication protocol. In this paper, a wireless hyperchaotic communication system based on dynamic feedback modulation and RF XBee protocols is investigated and realized experimentally. The transmitter and the receiver are implemented separately on two Xilinx Virtex-II Pro circuits [32] and connected with the XBee RF module based on the Wi-Fi or ZigBee protocols [26,33]. To ensure and maintain this connection, we have developed a VHSIC (very high speed integrated circuit) hardware description language (VHDL)-based hardware architecture to adapt the implemented hyperchaotic generators, at the transmitter and receiver, to the XBee communication protocol. Note that the XBee modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage-compatible Universal Asynchronous Receiver/Transmitter (UART) [33]. The used hyperchaotic generator is the well-known and the most investigated hyperchaotic Lorenz system [34]. This hyperchaotic key generator is implemented on FPGA technology using an extension of the technique developed in [27-29] for three-dimensional (3D) chaotic systems. This technique is optimal since it uses directly VHDL description of a numerical resolution method of continuous chaotic system models. A number of transmission tests are carried out for different distances between the transmitter and receiver. The real-time results obtained validate the proposed hardware architecture. Furthermore, it demonstrates the efficiency of the proposed solution consisting on the association of wireless protocols to hyperchaotic modulation in order to build a reliable digital encrypted data or image hyperchaotic communication system.

The remainder of this paper is organized as follows: the 'Hyperchaotic synchronization and encryption technique' section proposes an adapted feedback hyperchaotic synchronization based on a dynamic feedback modulation. This section details the proposed synchronization and data masking principle by considering hyperchaotic systems. The 'FPGA implementation of hyperchaotic Lorenz generator' section briefly introduces continuous Lorenz hyperchaotic system, which is used as key stream generators of the proposed digital encryption hyperchaotic modulation. This section then details the hardware architecture used for implementing the Lorenz hyperchaotic generator. A register transfer level (RTL) architecture for embedded hardware implementation of the considered key stream generator is also given in this section. The 'Experimental framework' section presents our experimental framework used to realize and validate the wireless hyperchaotic communication scheme. This section gives details of the transmitter and the receiver blocks with a short description of the XBee RF modules. The 'Wireless real-time data or image transmission tests and results' section presents different real-time results proving that the proposed system is suitable for efficient secure real-time data or image transmissions of wireless sensor networks. Performance analysis through real wireless data transmission tests is also discussed in this section. The 'Security analysis' section gives the statistical analysis of the proposed image encryption scheme, which increases the complexity of the random bit generation and hence making it difficult for an intruder to extract information about the proposed encryption/decryption hyperchaotic modulation. Finally, the 'Conclusions' section draws appropriate conclusions.

Hyperchaotic synchronization and encryption technique

Contrary to a trigger-based slave/receiver chaotic synchronization by the transmitted chaotic masking signal, which limits the performance of the rate synchronization transmission [24], we propose a digital feedback hyperchaotic synchronization (FHS). More precisely, we investigate a new scheme for the secured transmission of information based on master-slave synchronization of hyperchaotic systems, using unknown input observers. The proposed digital communication system is based on the FHS through a dynamic feedback modulation (DFM) technique between two Lorenz hyperchaotic generators. The proposed FHS-DFM technique used for chaotic masking communications is depicted in Figure 1. This technique is an extension and improvement of the one developed in [35] for synchronizing two 3D continuous chaotic systems in the case of a wired connection. The proposed digital feedback communication scheme synchronizes the master/transmitter and the slave/receiver by the injection of the transmitted masking signal in the hyperchaotic dynamics of the slave/receiver. The basic idea of the FHS is to transmit a hyperchaotic drive signal $S(t)$ after additive masking with a hyperchaotic signal $x(t)$ of the master (transmitter) system (x, y, z, w). Hyperchaotic drive signal is then injected both in the three subsystems (y, z, w) and (y_r, z_r, w_r). The subscript r represents the slave or receiver system (x_r, y_r, z_r, and w_r). At the receiver, the slave system regenerates the chaotic signal $x_r(t)$ and a synchronization is obtained between two trajectories $x(t)$ and $x_r(t)$ if

$$\lim_{t\to\infty} |x(t) - x_r(t)| = 0. \tag{1}$$

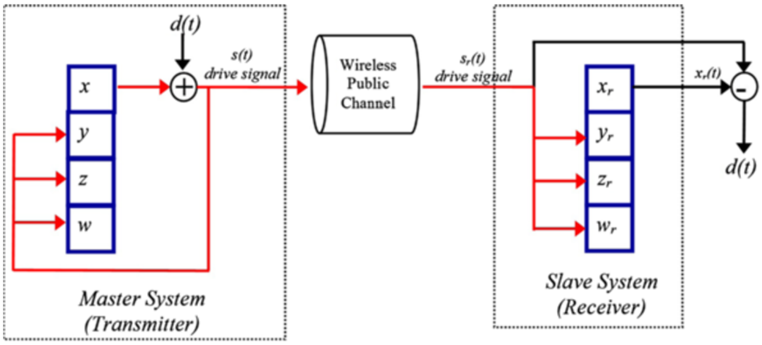

Figure 1 HS-DFM principle scheme. This figure presents the hyperchaotic feedback synchronization via a dynamic feedback modulation (HS-DFM) technique.

This technique can be applied to chaotic modulation. In our case, it is used for generating hyperchaotic keys for stream cipher communications, where the synchronization between the encrypter and the decrypter is very important. Therefore, at the transmitter, the transmitted signal after the additive hyperchaos masking (digital modulation) is

$$S(t) = x(t) + d(t), \tag{2}$$

where $d(t)$ is the information signal and $x(t)$ is the hyperchaotic carrier. At the receiver, after synchronization of the regenerated hyperchaotic signal $x_r(t)$ with the received signal $S_r(t)$ and the demodulation operation, we can recover the information signal $d(t)$ correctly as follows:

$$d(t) = S_r(t) - x_r(t). \tag{3}$$

Therefore, the slave/receiver will generate a hyperchaotic behavior identical to that of the master/transmitter allowing to recover correctly the information signal after the demodulation operation. The advantage of this technique is that the information signal $d(t)$ does not perturb the hyperchaotic generator dynamics, contrary to the ACM-based techniques of [10] and [36], because $d(t)$ is injected at both the master/transmitter and slave/receiver after the additive hyperchaotic masking (Figure 1). Thus, for small values of information magnitude [35], the information will be recovered correctly. It should be noted that we have already confirmed this advantage by testing experimentally the HS-DFM technique performances for synchronizing hyperchaotic systems (four-dimensional (4D) continuous chaotic systems) in the case of wired connection between two Virtex-II Pro development platforms. After many experimental tests and from the obtained real-time results, we concluded that the HS-DFM is very suitable for wired digital chaotic communication systems. However, in the present work, one of the objectives is to test and study the performances of the HS-DFM technique in the presence of channel noise through real-time wireless communication tests as it is shown in Figure 1. To perform the proposed approach, a digital implementation of the master and slave hyperchaotic systems is required. Therefore, we investigate the hardware implementation of the proposed FHS-DFM technique between two Lorenz hyperchaotic generators using FPGA. To achieve this objective, we propose the following details of the proposed architecture.

FPGA implementation of hyperchaotic Lorenz generator

System model and numerical resolution

In general, autonomous continuous hyperchaotic systems are modeled by the following four non-linear differential equation systems:

$$\begin{aligned}
\dot{x} &= F(x, y, z, w),\\
\dot{y} &= G(x, y, z, w),\\
\dot{z} &= Q(x, y, z, w),\\
\dot{w} &= P(x, y, z, w),
\end{aligned} \tag{4}$$

where F, G, Q, and P are non-linear equations and x, y, z, and w are the four state variables of the dynamical system. For computing the solutions of the system (4), we use the fourth order Runge-Kutta (RK-4) numerical method for resolving the continuous chaotic system models because it produces a more accurate estimate of the solution [27-29]. The flowchart of this method for resolving system (4) is illustrated in Figure 2, where x_0, y_0, z_0, and w_0 are the initial conditions, h is the iteration step, and k_i, m_i, l_i, and p_i ($i = 0$ to 3) are the intermediate slopes.

