Synchronization of Two Chaotic Stream Ciphers in Secure CDMA Communication Systems

Ahmed S. Alshammari  
Department of Electrical Engineering  
University of Hail  
Hail, Saudi Arabia  
ahm.alshammari@uoh.edu.sa

Abstract—In the basic processing of Code Division Multiple Access (CDMA) systems using chaotic code sequences, a pair of chaos generators in the transmitter and the receiver are used to generate the same sequence, in order to synchronize and retrieve the transmitted signals. In addition, a filter with a simple structure should be integrated into chaotic signals to achieve the maximum of the signal to noise ratio and mitigate the harmful effects of multipath. Another effective property of chaos signal is that a wireless multipath channel does not change the amount of contained information. Considering those issues, the present paper describes a practical system for synchronizing chaotic signals in secure CDMA communication systems.

Keywords—synchronization; CDMA; chaotic; communication systems; FPGA

I. INTRODUCTION

During the last years, CDMA communication systems have been a much focused spot in modern secured communication systems technology [1-3]. CDMA is a multiplexing technique based on the spread spectrum principle [4, 5]. Its main advantage is that it allows the increase of the resistance to selective frequency fading, by spreading the power over a wide frequency band of a channel. In addition, CDMA gives the signal the ability to transmit noise of a form that makes it hardly detectable by unintended receivers [6]. Another advantage, due to the spread spectrum, is its resistance to jammers that may appear during transmission. The use of spreading sequences as codes distinguishes different users, giving the advantage of simultaneously exploiting the entire frequency band and time intervals [7], resulting in better utilization of the available resources. The conditions imposed by the orthogonality of code sequences reduce the interference between users [8]. Although CDMA has been successfully utilized in radio and optical communication systems, by either coherent or non-coherent approaches, many studies have focused on its application in other fields. In [9], an investigation was performed on induced random chip flipping used for multicarrier-based CDMA systems, securing multiple access. Researchers in [10] employed the Physical Layer Security (PLS) to protect confidential information in wireless CDMA, by proposing a binary signature design able to mitigate the reception of confidential messages by eavesdroppers. In [11], after evaluating the disguised jamming impacts on conventional CDMAS, a robust method was proposed using Advanced Encryption Standard (AES) to produce security-enhanced scrambling codes instead of conventional ones, to content disguised jamming using secure scrambling. Other researches contributed on CDMA applications in Collaborative Reader [12] and in Centimeter Accuracy Locating [13] in harmonic RFID systems. The detection of a received MC-CDMA signal and its time-delay estimation are very important in many applications, as multipath [14] and Multiple-Access Interference (MAI) can affect them. In this context, some approaches focused on mitigating the multipath and MAI impact, such as data-aided precoding [15], Reiterative Minimum Mean-Square Error (RM MSE) filters [16, 17] and Ultra-Wide Band (UWB) [18]. Atmospheric Optical (AO) communication emerged as a good candidate for broadband access environments, thanks to its ability to provide huge data rates. A suitable tool for the mathematical formulation of Bit-Error Rate (BER) on a secured relay-assisted AO/CDMA system in multiuser optical communications and especially over log-normal turbulence channel was deployed in [19]. Digital CDMA systems using chaotic signals for data spreading offer a high degree of security against unauthorized reception were reported in [20]. However, this technique adds an extra level of complexity regarding the synchronization between the transmitter and the receiver. The clock signal must be recovered, the two clocks must synchronize, and the receiver’s digital chaotic generator must be synchronized with the transmitter’s [21]. The design of a passband system requires a procedure for carrier recovery. A block diagram of a baseband system is shown in Figure 1.

