International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

A New Architecture of High Performance WG Stream Cipher

Grace Mary S.[1], Abhila R. Krishna[2]

[1]P G Scholar, VLSI and Embedded Systems, Department of ECE T K M Institute of Technology, Kollam, India

[2]Assistant Professor, Department of ECE, T K M Institute of Technology, Kollam, India

Abstract: ***Cipher is an algorithm for transforming the message. Stream ciphers are light weight symmetric key cryptosystems. These ciphers encrypt a plain-text or decrypt a cipher-text by adding the plain-text or cipher-text bit by bit with the generated keystream bits. Welch-Gong (WG) stream cipher is a cryptographically secure stream cipher. The keystream necessary for generating the WG stream cipher is generated using Welch-Gong keystream generator. The architecture is designed in such a way to reduce the computational complexity by reducing the number of multipliers. The WG design uses the properties of the trace function by using type II optimal normal basis (ONB) representation to optimize the hardware cost in the WG transformation. In-order to reduce the complexity, some necessary signals needed to generate the initial feedback signal is eliminated and further recovered through a new module called Serialized Key Initialization Module (SKIM). The area overhead can be reduced and the performance can be increased by replacing the finite field multiplier used in the WG transformation by high speed Vedic multiplier. The architecture is synthesized using Xilinx ISE Design Suite13.2 and simulated using Xilinx ISim.***

Keywords: Welch-Gong (WG), Optimal Normal Basis (ONB), Welch-Gong (WG) transformation, stream cipher, Serialized Key Initialization Module (SKIM).

1. Introduction

In the present world, the transmission of data between the sender and the receiver is an important factor. In order to establish a secure communication, data is transmitted as block ciphers or stream ciphers. Stream ciphers are more preferred for communication since they can be built using simple devices and also more immune to error propagation [3]. Synchronous stream ciphers consist of a keystream generator which produces a sequence of binary digits. The generated sequence is called as key-stream. The keystream is added to the plain-text digits to produce the cipher-text. A secret key K is used to initialize the key-stream generator and each secret key corresponds to a generator output sequence. The secret key is shared between the sender and the receiver to recover the original message.

Stream ciphers can be divided into bit-oriented stream ciphers and word-oriented stream ciphers [3]. The bit-oriented stream ciphers are designed using binary linear feedback shift registers (LFSRs) together with filter or combiner functions. They can be implemented in hardware very efficiently, but their software implementations are very slow [6]. Most of these stream ciphers are based on LFSRs but they operate on words instead of bits. This results in a very high throughput in software. While word oriented stream ciphers are fast in software and provide high security. Many bit oriented stream ciphers that exist so far are fast and can be implemented with a small amount of hardware but they all suffer from various crypt-analytic attacks. They do not satisfy properties such as exact period, linear complexity and good statistical properties, such as ideal two-level auto-correlation.

Many hardware oriented stream ciphers have been built using linear feedback shift registers (LFSRs) and filter/combiner Boolean functions. However, they are found to be vulnerable to algebraic attacks [7] [9]. So in the eSTREAM stream cipher project nonlinear feedback shift registers based stream cipher was introduced [8]. But the major drawback caused by them was their insecurity and difficulty in analyzing the design. As 4G technology began to emerge, a new class of stream cipher known as ZUC stream cipher was introduced but they also failed to satisfy some of the cryptographic properties [2].

So an alternative approach was to design a stream cipher in such a way that it is very easy to analyze. The WG family of stream ciphers have been designed using this approach [9]. To defeat the algebraic attacks on LFSR based stream ciphers, WG relies on non-linear Boolean functions with large number of inputs, high degree and complex ANF (Algebraic Normal Form). But the design of such a system was very complex. A Welch-Gong (WG) (29, 11) [29 corresponds to GF (2^{29}) and 11 is the length of the LFSR] stream cipher consists of a WG key-stream generator which produces a long pseudo-random key-stream [1]. The key-stream is XOR-ed with the plain-text to produce the cipher-text. The WG key-stream generators use Welch-Gong (WG) transformations as the filtering functions. The WG transformations have very large Algebraic Normal Forms (ANFs) and can be implemented in optimal normal basis form. WG stream cipher is a stream cipher designed on the basis of WG transformations to produce key-stream bits with good balance property, ideal tuple distribution, large linear complexity etc.

WG cipher is secure against algebraic attacks and satisfies the randomness properties such as decimation property, autocorrelation, balance property, 2 tuple distribution and linear span [10]. Hence it has a potential to be adopted in practical applications. The direct design using optimal

normal basis (ONB) reduced the number of multiplications and inversion over the Galois field (2^{29}). The inversion operation is replaced with a computation of the power 2^k-1 where k=10 [8].