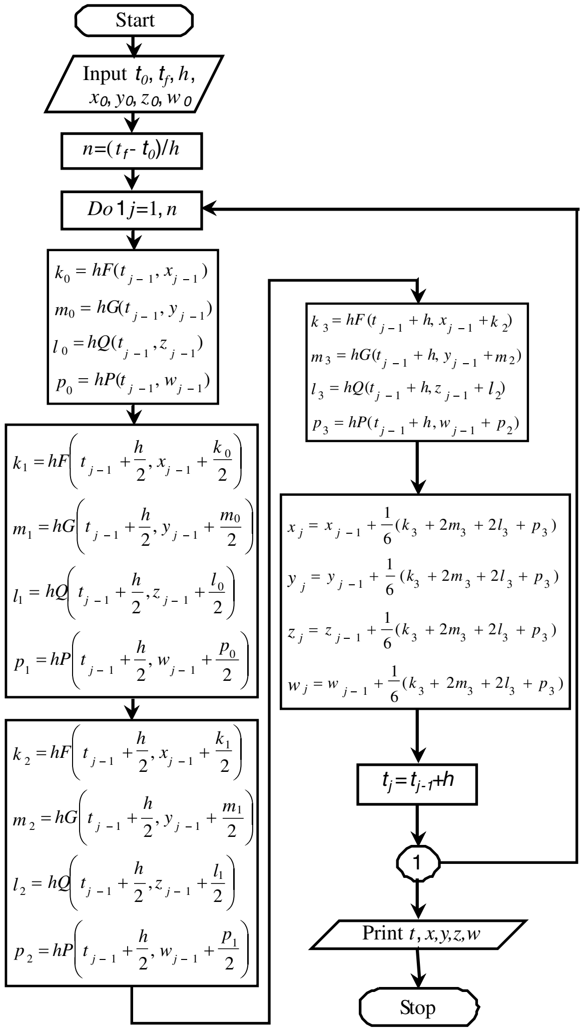
Start
Input $t_0, t_f, h, x_0, y_0, z_0, w_0$
$n=(t_f - t_0)/h$
Do 1 $j=1, n$
$k_0 = hF(t_{j-1}, x_{j-1})$
$m_0 = hG(t_{j-1}, y_{j-1})$
$l_0 = hQ(t_{j-1}, z_{j-1})$
$p_0 = hP(t_{j-1}, w_{j-1})$
$k_1 = hF\left(t_{j-1} + \frac{h}{2}, x_{j-1} + \frac{k_0}{2}\right)$
$m_1 = hG\left(t_{j-1} + \frac{h}{2}, y_{j-1} + \frac{m_0}{2}\right)$
$l_1 = hQ\left(t_{j-1} + \frac{h}{2}, z_{j-1} + \frac{l_0}{2}\right)$
$p_1 = hP\left(t_{j-1} + \frac{h}{2}, w_{j-1} + \frac{p_0}{2}\right)$
$k_2 = hF\left(t_{j-1} + \frac{h}{2}, x_{j-1} + \frac{k_1}{2}\right)$
$m_2 = hG\left(t_{j-1} + \frac{h}{2}, y_{j-1} + \frac{m_1}{2}\right)$
$l_2 = hQ\left(t_{j-1} + \frac{h}{2}, z_{j-1} + \frac{l_1}{2}\right)$
$p_2 = hP\left(t_{j-1} + \frac{h}{2}, w_{j-1} + \frac{p_1}{2}\right)$
$k_3 = hF(t_{j-1} + h, x_{j-1} + k_2)$
$m_3 = hG(t_{j-1} + h, y_{j-1} + m_2)$
$l_3 = hQ(t_{j-1} + h, z_{j-1} + l_2)$
$p_3 = hP(t_{j-1} + h, w_{j-1} + p_2)$
$x_j = x_{j-1} + \frac{1}{6}(k_3 + 2m_3 + 2l_3 + p_3)$
$y_j = y_{j-1} + \frac{1}{6}(k_3 + 2m_3 + 2l_3 + p_3)$
$z_j = z_{j-1} + \frac{1}{6}(k_3 + 2m_3 + 2l_3 + p_3)$
$w_j = w_{j-1} + \frac{1}{6}(k_3 + 2m_3 + 2l_3 + p_3)$
$t_j = t_{j-1} + h$
1
Print t, x, y, z, w
Stop

Figure 2 Flowchart of the RK-4 method for resolving continuous hyperchaotic systems model. The flowchart describes the fourth order Runge-Kutta (RK-4) numerical method used for resolving the continuous chaotic system models, where x_0, y_0, z_0, and w_0 are the initial conditions, h is the iteration step, and k_i, m_i, l_i, and p_i ($i = 0$ to 3) are the intermediate slopes.

In this work, we are interested in the hyperchaotic Lorenz system modeled as follows [34]:

$$\begin{aligned}
\dot{x} &= a(y - x),\\
\dot{y} &= x(b - z) - y + w,\\
\dot{z} &= xy - cz,\\
\dot{w} &= -fx.
\end{aligned} \tag{5}$$

In [34], it has been proven that this 4D lorenz system exhibits hyperchaotic behaviors and presents a two-dimensional bifurcation diagram for the following parameter conditions : $a = 10$, $c = 8/3$, $0 < b < 30$, and $0 < f < 15$. Therefore, the system preserves its hyperchaotic behavior and bifurcation diagram for the following considered parameter values $a = 10$, $b = 28$, $c = 8/3$, and $f = 5$ and with the initial conditions $x_0 = y_0 = z_0 = w_0 = -10$. The Matlab simulation result, using the presented RK-4 method, of the $(x - y)$ hyperchaotic Lorenz attractor is given in Figure 3.

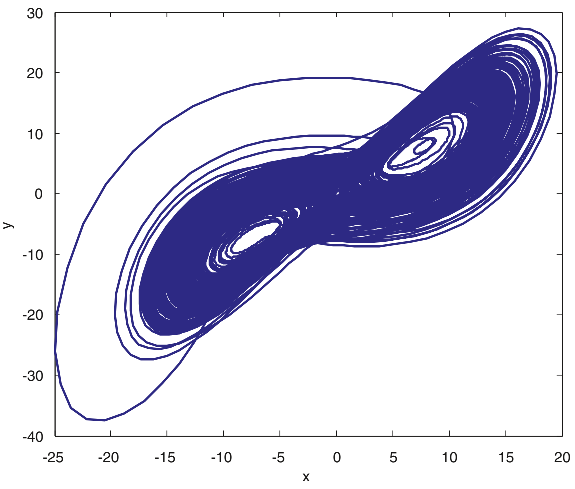

Figure 3 Matlab simulation result of the $(x - y)$ hyperchaotic Lorenz attractor.

Hardware architecture

To implement the hyperchaotic Lorenz generator, we use an optimal VHDL hardware structural description of the RK-4 method described by the flowchart in Figure 2. Indeed, contrary to the use of automatic code-based design approach which leads to non-optimal VHDL codes (for instance, the Simulink/Matlab automatic code generation tool associated to Xilinx System Generator design tool), the low-level aspect of our digital chaos implementation keeps the user very close from realities of the physical implementation (low-level architecture). Therefore, the results in terms of performance and density of resources used remain within the designer's reach. As to logic exploration architecture, one proposed RTL architecture of the 4D Lorenz hyperchaotic system (Lorenz_Generator block) is depicted in Figure 4. The four outputs S_0, S_1, S_2, and S_3 are the hyperchaotic signals, encoded on 32-bit (16Q16) fixed-point data format [27-29]. Note that the architecture depends on the four bifurcation parameters a, b, c, and f and is based on the structural feedback of the four main blocks: F, G, H, and Q. These four functional units realize the non-linear functions of the equation set (5). These units correspond to logic assignments composed of adder, subtractor, and multiplier logic arithmetic operators in accordance with the set of equations (5) and the RK-4 resolution method. This proposed hardware description also includes two

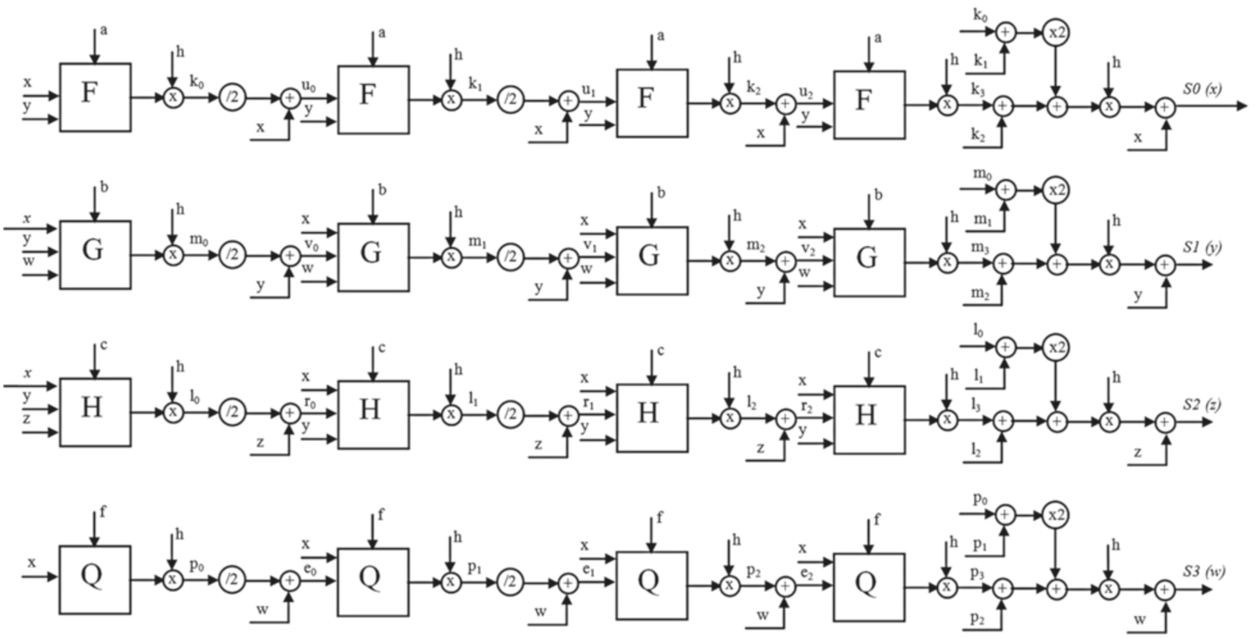

Figure 4 Architecture of the hyperchaotic Lorenz generator module. The four outputs S_0, S_1, S_2, and S_3 are the hyperchaotic signals, encoded on 32-bit (16Q16) fixed-point data format. The two inputs, clk and reset, are the clock signal and the reset signal, respectively.

inputs, clk and reset, which are the clock and reset signals, respectively. This data-path architecture is controlled by a specific finite-state machine in order that this structure performs the RK-4 resolution method. Precisely, the developed hardware architecture is based on the Moore state machine (MSM) presented in Figure 5 where ten states (ST0 to ST9) are used.