This paper describes and validates experimentally an approach for synchronizing chaotic generators. Both transmitter and receiver chaotic generators use the same parameters, but they also need to start at the same time step in relation to the transmitted data, in order to recover them correctly. The chaotic generators were implemented with the Lorenz model [22].
Four methods have been proposed on the synchronization of analogue chaotic generators [23]. The first three methods, collectively known as continuous chaotic synchronization, are additive chaos masking, chaotic modulation and chaotic cryptosystem [24, 25]. The disadvantage of using these techniques is the requirement of large bandwidth due to continuous synchronization framework injection into the slave [26]. The fourth method, known as impulsive synchronization, overcame this issue consuming much less bandwidth [26]. Authors of [27] stated that impulsive and continuous synchronization are not reliable for chaotic communication systems, as both of them inject the drive signal into the slave system, which does not provide robust channel noise sensitivity. The method presented in [27] was based on an asynchronous serial communication protocol without the need of injecting signal into the dynamics of the slave system. Many researchers focused on the synchronization of chaotic generators. In [20], a novel detection method was proposed and was experimentally validated for weak sinusoidal signal that may exist in strong noise between two synchronization systems. In [28], a chaotic secure communication system was proposed, achieving significant improvement in the chaotic dynamic behavior by the application of phase modulation of the single loop feedback light.

In this paper, synchronization is based on sending a sync stream block used to trigger the receiver chaotic generator. When the sync stream is detected, the transmitter’s chaotic generator output signal is identical to the receiver’s, since both chaotic generators are also clock-synchronized. This synchronization method was implemented, tested and validated in hardware using two FPGA boards. The design procedure was based on different design tools: Matlab, Xilinx System Generator, ISE and the Hardware Description Language VHDL.

II. CLOCK RECOVERY

A. Clock Recovery Based on a Simulink Model

In serial digital data transmission, the local clock on the receiver must be adjusted in frequency and phase to the clock rate of the incoming signals [29]. The purpose of the recovered clock is to re-time the incoming serial data so that they can be received and processed synchronously, and the data transmitted can be properly retrieved [30]. The first step is to synchronize the clocks between the transmitter and receiver. Figure 2 shows that the clock is missing from the random data. The first step of the clock recovery process is to detect the rising and falling edges, and then recover the clock based on a band-pass filter. Figure 3 shows the SIMULINK model of the clock recovery. Figure 4 shows the detected rising and falling edges. The FFT function shows the clock signal and its harmonics that appear in the signal, as shown in Figure 5.
Figure 6 shows the simulated results of the clock data recovered from the random transmitted data.

B. Clock Recovery based on System Generator® model

Figure 7 shows the clock recovery model based on System Generator. The FIR filter is able to recover the desired frequency from the random data generator. Figure 8 shows that the clock is recovered from the random data generator.

III. SYNCHRONIZATION METHOD

At first, the receiver recovers the master clock of the digital system. The second process is to synchronize the transmitter’s and receiver’s chaotic generators. The principle scheme of the synchronization is presented in Figure 9.

When the sync stream block transmits the 32-bit stream, a known constant delay is added before the Lorenz generator starts generating the signal at the transmitter. The receiver’s synchronization unit uses the received signal $r(t)$ to generate the de-spreading signal, which is identical to the Lorenz chaotic generator at the transmitter $c(t)$, and they are both clocked at the same clock rate. For data recovery, the received signal is multiplied with the chaotic signal, which is synchronized with the received signal, and then it is accumulated. The cross-product and summation process consist of the product block and the accumulator, the accumulator, and the rational block which compares the threshold with the accumulated value.

IV. SYNCHRONIZATION OF CHAOTIC SIGNALS

A. Transmitter System

The transmitter system includes the Lorenz generator, a spreading block, and a block generating the sync stream which is shown in Figure 10. The Lorenz generators are represented by 32-bit fixed point with 20 fractional bits. The chaotic signals are then serialized by using a parallel to serial convertor. The sync stream generates 32-bit length known data. The bit sequence $[1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]$ was used for this purpose. The sync stream block contains a counter, a ROM, a comparison block, an accumulator block, a register, and a convert block. Figure 11 shows the simulated signals of the sync sequence block.
B. Receiver System

The receiver system includes a Lorenz generator identical to the transmitter’s, and a sync detector block. The sync detector was deployed by using a Finite State Machine based on VHDL. When the 32-bit sequence is received, the finite state machine is able to detect it and enable a signal for chaotic generators. Figure 12 shows the receiver’s sync detector and Figure 13 shows the simulated signal of the sync sequence block at the receiver system. It is clear that the desired sync signal is detected perfectly and stays at one constantly.

