International Journal of Computing and Optimization
Vol. 4, 2017, no. 1, 1 - 9
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ijco.2017.61023

Designing a Complex Code Generator Using LFSR for Multichannel Encryption

Farah Maqsood

Department of Optometry and vision sciences, King Saud University
Riyadh 11495, Saudi Arabia

Abstract

Multiple key streams can be generated with the help of a single generator for the encryption of multiple messages in parallel. In this paper a novel Linear Feedback Shift Register (LFSR) based line encryption scheme is presented for the generation of parallel keys using Gold codes, which increases the complexity of the key with lesser number of hardware.

Keywords: Line Encryption, Secure Communication, Linear Feedback Shift Register, Multichannel

1. Introduction

Various techniques have been used for reliable and secure data transmission. Cryptography is a known practical method for protecting information transmitted through communication networks that use land lines, communication satellites and microwave facilities [1]. Cryptographic methods can be divided into block ciphers, stream ciphers and public key cryptosystems. Block ciphers require high capacity digital computers for their implementation at high data rate. With a properly designed pseudorandom number generator, stream cipher can be as secure as block cipher of comparable key length. Encryption is the process of disguising a message in such a way as to hide its substance [2]. In digital communication systems, encryption is a popular means for data security [3]. The security mechanisms used on the internet are made up of a number of technologies that encrypt/decrypt and authenticate. Encryption/decryption involves scrambling/unscrambling data that is transmitted to prevent understanding of intercepted data. Authentication involves verification of the identity of the participants in the communication [4]. Traditional stream ciphers do not include any randomness in generation of the ciphertext. They are based on the deterministic operations which expand a short secret seed into a long pseudorandom sequence [5]. Line encryption techniques are simple to implement in hardware and found useful for commercial applications. The primary advantage of a stream cipher is that stream ciphers are almost always faster and use far less code than do block ciphers [6]. In [4] there is also good evidence that stream cipher algorithms in general encrypt faster than block cipher algorithms and expanding the key size could reduce the encryption time on some algorithms that have fixed round number. Therefore, stream ciphers exhibit good properties including no error propagation, security levels properly selectable according to certain security criteria, and a higher processing ability than block ciphers, however, new high-speed communication systems are requiring faster data encryption [7].

For applications that require encryption/decryption of a stream of data, such as over a data communications channel or a browser/web link, a stream cipher might be the better alternative. For applications that deal with blocks of data such as file transfer, email and data base block ciphers may be more appropriate [8].

Among various possible stream ciphers, LFSR based structures have gained more popularity because of their simple structures and low hardware implementation costs. Hand-held communication devices pose a potential application, where hardware ciphers are highly needful.

The main disadvantage of LFSR based structure is its vulnerability to attack due to inherent linearity in the structure. LFSR based stream ciphers mainly employ two different methods to spoil this linearity. In the first method, nonlinearity is introduced by using a suitable cryptographic Boolean function and the in the second method, the LFSR is irregularly clocked to effect non-linearity. An LFSR keystream generator of length L produces maximal length sequence of periodicity $2^L - 1$ if the feedback polynomial is primitive [9].

Stream cipher

Stream ciphers convert plaintext to ciphertext 1 bit at a time. A key stream generator outputs a stream of bits: k_1, k_2 ...k_n. This keystream is XORed with a stream of plaintext bits, p_1, p_2 ...p_n to produce the stream of ciphertext bits.

$$c_i = p_i \oplus k_i \tag{1}$$

At the decryption end, the ciphertext bits are XORed with an identical key stream to recover the plaintext bits.

$$p_i = c_i \oplus k_i \tag{2}$$

The systems security depends entirely on the insides of the key stream generators. The best attack against a cipher should be key exhaustion (trying every possible key until eavesdropper (interceptor) finds one that works). If key exhaustion is the best attack, then key size determines the symmetric key algorithm's strength. To find an *n*-bit key, it is, on average necessary to try 2^n-1 keys, but if one makes n sufficiently large, it becomes wildly impractical [10, 11].