The operating principle of the developed MSM (Figure 5) is as follows:

- *ST0.* In the initial state, all outputs are initialized to 0, and the state variables are initialized by the initial condition values $x = x_0$, $y = y_0$, $z = z_0$, and $w = w_0$. The process then passes unconditionally to the next state ST1.
- *ST1.* Compute the initial slopes k_0, m_0, l_0, and p_0, and the first intermediate points u_1, v_1, r_1, and e_1 defined by the following equations (see Figure 2):

$$u_1 = x_{j-1} + k_0/2 \tag{6}$$

$$v_1 = y_{j-1} + m_0/2 \tag{7}$$

$$r_1 = z_{j-1} + l_0/2 \tag{8}$$

$$e_1 = w_{j-1} + p_0/2 \tag{9}$$

  At the next clock cycle, the machine passes unconditionally to the next state, ST2.
- *ST2.* Assign the values of the first intermediate points u_1, v_1, r_1, and e_1 to the variables α, β, γ, and θ, respectively. The use of these variables permits to optimize our architecture. Indeed, we use the same module to calculate all the slopes of the RK-4 method, and the same module is used for the calculation of the intermediate points. At the next clock cycle, the machine passes unconditionally to the next state, ST3.
- *ST3.* Compute the slopes k_1, m_1, l_1, and p_1 and the second intermediate points u_2, v_2, r_2, and e_2.

$$u_2 = x_{j-1} + k_1/2 \tag{10}$$

$$v_2 = y_{j-1} + m_1/2 \tag{11}$$

$$r_2 = z_{j-1} + l_1/2 \tag{12}$$

$$e_2 = w_{j-1} + p_1/2 \tag{13}$$

  At the next clock cycle, the machine passes unconditionally to the next state, ST4.
- *ST4, ST6.* Use the same instructions as stated in ST2; update the variables α, β, γ, and θ, and at the next clock cycle, the process passes unconditionally to the next state.
- *ST5.* Compute the slopes k_2, m_2, l_2, and p_2 and the last intermediate points u_3, v_3, r_3, and e_3 defined by the following equations:

$$u_3 = x_{j-1} + k_2 \tag{14}$$

$$v_3 = y_{j-1} + m_2 \tag{15}$$

$$r_3 = z_{j-1} + l_2 \tag{16}$$

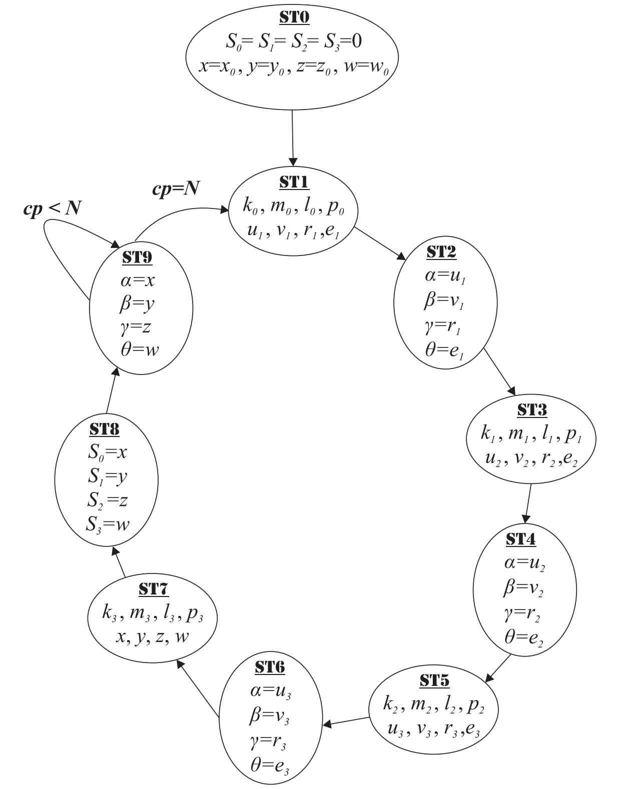

Figure 5 Moore state machine of the RK-4 hardware description. The Moore state machine (MSM) uses ten states (ST0 to ST9).

$$e_3 = w_{j-1} + p_2 \tag{17}$$

At the next clock cycle, the process passes unconditionally to the following state.

- *ST7*. Compute the slopes k_3, m_3, l_3, and p_3 and then the solutions *x*, *y*, *z*, and *w* (see Figure 2).
- *ST8*. Assign the hyperchaotic solution values *x*, *y*, *z*, and *w* to the outputs S_0, S_1, S_2, and S_3, respectively. At the next clock cycle, the process passes unconditionally to the final state, ST9.
- *ST9*. In the final state, the actual solution values *x*, *y*, *z*, and *w* are assigned to the variables α, β, γ, and θ, respectively, to compute the next hyperchaotic solution values. At the next clock cycle, if the counter value, cp, is equal to a defined integer value *N*, the process goes back again to the first state, ST1, and then the process is revived again for calculating the next solution values. Otherwise, the process stays waiting at state ST9. The value of *N* is chosen to ensure synchronization between the embedded hyperchaotic generator and an external connected device, permitting the control of the throughput of the embedded hyperchaotic generator.

Synthesis results and performance analysis

The synthesis results after the place and route of the implemented architecture on the Xilinx Virtex-II (Pro XC2VP30), Virtex V, VI, and VII FPGAs [32,37-39] are detailed in Table 1. Herein, the maximum frequency and the hardware resource's consumption in terms of slices, digital signal processing (DSP) blocks and multipliers required are specified. The results demonstrate that the proposed hyperchaotic Lorenz generator can be easy and efficiently implemented on FPGA technologies by providing real-time hyperchaotic signals and attractors. It can be stated that an attractive tradeoff between high speed and low logic resources is achieved. Indeed, our implementation on a Xilinx Virtex-II Pro device uses only 2067 CLB-Slices (15% of the size circuit), 36 multipliers (26%) and no block RAMs is used under the maximum frequency of 25.364 MHz. Similarly, our implementation on a Xilinx Virtex V device uses 4721 CLB-Slices, 20 DSP blocks under the maximum frequency of 36.271 MHz. We note that the use of DSP blocks with most recent devices improves the performance.

To evaluate the performance of the proposed hardware implementation, the throughput rate and time latency metrics are used. In our case, the throughput rate (defined as the number of bits per unit of time)

Table 1 Synthesis results

	Virtex II	Virtex V	Virtex VI	Virtex VII
	(*2vp30ff896-7*)	(*5vlx30ff676-3*)	(*6vcx75tff784-2*)	(*7vx330tffg1157-3*)
Selected device				
Slices LUTs	2,067 (15%)	4,721 (24%)	3,238 (6%)	3,238 (1%)
Slice registers	1,150 (4%)	1,105 (5%)	1,100 (1%)	1,100 (< 1%)
IOBs	130 (23%)	130 (32%)	130 (36%)	130 (21%)
MULT18X18s	36 (26%)	-	-	-
DSP48Es	-	20 (62%)	60 (20%)	60 (5%)
GCLKs	1 (6%)	1 (3%)	1 (3%)	1 (3%)
Maximum frequency (MHz)	25.364	36.271	28.507	35.842

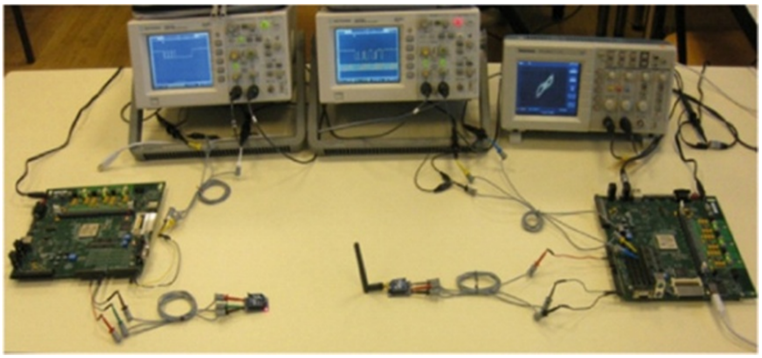

Figure 6 **Photo of the experimental device.**

corresponds to 32 bits of wordlength for ten operating clock cycles (for number of MSM states, see Figure 5) after the initialization phase at the output of the FPGA circuit.

Latency is defined as the time required to generate one single wordlength signal after the start of the generator. Therefore, a minimum and maximum throughput of 81.16 and 116 Mbps with a maximum and minimum time latency of 394.25 and 275.7 ns are obtained for Virtex II and Virtex V technologies, respectively. The results prove that our hardware architecture can be implemented in the recent FPGA devices with a significant amelioration of its performances in terms of throughput and logic area cost.

Experimental framework

To test and validate the proposed approach, we have realized the experimental framework depicted in Figure 6. For this purpose, we consider the available XUP Xilinx Virtex-II Pro development embedded platforms for physical hardware implementations [32]. The XUP System consists of a high-performance Virtex-II Pro FPGA (XCV2PFF896-7) surrounded by peripheral components that can be used to create a complex hardware system. Note that an audio CODEC (AC97) and stereo power amplifier are included on the XUP platform so as to provide complete analog functionality, allowing the external generation of chaotic signals in analog form for real measurements [40]. Both the transmitter and the receiver are mounted on a Virtex-II Pro Development System [32] connected to an XBee module based on Zigbee or Wi-Fi communication protocols [26,33]. The experimental transmission test consists of transmitting a binary information data, encoded on a 32-bit fixed-point data format, and masked (secured) by the master hyperchaotic samples (Figure 1), encoded also on a 32-bit fixed-point data format. At the receiver, we verify, after the demodulation operation (unmasking), the correct recovery of the transmitted data information. To view the result on a digital oscilloscope, a constant value is used as information data to test the feasibility of the proposed wireless communication system based on the association of the hyperchaotic communication with the ZigBee communication technology.

XBee RF modules

XBee modules offer the advantage to interface to a host device through a well-known logic-level asynchronous serial port. In fact, devices having a UART interface can connect directly to the pins of the RF modules [26,33]. In our experiment, we use the XBee modules in their transparent mode with the minimum connections VCC, GND, T_x, and R_x as showed in Figure 7. When operating in this mode, the modules act as a serial line replacement, i.e., all UART data, consisting of a start bit (low), eight data bits (least significant bit first), and a stop bit (high), received through the R_x pin queued up for RF transmission. When RF data is received, the data is sent out the T_x pin.