In this paper, the properties of the trace function is adopted by using type II optimal normal basis representation in order to optimize the hardware cost by reducing the number of multipliers used in the WG transformation. This leads to the usage of a single finite field multiplier in the design which can be further replaced by high speed Vedic multiplier to increase the performance and reduce the area overhead. So this revised architecture can offer high clock speed, and better area and power consumption. Another important feature is that WG satisfies the randomness properties and can be adopted in practical applications when the application needs to satisfy complete security. The coding can be synthesized by the Xilinx ISE Design Suite 13.2, simulated using Xilinx ISim simulator.

2. Methodology

The WG stream cipher is generated using Welch-Gong keystream generator. The block diagram of Welch-Gong Stream cipher is shown in the Fig. 1.

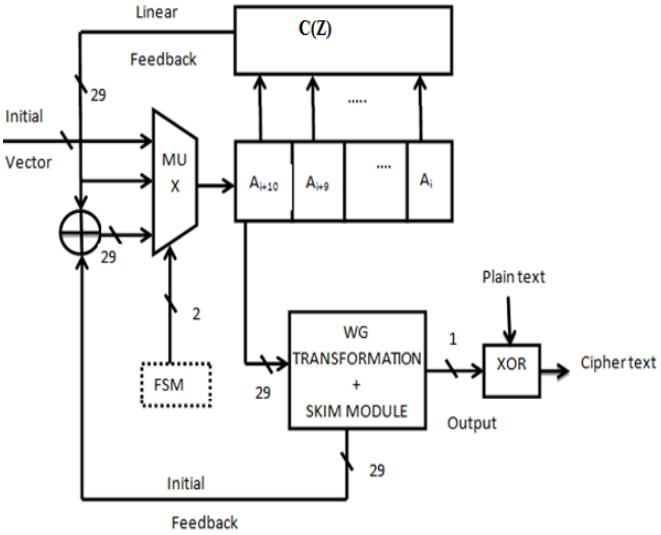

Figure 1: Block Diagram of Welch-Gong Cipher

The block diagram consists of linear feedback shift register (LFSR), WG Transformation block, Serialized key initialization (SKIM) block, and Finite State Machine (FSM) block. The initial vector, linear feedback signal, initial feedback signal are the inputs provided to the LFSR along with the key that is provide internally. Initial vector is a random block of bits which is used to introduce randomness to the output of the cipher. The WG (29, 11) [29 corresponds to Galois Field (2^{29}) and 11 is the length of the LFSR] have 3 phases of operation that is loading phase, key initialization phase and running phase [1]. During the loading phase the initial vector is loaded to the LFSR and loading phase requires l clock cycles. During the key initialization phase, linear feedback and initial feedback are fed as input to the LFSR and it runs for 2l clock cycles where key initialization takes place. Initial feedback signal is provided so as to update the LFSR during each clock cycle. Once the cipher has been initialized, the contents of the LFSR constitute the internal state of the cipher. During the run phase the LFSR is clocked once to generate the key-stream. When LFSR is clocked the contents of the last stage of LFSR is fed to WG transformation block which is integrated with SKIM module. The WG transformation block produces the key-stream bits, and SKIM module generates the initial feedback signal which is fed back to the LFSR. FSM controls all the 3 phases of operation. The generated key-stream bits are XOR-ed with the plain text to produce the WG stream cipher-text.

3. High Speed, Low Complexity WG Stream Cipher

The WG stream cipher is designed on the basis of WG transformation to produce keystream bits with guaranteed keystream properties such as balance property, long period, ideal tuple distribution, large linear complexity etc. The WG design utilizes the properties of the trace function by using type II optimal normal basis to minimize the hardware complexity of the transform. In order to reduce the complexity some necessary signals needed to generate the initial feedback signal is eliminated and then further recovered through Serialized Key Initialization Module [1]. Initial feedback signal is necessary to perform the key initialization phase. High speed Vedic multiplier is incorporated in the WG transformation circuitry to improve the performance of the whole circuitry by increasing the speed of operation.

3.1 Design of WG Hardware

The concept of optimal normal basis is introduced to reduce the hardware complexity of multiplying field elements. The multiplier uses an efficient linear transformation to covert the normal basis representation of elements to suitable polynomials. These polynomials are multiplied according to the implementation platform. The normal basis representation is suitable for hardware implementation and squaring can be done by simple cyclic shift which is free of hardware.