2. Line encryption circuits for multichannel

Various techniques have been reported for parallel and secure data transmission. For the encryption of multiple channels of communications in a single box one can use a different pseudo-random sequence generator for each stream as in [10-12], but use of a different pseudo-random sequence generator for each stream requires more hardware and all the different generators have to be synchronized. So it would be simpler to use a single generator for this purpose. Multiple streams can be generated with the help of a single generator by clocking it multiple times. Three independent streams can be generated by clocking the generator three times faster and sending one bit into each stream. This technique works, but for N independent streams it may have trouble in clocking the generator N times faster. Another technique is to use the same sequence for each channel with a variable time delay, but this method is insecure. A scheme patented by NSA [10], which seems to be promising is shown in Fig (1).

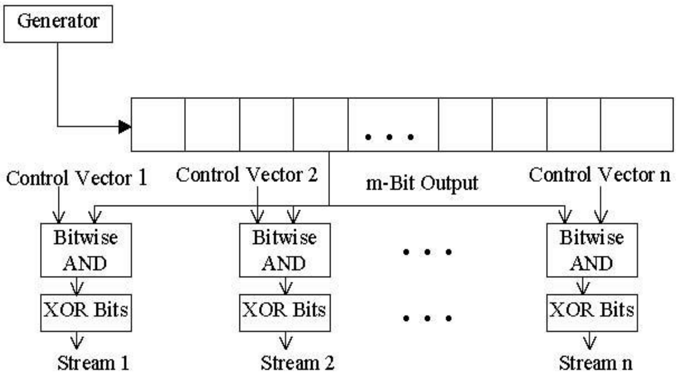

Fig. 1 Multiple bit generator

In this scheme output of a pseudo-random generator is passed through an m-bit Serial Input Parallel Out Shift Register. If there are n messages to be encrypted in parallel *n* different control vectors are used to generate *n* different streams (as shown in Fig. 1. to encrypt these messages separately. Here also the circuit is relatively complicated and a simpler scheme may be found preferable. Sometimes cryptographically secure pseudorandom numbers are not good enough. If an adversary gets a copy of that chosen generator and the master key, the adversary can create the same keys and break cryptosystem.

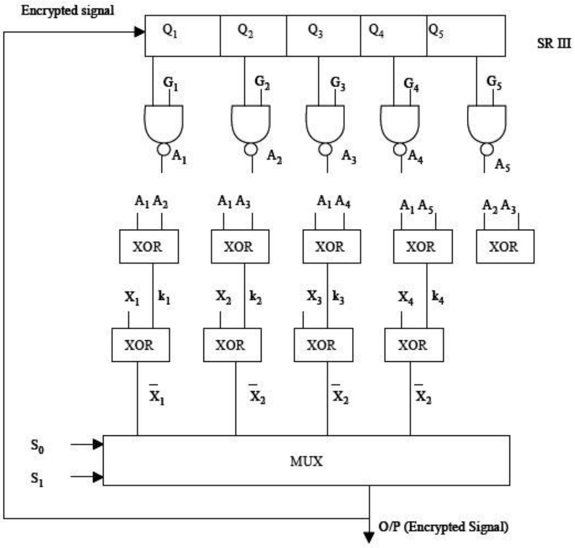

Fig. 2 Key Codes Generator and Multichannel Encrypter

In this paper a novel scheme is presented using Gold codes as shown in Fig. 2. In this scheme output of the multiplexer is given into a 5-bit Serial Input Parallel out Shift Register. It reduces hardware but increases the complexity of the key. As Gold codes work very well and give satisfactory results, so we have used Gold codes in our scheme for the generation of the keys. Gold codes are generated by modulo-2 addition of a pair of m sequences. These Gold codes can be generated in parallel using the circuit shown in Fig. 3. NAND gates outputs in pair are connected to XOR gate to generate keys k_1 to k_5 as shown in Fig. 2. With the connections shown, ten different keys may be generated which depend not only on Gold codes and initial states of SRIII but also on the data. The different data outputs X_1, X_2, X_3, X_4 with the keys k_1, k_2, k_3, k_4 are connected to the encrypter (XOR gates).