Transmitter architecture

The transmitter architecture implemented on the first FPGA circuit is presented in Figure 8. It is composed mainly of three modules : Clock_Generator, Lorenz_Generator, and parallel input/serial output (PISO). The details of the functioning of each module are as follows :

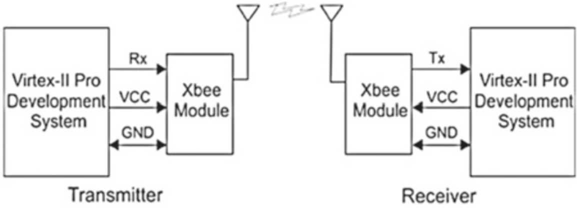
Virtex-II Pro Development System
Rx
VCC
GND
Xbee Module
Transmitter
Xbee Module
Tx
VCC
GND
Virtex-II Pro Development System
Receiver

Figure 7 **XBee modules and Virtex-II Pro development system interconnections.**

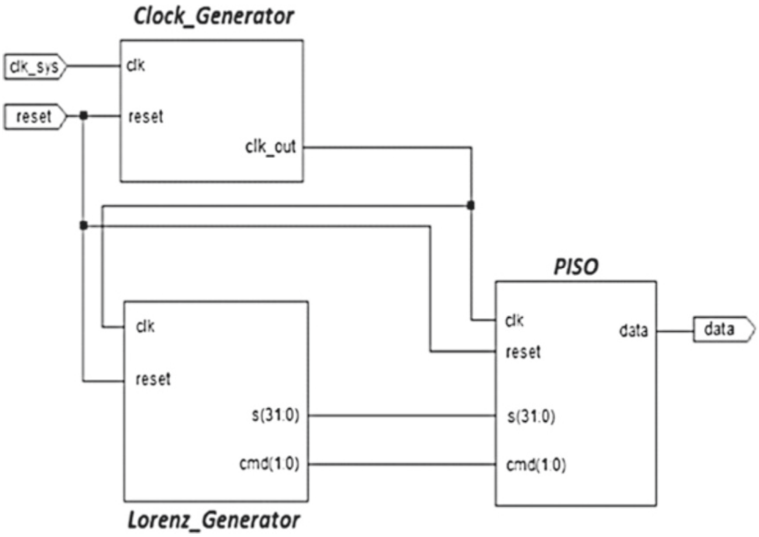

Figure 8 Transmitter architecture. The transmitter architecture is composed mainly of three modules: Clock_Generator, Lorenz_Generator, and the parallel input/serial output (PISO) modules.

- *Clock_Generator*. This module generates and provides the clock (clk_out) and reset signals to the two other modules. The frequency of the clock signal is imposed either by the serial interface data rate of the XBee modules (see Table 2) or the implemented logic blocks in the FPGA. Depending on the maximum data rate allowed, the clk_out signal frequency is defined for driving the transmitter architecture. To obtain this frequency, we have implemented a clock divider architecture to derive it from the frequency of the 100 MHz global clock signal (clk_sys) provide by the Xilinx Virtex-II Pro XC2VP30 FPGA.
- *Lorenz_Generator*. This module represents the main module of the proposed transmitter architecture. It is based on the hardware architecture already detailed and presented in the 'FPGA implementation of hyperchaotic Lorenz generator' section for implementing the hyperchaotic Lorenz system but with some modifications introduced in the MSM of Figure 5 in order to adapt with the hyperchaotic generator data output S to the XBee module data input R_x. The adaptation includes introducing a 2-bit control signal (cmd) in the hyperchaotic generator architecture which will control the operation of the parallel/serial converter module (PISO) to provide data frames according to the XBee communication protocols (Zigbee or Wi-Fi) [26,33]. Therefore, the changes in the Lorenz_Generator architecture are as follows :
At the initial state ST0, we set cmd = '11', and the process follows the same steps as in the 'FPGA implementation of hyperchaotic Lorenz generator' section until reaching state ST7. However, to realize our solution, we have added an 11th state to the MSM, and then from the state ST8, the next steps are as follows:

  - *ST8*. We realize the modulation operation using additive chaos masking of [10] and [36], i.e., the information signal samples d are masked by those of the hyperchaotic signal x and then the

Table 2 Wireless performance and data or image transmission results

Protocol	Zigbee (WPAN)	Wi-Fi (WLAN)
Maximum distance (indoor/urban range)	Up to 100 ft (30 m)	Up to 120 ft (32 m)
Transmit power output of Xbee modules	1 mW (0 dBm)	>15 dBm
Maximum data bit rate (modulation rate)	250 Kbps (6,250 baud)	25.364 Mbps (634,100 baud)
Maximum RF data rate	250 Kbps	54 Mbps
Supply voltage	2.8 to 3.4 V	3.1 to 3.6 V
Operating frequency modulation	ISM 2.4 GHz	ISM 2.4 to 2.5 GHz

Table 2 gives the obtained wireless performance of the proposed system based on Virtex II FPGA and Xbee RF modules.

result is assigned to the output of the module $S_0 = x + d$. At the same time, we put cmd ='10', and the process passes unconditionally to state ST9.

- *ST9*. At this state, we realize the dynamic feedback operation, used by the FHS-DFM technique, by assigning the additive hyperchaos masking results S_0 to the variable α. Hence, the actual sample S_0 is injected to the master dynamics, and it is used to generate the next hyperchaotic sample x. We put cmd ='00', and the passage to the next state, ST10, is controlled by the parameter value T of the counter ct. More details are given in the next paragraph.
- *ST10*. This tenth state is added to control the eventual distance between two successive data frames which can imposed by the XBee communication protocols [26,33]. This is ensured by the parameter value N of the counter cp as shown in Figure 9. In this last state, cmd is set to the value 11.

- *PISO*. This module is a binary parallel/serial converter. Under the command signal cmd, the module converts the parallel data samples S_0 (coded in 32 bits), received from the Lorenz_Generator module after the modulation operation, to a serial data. The command signal values are presented in Table 3. The transmitted data frame is formed by four successive data frames of 8 bits that started and ended by a start bit ('0') and a stop bit ('1'), respectively. Therefore, the data frame wordlength is $T = 40$ bits. Note that this data format is imposed by the XBee RF module's communication protocols [26,33].

Table 3 PISO commands

Value	Command
10	Form the serial data frame
00	Start serial transmission
11	Wait and put the line at the high level

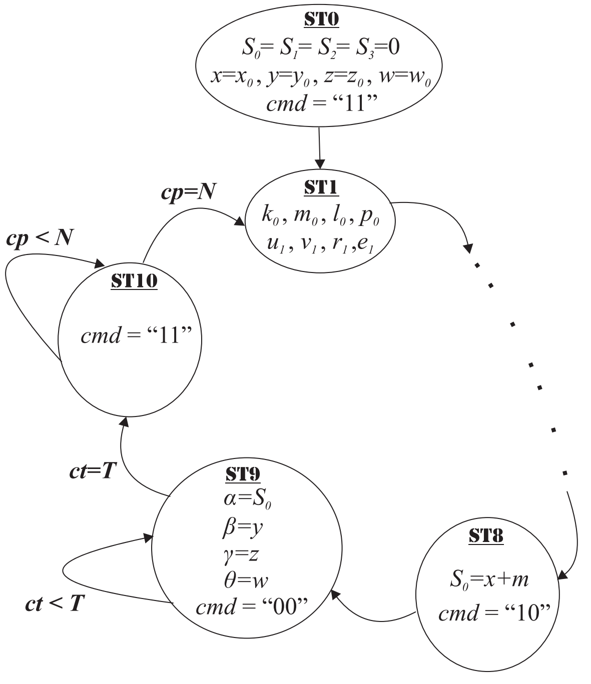

Figure 9 Lorenz_Generator state machine at the transmitter.

Receiver architecture

The receiver architecture is presented in Figure 10. It is composed of four modules : Clock_Generator, Lorenz_Generator, PISO, and serial input/parallel output (SIPO). The first three modules are similar to those of the transmitter but with an adjustment to the receiver. The details of the functioning of each module are as follows:

- *SIPO*. This module is a binary serial-to-parallel converter. Once the start bit is detected at the receiver by the SIPO module, the serial/parallel conversion of the received data begins. At the same time, the module generates a clock signal (clk_lz) at the same frequency as the clk_out clock signal generated by the Clock_Generator module. This means that the generation of the clk_lz clock signal is triggered at each start bit detection. This constitutes our interesting solution to overcome the problem of shifting data frames at the XBee RF module output (T_x). Indeed, this solution permits to adapt and to synchronize the execution of the implemented receiver architecture to the cadence of the XBee RF module data output (T_x). However, to synchronize the SIPO and the Lorenz_Generator modules, we use a 1-bit command signal (cm). Initially, this last one is set to '0', and it is set to '1' during one clock period only when the converted 32-bit parallel data S_r are available at the output of the SIPO module, i.e., the serial-to-parallel conversion is finished.
- *Lorenz_Generator*. This module is similar to the hyperchaotic Lorenz generator used at the transmitter. This means that it generates the hyperchaotic keys with the same values of the parameters and initial conditions at the transmitter in order to allow the hyperchaotic synchronization as proposed in the 'Hyperchaotic synchronization and encryption technique' section. Under the command signal cm and the rhythm of the clk_lz clock signal, the module regenerates identical hyperchaotic samples (x_r) to that of the transmitter. It

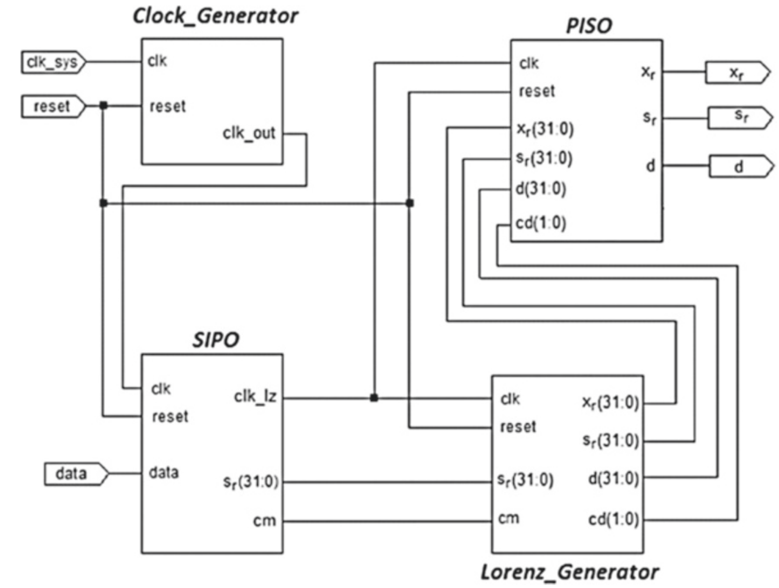

Figure 10 Receiver architecture.

synchronizes them with the received and parallel converted data samples S_r and recovers the information data d after a demodulation operation based on the arithmetic subtraction. Finally, it controls the parallel-to-serial conversion operation of the PISO module using the 2-bit command signal cd. The cd commands are the same as those of cmd command signal presented in Table 3. As the transmitter architecture, the Lorenz_Generator module uses the same state machine presented in Figure 5, but with the modifications introduced and presented in Figure 11.