V. FPGA Implementation and Results

A. Design Process

The architecture of the studied system is summarized in the following stages:

- Simulink was used for designing, testing and analyzing all integrated subsystems: the Lorenz chaotic generator, its analogue-to-digital signal convertor, the scrambling scheme of Lorenz signals, the four-user digital communication system based on a Lorenz stream cipher, the user data spreading based on Lorenz system, the bipolar encoding, the de-spreading blocks using cross-correlation, cross product, dot product and summation, the data extraction process, the user data threshold tracking value, and the Lorenz stream cipher for four users with and without added noise. Repetitive tests and comparisons between transmitted and recovered data were conducted for all Simulink models, using the Error Rate Calculation to evaluate the system performance and obtain the desired results. It is important to note that many Simulink blocks, not available in the library, were designed in order to ensure compatibility with Xilinx blocks. Xilinx blocks, such as buffers, were designed in the Xilinx System Generator.

- Consequently, the chaotic signal synchronization, the Lorenz system, the data and clock recovery, and their VHDL code were generated using System Generator.

- Afterwards, the synthesis and the configuration of the target device were ensured via the Integrated System Environment (ISE), and a program file, dedicated to a specific device, was generated before being downloaded to the FPGA card. The results were viewed using an oscilloscope.

- The final step included the testing of the system and the comparison between experimental and simulation results.

Table I shows the digital communication system specifications, and Figure 14 summarizes the design procedure of the studied system implementation using an FPGA board (Spartan 6).

<table>
<thead>
<tr>
<th>TABLE I. PARAMETER SUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main clock frequency</td>
</tr>
<tr>
<td>Modulation</td>
</tr>
<tr>
<td>Spreading code</td>
</tr>
<tr>
<td>Spreading user data frequency</td>
</tr>
<tr>
<td>User data frequency</td>
</tr>
<tr>
<td>Data rate</td>
</tr>
<tr>
<td>FPGA development board</td>
</tr>
<tr>
<td>FPGA family</td>
</tr>
<tr>
<td>FPGA (IC) model number</td>
</tr>
<tr>
<td>Number of user data streams</td>
</tr>
<tr>
<td>generated and encrypted</td>
</tr>
<tr>
<td>Method of user data stream</td>
</tr>
</tbody>
</table>

B. Test and Results of the System

The Lorenz equations where \( x \), \( y \) and \( z \) are state variables, while \( A, B \) and \( C \) are parameters, are:

\[
\begin{align*}
    x &= A(y-x) \quad (1) \\
    y &= B.x - y - x.z \quad (2) \\
    z &= x. y - C.z \quad (3)
\end{align*}
\]

In hardware implementation, the chaotic signal was digitized to 32-bit with 25 fractional bits. Figure 15 shows the real time results, as visualized by the oscilloscope.
Alshammari: Synchronization of Two Chaotic Stream Ciphers in Secure CDMA Communication Systems

C. System Performance Analysis

The system bit rate achieved was 2Mbps. However, this data rate could increase by using one modulation scheme, such as BPSK (Binary Phase Shift Keying). The synchronization technique is robust against channel noise and it does not affect the efficiency of channel usage.

VI. CONCLUSION

This paper described a practical system for synchronizing two chaotic generators used in a digital CDMA. The method based on a sync stream transmitted to the receiver in order to trigger the Lorenz chaotic generator in the same time step. Thus, the chaotic synchronization was maintained and it was not affected by channel noise. Both the receiver and transmitter were implemented using two separate Spartan 6 FPGA boards. The proposed technique was validated experimentally, and the
obtained results demonstrate the robustness of the system. The advantage of a synchronization method is that there is no need to inject a signal into the dynamics of the slave system, which affects the channel efficiency. In addition, this method is not affected by high noise. However, the disadvantage of this method is that when the synchronization signal is affected, the synchronization between the two systems will be lost.

REFERENCES