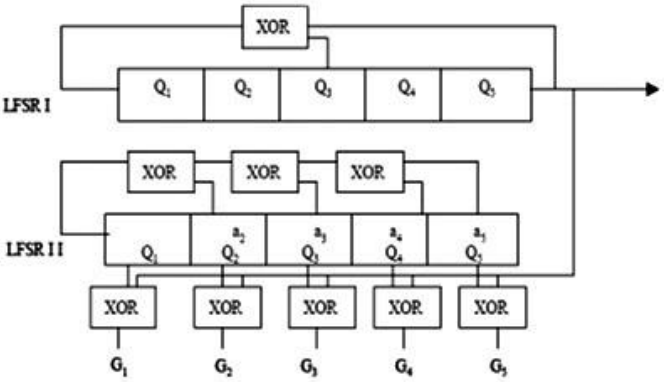

Fig. 3 Generator of Gold Codes (G1-G5)

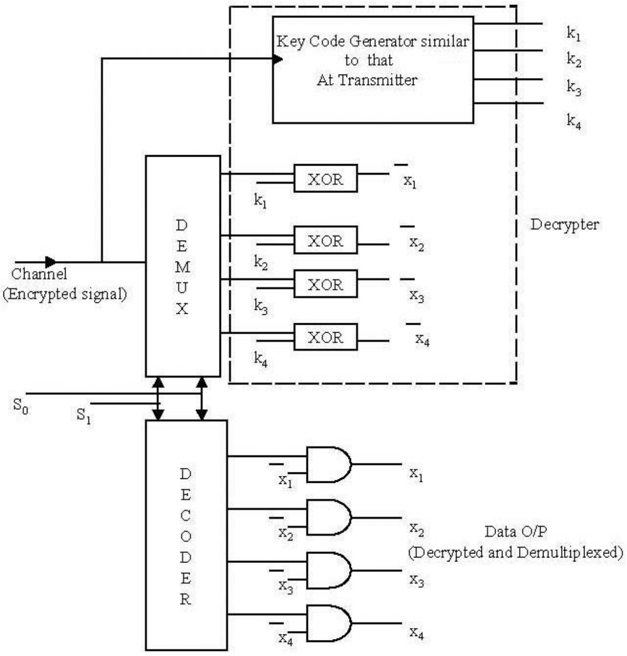

Fig. 4 Multichannel Decrypter circuit

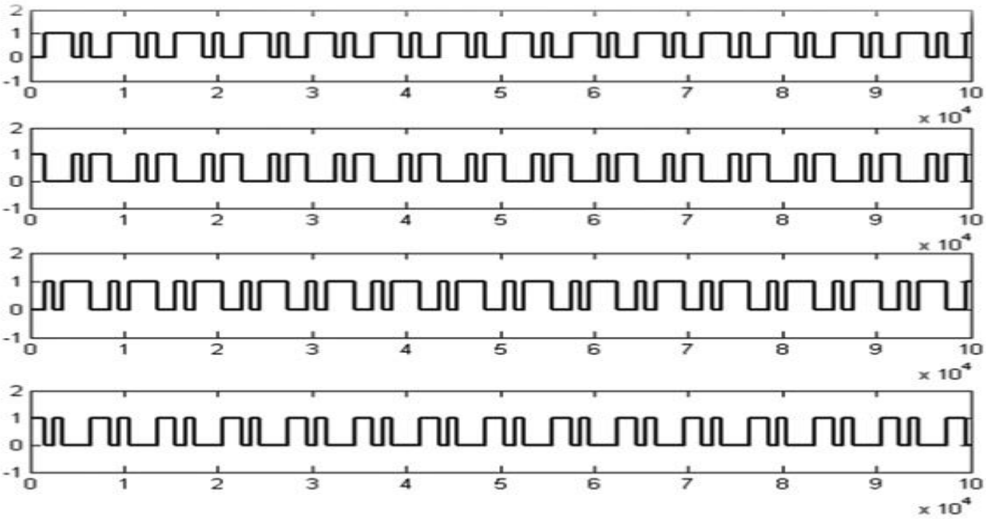

Fig. 5 Four Sequences 7 bit Messages

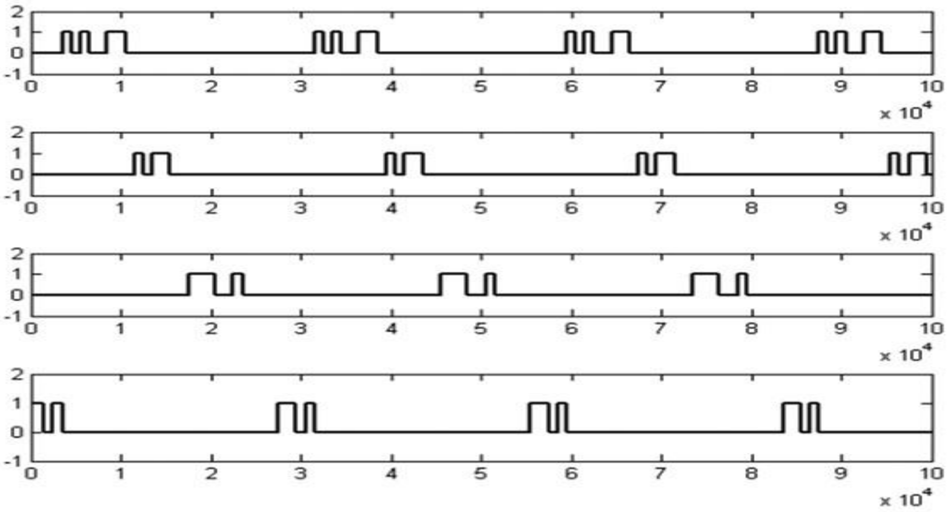

Fig. 6 Four outputs X_1, X_2, X_3 and X_4 after decryption and demultiplexing

The corresponding encrypted data are connected to different channels of multiplexers as shown in Fig. 2. A similar Gold/Key codes generators synchronized with the generators at transmitting end are used at the receiving end to obtain the requisite keys (k_1, k_2, k_3, k_4) as shown in Fig. 4. At this end encrypted signal is connected to the demultiplexer. Output lines of the demultiplexer are connected to different XOR gates. The other inputs of the XOR gates are keys k_1, k_2, k_3, k_4 corresponding to the key at transmitter and thus decrypted data are obtained at the different outputs of the XOR gates. The decrypted data will be