At the first clk_lz clock cycle, the module starts the execution of the state machine instructions from ST0 to ST7. Therefore,

- At the synchronization step ST8, we assign the generated hyperchaotic samples x to the output x_r and the received and parallel converted data S_r to the output S_r (Figure 11). Note that we have used the same parameter name S_r for the parallel converted data at the input and the output of the Lorenz_Generator module because no change is made for this data at this module. Therefore, if the command signal cm is set to 1, i.e., the received data S_r are available at the output of the SIPO module, then the process passes to the next state, ST9. Otherwise, it stays waiting at the current state.
- At the demodulation step ST9, after the synchronization of the x_r and S_r samples at the previous state, the information data d is

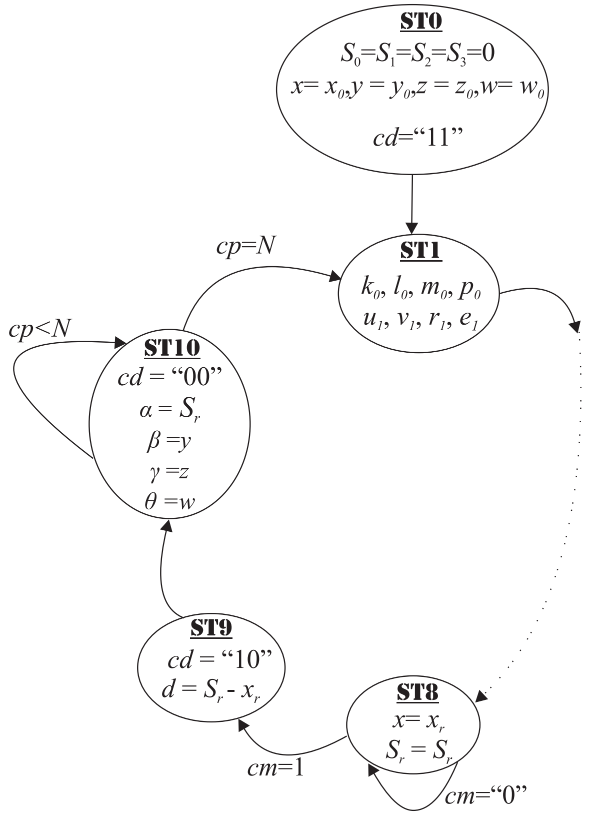

Figure 11 Lorenz_Generator state machine at the receiver.

recovered using the subtraction arithmetic operation $d = S_r - x_r$, and then the command signal cd takes the value 10. Finally, the process passes unconditionally to the next state, ST10.

- At the last state ST10, the command signal cd takes the value 00, and the received data sample S_r is assigned to the parameter α, which is used to generate the next hyperchaotic sample x, instead of the actual generated hyperchaotic sample x. This instruction permits to realize the hyperchaotic synchronization principle presented in Figure 6, in which the received signal $S(t)$ is injected to the dynamic of the hyperchaotic generator of the receiver. Finally, after N clock cycles, the process passes to state ST1. The value of N is fixed by the time needed by the PISO module to finish the parallel-to-serial conversion of the hyperchaotic samples x_r. Then, in our experiments, N = 33 clock cycles, considering that the samples x_r are encoded on 32 bits.

- *PISO*. This module is similar to that in the transmitter block. Under the command signal cd, it converts the 32-bit parallel outputs of the Lorenz_Generator module, i.e., the regenerated hyperchaotic samples x_r, the received data samples S_r, and the recovered data information d, to 32-bit serial frames. This module is introduced in the architecture just for validating the proposed approach by real-time viewing and comparing these serial data frames on a digital oscilloscope.

Wireless real-time data or image transmission tests and results

The main considered application is the security of wireless sensor networks (WSNs) which are becoming more and more important, and they can gain advantage of reconfigurable technology, in terms of flexibility, energy consumption, and sensor lifetime. This is particularly true for the networks of data diffusion based on embedded systems, which can be used for the protocol communication. Indeed, a WSN provides different aspects in the sharing of information by deploying a system that is able to execute wireless exchange of data, image, or video [41] according to transmission rate performance. Subsequently, we considered the wireless data or image transmission with the Zigbee and WiFi protocols in a WSN context.

The ciphered data or image is transmitted through a public and unsecure channel. Using self-synchronization technique at the receiver side [32], the key can be recovered at the receiver and the decryption operation from the transmitted scrambled image with the regenerated key, allowing us to recover the plain image.

In the experimental transmission tests, we use binary data encoded on 32-bit fixed-point format with hexadecimal representation as information signal. At the transmitter, these information data are masked by the hyperchaotic samples x of the Lorenz_generator encoded also on 32-bit fixed-point data format. The encrypted signal samples $S_i(t)$ are then converted to serial data format by the PISO module, sent to its associate XBee Pro RF module according to the considered protocol (Zigbee or Wi-Fi), and then transmitted to the receiver. At the receiver, the associated XBee Pro RF module transmits the received data to the SIPO module according to the asynchronous serial communication protocol to regenerate the received encrypted signal sample format $S_r(t)$, allowing for the hyperchaotic synchronization and recovery of the masked information $d(t)$ according the proposed scheme depicted in Figure 1.

Table 2 summarizes experimentally the performance and results in terms of digital transmission rate (symbol rate or modulation rate), maximum distance, and frequency modulation according to the considered wireless protocols by the Xbee RF modules and Virtex II technology. For these measurements, we have placed the transmitter and the receiver at two neighboring rooms at the distance about 20 m with a received signal strength indicator (RSSI) of −2 dBm. With this disposition, we obtain a packet error rate (PER) of 0% at the receiver. The maximum distance that we can obtain experimentally between the transmitter and the receiver (indoor range) is more than 30 and 32 m for Zigbee and Wi-Fi protocols, respectively. Therefore, the indoor/urban ranges of the XBee RF modules used are up to 30 m (see Table 2), and the sensitivity of the XBee RF module receivers reaches −92 dBm with a PER of 1% [33]. The XBee modules offer the advantage to realize a wireless communication application without errors (PER = 0%) according to environment application, distance, disposition, and channel chosen as is shown in [42].

The maximum bit rate of the proposed system is limited either by the RF modules or implemented hardware logic blocks. Indeed, the hardware FPGA implementations allow parallel/serial and serial/parallel converters with minimum and maximum rates of 25.36 and 36.27 Mbps obtained with Virtex II and Virtex V technologies, respectively (see Table 1). However, for the considered Zigbee protocol, the maximum bit rate of the proposed system is limited by the RF modules. The hardware FPGA implementation performance (for working clock frequency, see Table 1) of the proposed system (at the transmitter and receiver) is considered, and it is larger than the serial interface data rate and better than the bit rate, which is allowed by the considered Zigbee RF modules. Consequently, the limitation is imposed by the transmission rate of the parallel/serial and serial/parallel converters toward Zigbee

Xbee RF modules while FPGA implementations allow to provide transmission rates to at least 25 Mbps. Thereby, for a null distance frame value ($N = 0$), the obtained data bit rate of the serial communication is 250 kbps (corresponding to a modulation rate of 625 symbols per second or baud, due to the maximum serial interface data rate of the Zigbee protocol based XBee Pro modules [33], and with an operating frequency modulation of 2.4 GHz. In the case of the Wi-Fi protocol, the maximum bit rate of the proposed system is limited by the work frequency of the implemented hardware modules. Although the FPGA implementation of the Lorenz generator allows to provide a throughput of 80 Mbps (maximal frequency of 25 MHz is allowed by the considered Virtex II platform with an encoded 32-bit encrypted data), the maximum transmission bit rate of the proposed system is limited by the hyperchaotic key generators up to 25 Mbps (for $N = 0$) and by a corresponding modulation rate of 625,000 baud. Therefore, the parallel/serial and serial/parallel converters to Wi-Fi Xbee RF modules limit the transmission rate up to the maximum work frequency, depending on the considered FPGA technology. Indeed, each symbol of the data transmission system carries 40 bits according to the data frame wordlength allowed by the serial interface Xbee modules. This digital modulation rate operates with a frequency modulation range between 2.4 and 2.5 GHz [26]. In summary, considering a synchronous system at the maximum work frequency allowed between the key stream generators and parallel/serial or serial/parallel converters, the limitations with respect to 54-Mbps and 25-kbps bit rates of the Wi-Fi and Zigbee RF modules are due to the work frequencies of hyperchaotic Lorenz generators and the serial interface data rate, respectively.