Proposition 1: In a type-II ONB, the trace of the field multiplication of any two GF (2^m) elements A = ($a_0, a_{1..}, a_{m-1}$) and B = (b_0 , b_1 ,….b_{m-1}) is computed as the inner product of A and B as shown in equation (1):

$$Tr\,(AB) = \sum_{i=0}^{m-1}(a_i b_i) \qquad (1)$$

The trace of the field multiplication of the two elements A and B can be computed using (1).However the result of multiplication will be lost. So the signals can be generated using the properties of trace function and thus WGTrans is computed as shown in (2):

$$WGTrans = Tr\,(1 \oplus X \oplus X^{r1}) + Tr\,(X^{2\wedge 2k}(X^{r1} \oplus X^{2k-1} \oplus X^{2\wedge 3k(2\wedge k-1)}) \qquad (2)$$

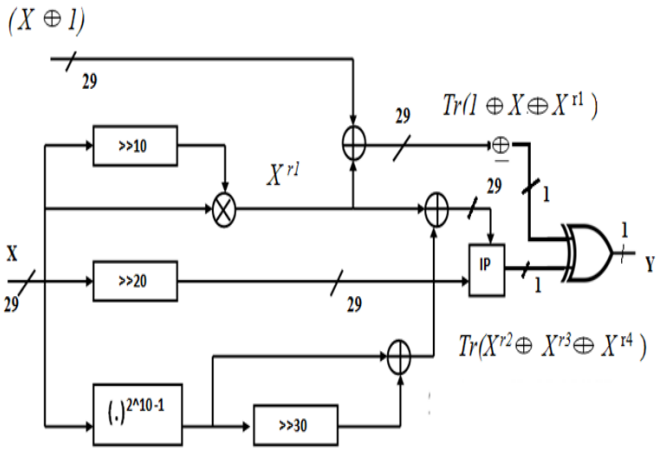

Figure 2: Design of WG transformation

The design of WG transformation is shown in Fig. 2. The architecture of the WG transformation is designed from (2). The keystream bits are obtained by XOR-ing the signals received from the inner product block and Galois field adder and the keystream bits are generated.

3.2 Architecture of FSM

The FSM controls the three phases of operation mainly the loading phase, key initialization phase and the run phase. The architecture of FSM is shown in fig. 3.

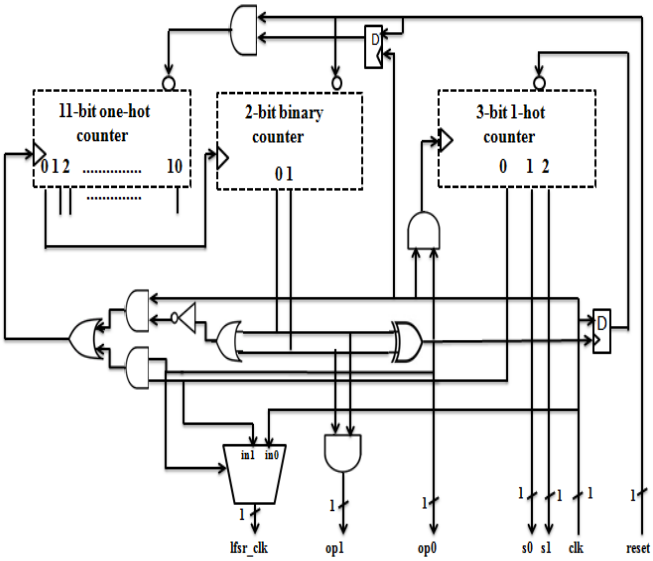

Figure 3: Architecture of FSM

FSM has two outputs namely output1 and op0, and two inputs namely clk and reset. It has 3 control signals namely lfsr-clk, s0 and s1. The lfsr-clk controls the clock input of the LFSR and triggers it to shift every three clock cycles. When the reset signal is low, the 11 bit one hot counter will be in its initial state (1,0,0,.....0) and 2 bit binary counter in (0,0)state. When the reset signal is pulled high the 11 bit one hot counter starts to increment one clock cycle later due to the 1 bit register connected to the input of the AND gate that drives its reset input. After 11 clock cycles, the 11 bit one hot counter will return to (1,0,0,.....0)state which will trigger the clock input of the 2 bit binary counter to(1,0) state thus indicating the start of key initialization phase. Then the clk signal starts triggering the 3 bit one hot counter to (1,0,0) state after consuming one clock cycle. During this phase, the first output bit of the 3-bit one-hot counter drives the clock input of the 11-bit one-hot counter. Therefore, it takes 33 clock cycles for the 11-bit one-hot counter to complete 11 counts and also takes 33 clock cycles for the 2-bit binary counter to increment. The 2-bit binary counter takes 66 clock cycles for to increment twice to start the running phase. When the running phase starts, the 2-bit binary counter will be in (1, 1)state, the 11-bit and the 3-bit one-hot counters stop counting, as their clock inputs become idle.