available on any one of the channels selected with the address lines in synchronization with the MUX. The output of the other channel will be or depending on the output lines of DEMUX, equal to logic 0 or logic 1 respectively.

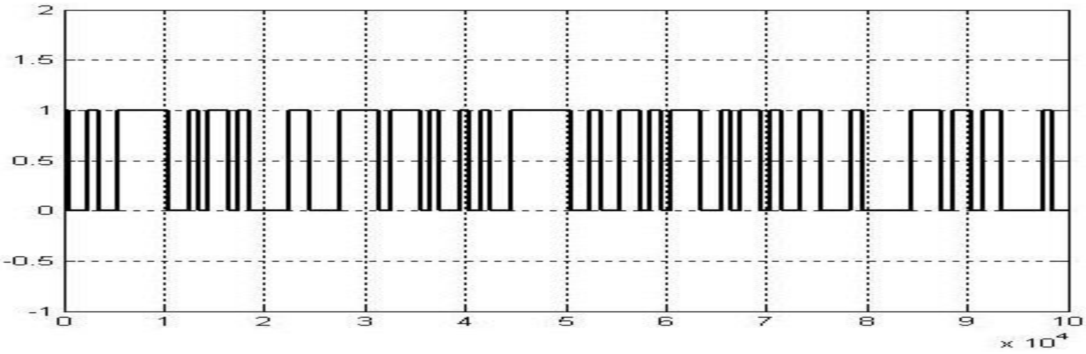

Fig. 7 Encrypted Data

Thus there is signal/data or (and hence noise) at the output of decrypter even if a line has not been selected. To remove this, these decrypted data are connected to AND gates and other inputs of AND gates are obtained from the output lines of the decoder using line control signal same as that of the DEMUX. Thus output of AND gate corresponding to selected line will be only the decrypted data at the corresponding time slot. In this way any sort of irrelevant data or noise is eliminated in the output. All the waveforms shown in Fig. 6 are the corresponding waveforms shown in Fig. 5 in the selected time slots.

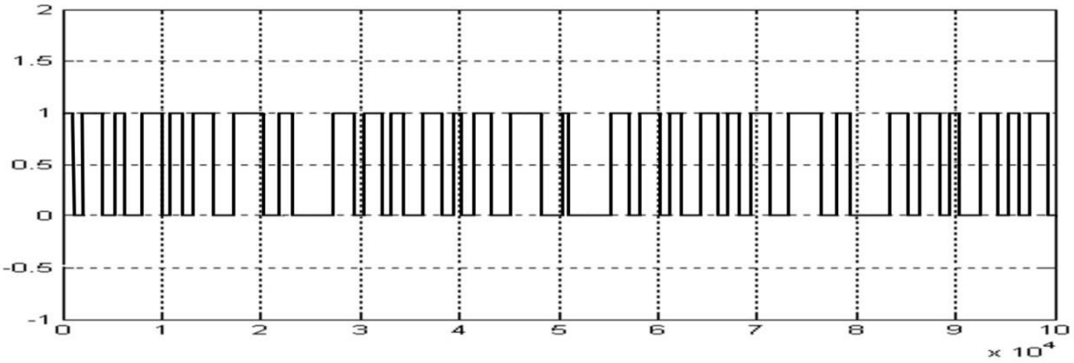

Fig. 8 Decrypted Data (Time Division Multiplexed)