An example of real-time data results obtained for a constant value of '00009999' is shown in Figures 12 and 13. These figures give snapshots of the real-time

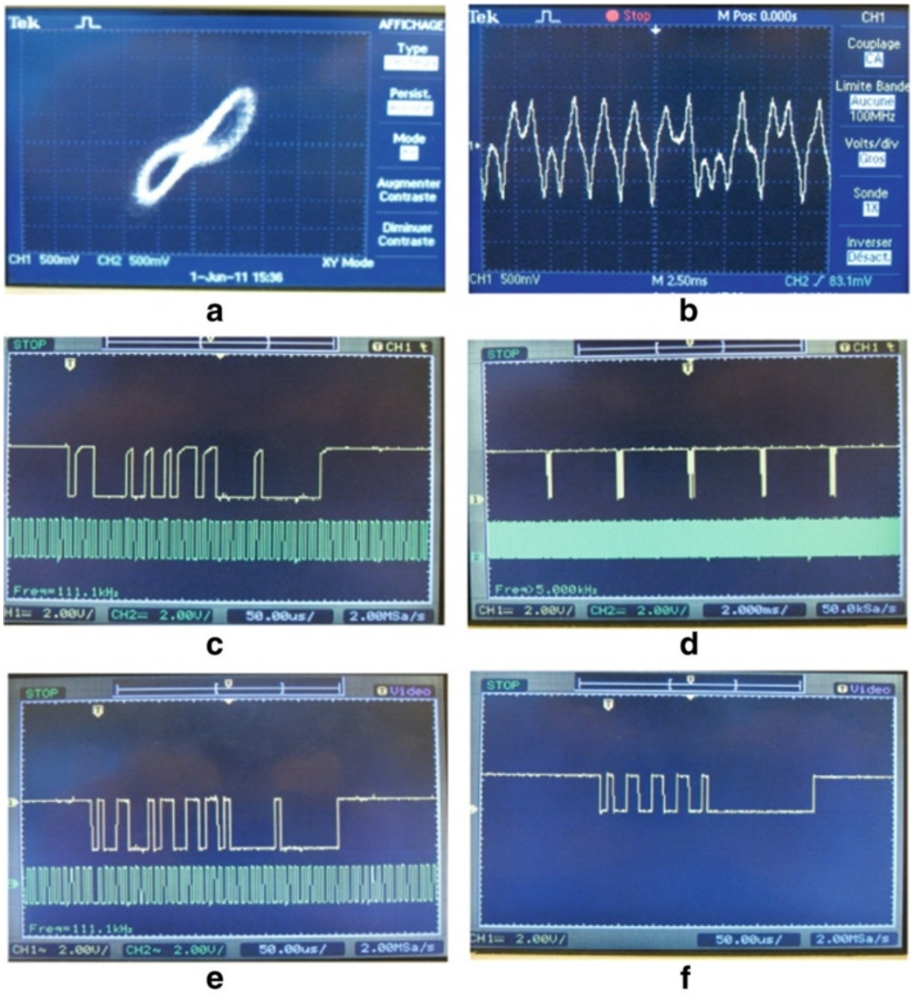

Figure 12 Real-time results of the transmission test. (a) ($x - y$) hyperchaotic attractor, **(b)** hyperchaotic drive signal x, **(c)** transmitted serial data frame and clock signal, **(d)** distances between transmitted data frames, **(e)** received data frame at the output of the XBee module (receiver), and **(f)** recovered information data.

digital modulation in the case of the transmission data rate of 250 kbps (the maximum serial interface data rate of the XBee Pro modules [18] allowed with a clk_out signal frequency equal to 250 kHz) with a distance frame value of $N = 500$ clock cycles, allowing signal captures. In Figure 12, we present the results of the transmission test for RSSI at -2 dBm with PER $=$ 0%, and Figure 13 presents the transmission test results for RSSI at -70 dBm with PER $\approx$ 1%. Figure 12a,b presents the $(x - y)$ hyperchaotic attractor and the hyperchaotic carrier signal x, respectively. An example of the transmitted serial masked data frames with the corresponding clock signal is presented in Figure 12c. Figure 12d shows the distance between the transmitted data frames, and Figure 12e presents an example of the received data frame at the output of the XBee module of the receiver with the corresponding clock signal clk_lz. Finally, the recovered information data (only for the constant value 00009999) is shown in Figure 12f. Note that the transmitted masked data frame presented in Figure 12c shows the robustness security of the information data, in the proposed wireless communication system, by the additive hyperchaos masking principle. In fact, the information data are totally hidden by the hyperchaotic ones. The results of Figure 12e validate our proposed solution for adapting and synchronizing the implemented receiver architecture and the XBee RF module. Indeed, from the results, we note that the clock signal clk_lz is triggered at the start bit detection of the received data frame. Finally, the results presented in Figure 12f validate the principle of the proposed wireless hyperchaotic communication system and then the relevant idea based on associating the hyperchaos communication principle with the ZigBee technology. In fact, from this figure, we see that the information data are recovered correctly without any error at a distance of 20 m because PER $=$ 0%.

In Figure 13, we present the first three successive recovered information data frames for an RSSI of -70 dBm and a PER of about 1%. These results show that the information data are totally lost because of the PER of 1%. This confirms the extreme sensibility of chaotic synchronization to small channel perturbation.

For secure real-time video transmission in the WSN, we consider an encrypted video transmission rate of 25 images per second with a spatial resolution of 128x128 gray level pixels coded at 8 bits. Each pixel value is extended to 40 bits according to the wordlength data frame format of the encryption process and serial transmission [26,33]. Consequently, we deduce a time constraint of 40 ms by image to assure that the real-time wireless transmission rate corresponds to a minimum bit rate of 16,384 kbps or modulation rate of 409,600 baud. Therefore, the proposed system based on Wi-Fi XBee modules, which can be performed up to the maximum modulation rate of 625,000 baud, achieves a wireless real-time encrypted transmission suitable for the WSN context. Figure 14 gives a snapshot of a wireless real-time image transmission, showing the feasibility and efficiency of the proposed digital encryption modulation system.

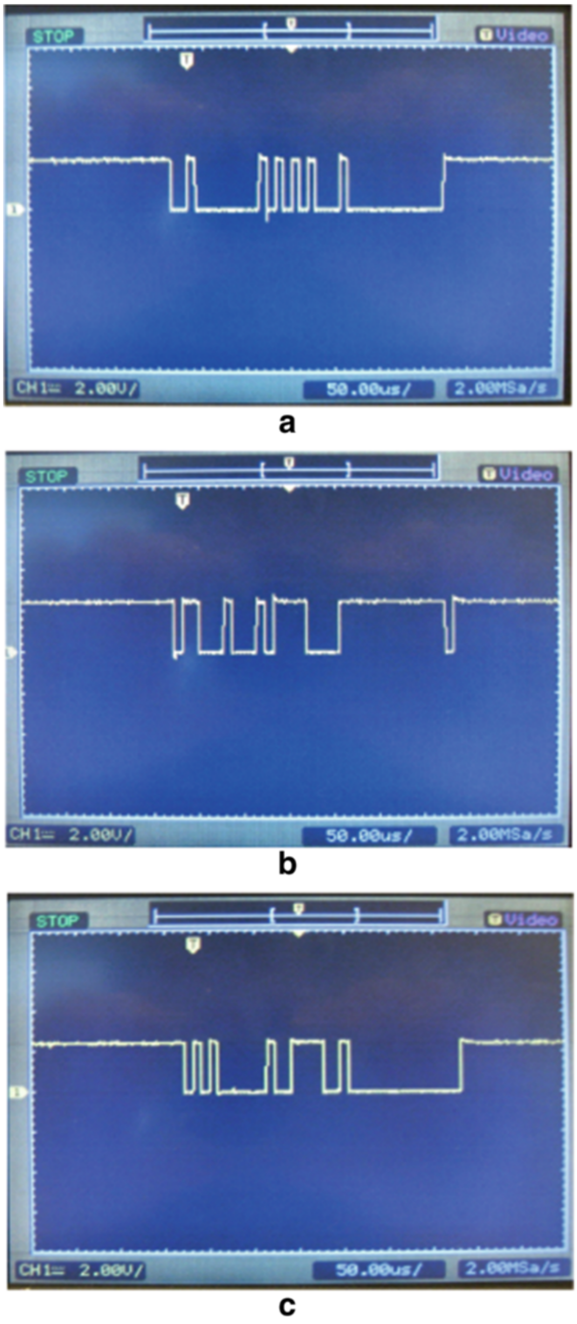

Figure 13 Oscilloscope snapshots of three successive recovered information data frames for RSSI = −70 dBm and PER ≈ 1% (a, b, c). The experimental measurements of the data frames have been done by a digital oscillocope with a timebase of 50 μs, a sensibility of 2 Volts-per-Division and a sample rate /channel of 2 MegaSAmples per second (MSA/s).

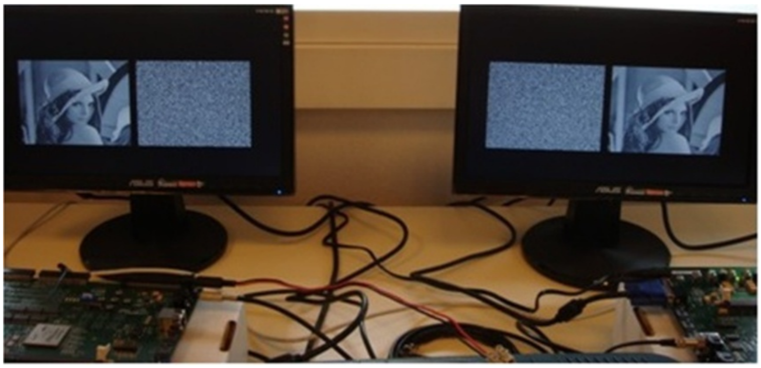

Figure 14 Snapshot of a wireless real-time image transmission based on the Wi-Fi protocol.

Security analysis

To test the robustness of the proposed scheme, security analysis including statistical analysis and differential analysis was performed. This evaluation of the quality of randomness is carried out to demonstrate the satisfactory security of the new proposed chaos-based cryptosystem.

NIST statistical analysis

A large number of statistical tests and whole test suites have been proposed to assess the statistical properties of random number generations. Statistical tests of the generated 128-bit encryption keys are commonly performed using the standard NIST SP 800-22 statistical test suite [43]. In this subsection, we present the performance test results of the proposed chaos-based key stream generator. Eighteen statistical tests are commonly used to determine whether one binary sequence possesses some specific characteristics that a truly random sequence would be likely to exhibit. Table 4 summarizes the results of NIST tests. Each one was performed 300 times on 1-Mbit substrings. A single test is considered as passed if the P value is above the significant level of 0.01 or below 0.99 [43]. The results of Table 4 show the measured values of P value T, knowing that if P value $T \geq 0.0001$, then the sequences can be considered to be uniformly distributed. Similarly, the minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately 0.972766 for a sample size equal to 300 binary sequences (for more details, see the reference [43]). Figure 15 gives the Matlab simulation result of the $(x - y - z)$ phase space and confirms these statistical results. Indeed, this phase space is occupied by random trajectories proving that the sequences are really uniformly distributed. Consequently, we can conclude that the proposed chaos-based key stream generator successfully passes all the NIST tests and provides good randomness keys. Therefore, these results demonstrate that the proposed chaos-based key stream is very useful for the consideration of reducing negative dynamic degradation influence due to the finite precision of the digital hardware implementation.

Histogram and differential analysis

To demonstrate the effectiveness of confusion and diffusion proprieties, a histogram test is carried out and shown. Since an image histogram illustrates how pixels in an image are distributed by plotting the number

Table 4 NIST tests results

Statistical tests	*P* value *T*	Proportion
Frequency	0.644060	0.9800
Block frequency	0.023545	0.9833
Cumulative sum up	0.671779	0.9900
Cumulative sum down	0.699313	0.9933
Runs	0.117661	0.9933
Longest run	0.630178	0.9967
Rank	0.840081	0.9867
Discrete Fourier transform	0.100508	1.0000
Non-overlapping templates	0.588652	0.9833
Overlapping templates	0.547637	1.0000
Universal	0.568055	0.9833
Approximate entropy	0.162606	0.9933
Random excursions	0.249991	0.9947
Random excursion variant	0.711827	0.9840
Serial 1	0.706149	0.9833
Serial 2	0.110952	0.9833
Lempel-Ziv value	0.071177	0.9900
Linear complexity	0.487885	0.9967

Table 4 gives the results of all statistical tests for the considered sequence.

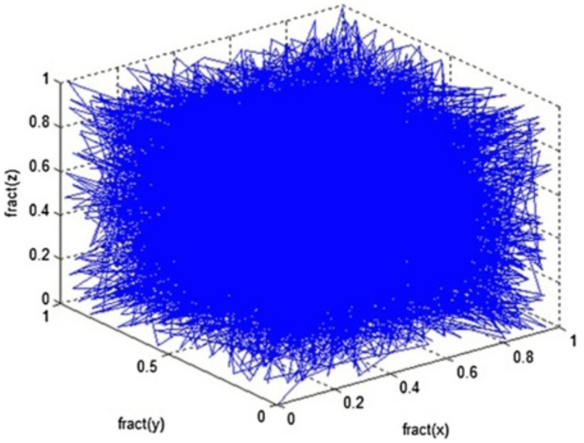

Figure 15 Matlab simulation result of ($x - y - z$) phase space showing that encryption key sequences are really uniformly distributed.

of pixels at each gray-scale intensity level. By selecting several 256 gray-scale images with a resolution of 256x256 with different image contents, their histograms were calculated. The typical example (cameramen image) among them is shown in Figure 16. The obtained histogram of the ciphered image is fairly uniform and significantly different from that of the original image showing then the sensitivity to the plain image.

To avoid the known-plaintext attack and the chosen-plaintext attack, the changes in the ciphered image should be significant even with a small change in the original one. According to the proposed encryption process, this small difference should be diffused to the whole ciphered data, with respect to diffusion and confusion. Consequently, this differential attack would become very inefficient and practically useless. Generally, the differential analysis can be reflected by the number of pixels' change rate (NPCR) and unified average changing intensity (UACI) evaluations [44]. NPCR stands for the number of pixels' change rate while one pixel of plain image changed. The more NPCR gets close to 100%, the more sensitive the cryptosystem to the changing of plain image is and then the more effective for the cryptosystem to resist plaintext attack. UACI measures the average intensity of differences between two encrypted images. Currently, the bigger the UACI, the more effective is the cryptosystem to resist at a differential attack.

The plain cameraman image is used in the test. The first image is the original plain image, and the second is obtained by changing the first pixel gray-scale value (for the cameraman image, the change was from '28' to '29'). Therefore, the two images are encrypted with the same key to generate the corresponding cipher images C1 and C2 (encrypted images before and after one pixel of the plain image is changed). We obtain the results by simulating experiments which are shown in Table 5. We can find that the NPCR and the UACI are over 99% and 33%, respectively. These results show that the proposed encrypted algorithm is very sensitive to tiny changes in the plain image, even if there is only one different bit between the two plain images. Consequently, the decrypted images will be strongly different, showing the robustness of the proposed encryption scheme against one differential attack.

In summary, this security analysis proves that encrypted and synchronized generated signal is non-periodic and has a flat spectrum which is suitable for encryption image scheme by showing a robustness against plaintext attacks.

Conclusions

This paper proposes an experimental demonstration of a wireless hyperchaotic communication based on wireless communication protocols suitable for secure real-time data or image transmissions in wireless sensor networks. We have proposed a new digital synchronized modulation based on FHS through a DFM technique between two hyperchaotic generators. The choice of a DFM principle for implementing the FHS between two identical hyperchaotic systems of Lorenz in FPGA shows more robustness than the classic chaotic masking while allowing high transmission rates. In practice, we have associated and adapted the hyperchaotic communication principle with the XBee RF modules by developing a reconfigurable VHDL-based hardware architecture implemented on FPGA technology. Indeed, the proposed system can be used as hard key generator in a hyperchaotic synchronized data or image stream cipher/decipher, and it can be used for synchronizing any four-dimensional continuous chaotic system (such as hyperchaotic Lorenz system) where the master chaotic system is embedded

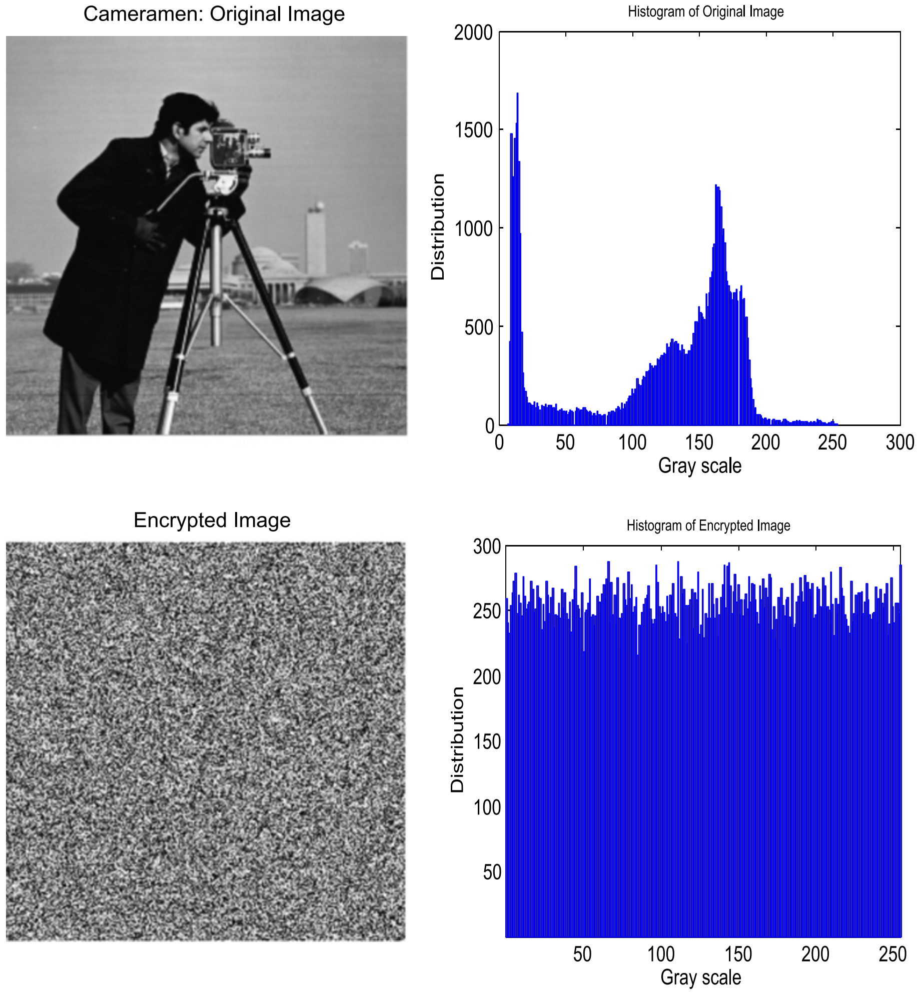

Figure 16 Test of the distribution (histogram) between adjacent pixels in a ciphered image.

in the proposed FPGA transmitter side and the slave chaotic system is embedded in the FPGA receiver side. Many real-time transmission tests are carried out between two distanced Virtex II-Pro Xilinx FPGA platforms. The obtained real-time results show the efficiency of the proposed idea consisting on associating the hyperchaotic communication, and the ZigBee or Wi-Fi communication protocols characterized by high immunity against channel noise. Indeed, we could recover correctly the information data on the distance about 20 m using the XBee RF modules at −2 dBm with a PER of 0%. Note that these performances can be improved using the most recent XBee modules. Experimental results applied to image encryption have demonstrated that our approach exhibits attractive performances and is useful in the field of real-time secure wireless data communications. The proposed technique may make it more applicable in such field (video, image, internet, etc.) and for all type of wireless network. Indeed, thorough experimental tests have been carried out with detailed numerical analysis, demonstrating the high security of the new data or image encryption scheme. More precisely, the proposed approach used to design a secure symmetric image encryption increases its resistance to various attacks such as statistical and key analysis attacks and can avoid the hidden security attacks in real-time applications. Finally, our perspective for the presented work consists on developing and realizing a secure wireless hyperchaotic communication network using the proposed modulation system.