3. Result and Discussion

The scheme presented in this paper is tested for the encryption and decryption of four 7-bit length pseudorandom message signals using Simulink in MATLAB. The encrypted data was observed at the output of MUX. In order to increase the secrecy of the message signals one can use optical fibers for the transmission of the encrypted data. At the receiving end the encrypted data was decrypted using similar circuit as shown in Fig. 4. The results obtained are presented in Fig. 5-Fig. 8, and it is clear that the scheme presented works satisfactorily. In this scheme Time Division Multiplexing (TDM) is done in a sample by sample manner. Depending on the frequency of the counter for the selection of the select lines, TDM can be done in bit-by-bit manner also.

4. Conclusion

In the present analysis, the proposed circuit performs multichannel data processing using stream cipher. Attention was concentrated on designing a simple circuit that can perform encryption of various message data simultaneously using time division multiplexing with high degree of secrecy.

The scheme presented in this paper for multichannel encryption increases the degree of secrecy of data with the minimum consumption of hardware resources. The scheme does not require a secure channel between each user and the center so it can be used for wireless communications.

References

[1] A. Vetro, H. Sun, P. DaGraca, T. Poon, Minimum drift architectures for three-layer scalable DTV decoding, *IEEE Trans. Consumer Electronics*, **44** (1998), 527-536. https://doi.org/10.1109/30.713160

[2] H. Lee, S. Moon, Parallel stream cipher for secure high-speed communications, *Signal Processing*, **82** (2002), no. 2, 259-265. https://doi.org/10.1016/s0165-1684(01)00180-3

[3] Muna Abdulla Al Shehhi, Joonsang Baek, Chan Yeob Yeun, The use of Boolean Functions in Stream Ciphers, *6th International Conference on Internet Technology and Secured Transactions*, (2011), 29-33.

[4] S. O. Sharif, S. P. Mansoor, Performance analysis of Stream and Block cipher algorithms, *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, (2010). https://doi.org/10.1109/icacte.2010.5578961

[5] M. Mihaljevic, H. Imai, A security evaluation of certain stream ciphers which involve randomness and coding, *2010 International Symposium on Information Theory & Its Applications,* Taichung, Taiwan, (2010), 789-794. https://doi.org/10.1109/isita.2010.5649616

[6] M. T. Sakalli, E. Bulus, F. Buyuksaracoglu, Cryptography Education for Students, *Information Technology Based Proceedings of the FIfth International Conference on Higher Education and Training*, Istanbul, (2004), 621-625. https://doi.org/10.1109/ithet.2004.1358246

[7] S. O. Sharif, S. P. Mansoor, Performance analysis of stream and block cipher algorithms, *3rd Int. Conf. Advanced Computer Theory and Engineering (ICACTE)*, (2010), 522-525. https://doi.org/10.1109/icacte.2010.5578961

[8] Iman Zarei Moghadam, Ali Shokouhi Rostami, Mohammed Rasoul Tanhatalab, Designing a Random Number Generator with Novel Parallel LFSR Substructure for Key Stream Ciphers, *2010 International Conference on Computer Design and Applications ICCDA*, (2010), 589-601. https://doi.org/10.1109/iccda.2010.5541188

[9] P. P. Depthi, P.S. Sathidevi, Hardware Stream Cipher Based on LFSR and Modular Division Circuit, World Academy of Science, Engineering and Technology 22, 2008.

[10] I. Kumar, *Cryptology*, Aegean Park Press, Laguna Hills CA 1997.

[11] B. Schneier, *Applied Cryptography Protocols, Algorithms and Source Code in C,* John Wiley & Sons Inc, New York, 1996.

[12] R.C. Dixon, *Spread Spectrum Systems with Commercial Applications*, John Wiley & Sons Inc, New York, 1994.

Received: November 9, 2016; Published: December 23, 2016