Table 5 Results of NPCR and UACI tests

Plain image	NPCR (%)	UACI (%)
Cameraman	99.586	34.77

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Laboratoire Systèmes de Communications, Ecole Militaire Polytechnique, Algiers, Algeria. [2]Équipe ASEC - Laboratoire Conception, Optimisation et Modélisation des Systèmes, Université de Lorraine, Metz, France.

Received: 29 January 2013 Accepted: 26 June 2013
Published: 29 July 2013

References

1. GA Spanos, TB Maples, Performance study of a selective encryption scheme for the security of networked, real-time video, in *Proceedings of the 4th International Conference on Computer Communications and Networks, Las Vegas, 20-23 Sept 1995* (IEEE, Piscataway, 1995), pp. 2–10
2. T Yang, A survey of chaotic secure communication systems. Int. J. Comput. Cogn. **2**(2), 81–130 (2004)
3. B Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd edn (Wiley & Sons, New York, 1996)
4. J Zambreno, D Nguyen, AN Choudhary, Exploring area/delay tradeoffs in an AES FPGA implementation, in *Field Programmable Logic and Applications,* ed. by J Becker, M Platzner, S Vernalde. Proceedings of the 14th International Conference, FPL 2004, Leuven, 30 August-1 September 2004. Lecture Notes in Computer Science, vol. 3203. (Springer, Heidelberg, 2004), pp. 575–585
5. X Yi, CH Tan, K SC, MR Syed, Fast encryption for multimedia. IEEE Trans. Consum. Electron. **47**(1), 101–107 (2001)
6. MP Kennedy, G Kolumban, Digital communication using chaos. Signal Process. **80**, 1307–1320 (2000)
7. G Alvarez, S Li, Some basic cryptographic requirements for chaos-based cryptosystems. Int. J. Bifurcation Chaos. **44**, 2129–2151 (2006)
8. JS Lin, TL Liao, CF Huang, J Yan, Design and implementation of digital secure communication based on synchronized chaotic systems. Digit. Signal Process. **20**, 229–237 (2010)
9. W Chang, Digital secure communication via chaotic systems. Digit. Signal Process. **19**, 693–699 (2008)
10. KM Cuomo, AV Oppenheim, SHS Trogatz, Synchronization of Lorenz-based chaotic circuits with applications to communications. IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process. **40**(10), 626–633 (1993)
11. H Dedieu, MP Kennedy, M Hasler, Chaos shift keying: modulation and demodulation of a chaotic carrier using self-synchronizing Chua's circuits. IEEE Trans. Circuits Syst. II: (Special Issue on Chaos in Nonlinear Electronic Circuits- Part C: Applications). **40**, 634–642 (1993)
12. KS Halle, WC Wah, M Itoh, LO Chua, Spread spectrum communication through modulation of chaos. Int. J. Bifurcation Chaos. **3**, 469–477 (1993)
13. S Hayes, C Grebogi, E Ott, Communicating with chaos. Phys. Rev. Lett. **70**, 3031–3034 (1993)
14. YC Lai, E Bollt, C Grebogi, Communicating with chaos using two-dimensional symbolic dynamics. Phys. Lett. A. **255**, 75–81 (1999)
15. U Feldmann, M Hasler, W Schwarz, Communication by chaotic signals: the inverse system approach. Int. J. Circuit Theory Appl. **24**, 551–579 (1996)
16. T Yang, A survey of chaotic secure communication systems. Int. J. Comput. Cogn. **2**(2), 81–130 (2004)
17. I Grosu, E Padmanaban, PK Roy, SK Dana, Designing coupling for synchronization and amplification of chaos. Phy. Rev. Lett. **100**, 234102 (2008)
18. M Eisencraft, AM Batista, Discrete-time chaotic systems synchronization performance under additive noise. IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process. **91**, 2127–2131 (2011)
19. J Schweizer, T Schimming, Symbolic dynamics for processing chaotic signals-I: noise reduction of chaotic sequences. IEEE Trans. Circuits Syst. I. **48**, 1269–1282 (2001)
20. M Ciftci, DB Williams, Optimal estimation sequential channel equalization algorithms for chaotic communications systems. EURASIP J. Appl. Signal Process. **4**, 249–256 (2001)
21. X Wang, Z Wang, A robust demodulation application communication using chaotic signals. Int. J. Bifurcation Chaos. **13**, 227–231 (2003)
22. K Murali, Heterogeneous chaotic systems based cryptography. Phys. Lett. A. **272**, 184–192 (2000)
23. S Li, G Alvarez, G Chen, X Mou, Breaking a chaos-noise-based secure communication scheme. Chaos. **15**, 013703 (2005)
24. S Sadoudi, C Tanougast, MS Azzaz, First experimental solution for channel noise sensibility in digital chaotic communications. Prog. Electromagnetics Res. C. **32**, 181–196 (2012)
25. SC Ergen, *ZigBee/IEEE 802.15.4 Summary* (IEEE, Piscataway, 2004)
26. Digi International Inc, *XBee Wi-Fi RF module, Product Manual v1.xEx - 802.11 Protocol* (Digi International Inc., Minnetonka, 2011)
27. S Sadoudi, C Tanougast, MS Azzaz, A Dandache, A Bouridane, Embedded Genesio-Tesi chaotic generator for ciphering communications. 7th International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP), Newcastle, 21-23 July 2010 (IEEE, Piscataway, 2010), pp. 234–238
28. S Sadoudi, C Tanougast, MS Azzaz, A Dandache, A Bouridane, Real-time FPGA implementation of Lü's chaotic generator for cipher embedded system. International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, 9-10 July 2009 (IEEE, Piscataway, 2009), pp. 1–4
29. S Sadoudi, MS Azzaz, M Djeddou, M Bensalah, An FPGA real-time implementation of the Chen's chaotic system for chaotic communications. Int. J. Nonlinear Sci. **7**(4), 467–474 (2009)
30. MI Sobhy, MA Aseeri, AER Shehata, Real time implementation of continuous (Chua and Lorenz) chaotic generator models using digital hardware, in *Proceedings of the Third International Symposium on Communication System Networks and Digital Processing (CSNDSP),* Staffordshire, 15-17 July 2002 (IEEE, Piscataway, 2002), pp. 38–41
31. LS Indrusiak, ECDS Junior, M Glesner, Advantages of the Linz-Sprott weak nonlinearity on the FPGA implementation of chaotic systems: a comparative analysis. Proceedings of the Int. Symp. Signals, Circuits and Sys. **2**, 753–756 (2005)
32. Xilinx Inc., *Xilinx University Program Virtex-II Pro Development System, UG69 (v1.1)* (Xilinx Inc., San Jose, 2008)
33. Digi International Inc, *Product Manual v1.xEx - 802.15.4 Protocol* (Digi International Inc., Minnetonka, 2009)
34. R Barboza, Dynamics of a hyperchaotic Lorenz system. Int. J. Bifurcation Chaos. **17**(12), 4285–4294 (2007)
35. V Milanovic, ME Zaghloul, Improved masking algorithm for chaotic communications systems. Elec. Lett. **32**, 11–12 (1996)
36. KML Kocarev, KS Halle, K Eckert, LO Chua, Experimental demonstration of secure communications via chaotic synchronization. Int. J. Bifurcation Chaos. **2**, 709–713 (1992)
37. Xilinx Inc, *Virtex 5 FPGAs datasheet* (Xilinx Inc., San Jose, 2008)
38. Xilinx Inc, *Virtex 6 series fpgas configurable logic block* (Xilinx Inc., San Jose, 2012)
39. Xilinx Inc, *Virtex 7 FPGAs datasheet* (Xilinx Inc., San Jose, 2013)
40. Analog Devices, *AC'97 SoundMAX Codec, AD1881A datasheet* (Analog Devices, Norwood, 2000)
41. J Yick, B Mukherjee, D Ghosal, Wireless sensor network survey. Comput. Netw. **52**(12), 2292–2330 (2008)
42. A Centeno, N Alford, Measurement of ZigBee wireless communications in mode-stirred and mode-tuned reverberation chamber. Prog. Electromagnetics Res. M. **18**, 171–178 (2011)
43. A Rukhin, J Soto, J Nechvatal, M Smid, E Barker, S Leigh, M Levenson, M Vangel, D Banks, A Heckert, J Dray, S Vo, A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, NIST Spec. Publication 800-22 Revision 1a, NIST, Gaithersburg, 2010
44. Y Wu, J Noonan, S Agaian, NPCR and UACI randomness tests for image encryption. Cyber Journals: Multidisciplinary Journals in Science and Technology. J. Selected Areas Telecommunications. **1**(4), April 2011 Edition, 31–38 (2011)

doi:10.1186/1687-5281-2013-43
Cite this article as: Sadoudi *et al.*: **Design and FPGA implementation of a wireless hyperchaotic communication system for secure real-time image transmission.** *EURASIP Journal on Image and Video Processing* 2013 **2013**:43.