3.3 Design of SKIM

Serialized Key Initialization Module (SKIM) generates the initial feedback signal through serialized computation. The finite field multiplier used in the WG transformation block is used to achieve multiplication operation during the serialized computations. The function of the SKIM module is to compute a partial value of initial feedback signal and stores it in Register R2. The architecture of the SKIM module is shown in fig. 4.

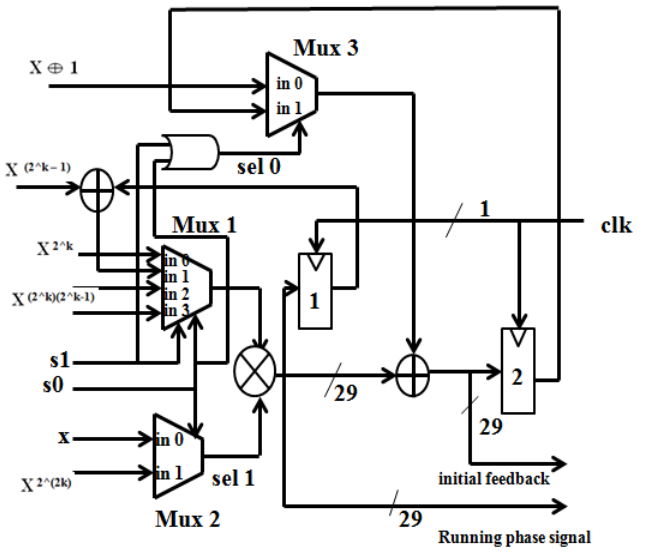

Figure 4: Architecture of SKIM

Initial feedback signal can be computed serially over 3 clock cycles, hence this complete round of serialized initial feedback computation is known as extended key initialization round. The control signals generated using FSM module is also used here as inputs. The operations in the SKIM module are realized in the table 1.

Table 1: Multiplexers outputs and next states of reg 1 and reg 2 as a function of s0 and s1

stages	s_0	s_1	Output			Next state	
			MUX_1	MUX_2	MUX_3	$Register_1$	$Register_2$
1	0	0	X^{2^k}	X	$X \oplus 1$	X^{r1}	$X^{r1} \oplus X \oplus 1$
2	0	1	$X^{r1} \oplus X^{2^k-1}$	$X^{2^{2k}}$	$X^{r1} \oplus X \oplus 1$	$X^{r4} \oplus X^{r2}$	$X^{r4} \oplus X^{r3} \oplus X^{r2} \oplus X^{r1} \oplus X \oplus 1$
3	1	0	$X^{2k(2^k-1)}$	X	$X^{r4} \oplus X^{r2} \oplus X^{r1} \oplus X \oplus 1$	X^{r3}	$X^{r4} \oplus X^{r3} \oplus X^{r2} \oplus X^{r1} \oplus X \oplus 1$

3.5 Linear Feedback Shift Register

The 11 stage LFSR consists of 11 D flip-flops, where each D FF has two additional inputs other than clock and reset. The additional inputs are (i) concatenated form of key and Initial Vector (ii) output generated from the previous D FF. When reset is pulled down the feedback function is calculated according to the (3):

$$C(z) = z^{11} + z^{10} + z^9 + z^6 + z^3 + z + \Upsilon \quad (3)$$

During the key initialization phase, the allocation of the key as well as initial vector takes place. The key is divided as blocks of 16 bits each and initial vector is divided as blocks of 8 bits which makes a total of 24 bits and the remaining bits are assigned as zeros thus forming 29 bits.

3.6 Vedic Multiplier

Multiplication is an important operation that is critically used in the integrated block of WG transformation and SKIM module to generate keystream bits. Presently, finite field multiplier i.e. Galois field multiplier is used which consumes more area and power. The multiplier is been replaced by Vedic multiplier which can be used to increase the speed and performance which also results in the parallel generation of partial products and faster carry generation.

4. Results and Discussion

The WG keystream generator consists of 11 stage LFSR, FSM, WG transformation block and SKIM module. The modules are modeled using Verilog in Xilinx ISE Design Suite 13.2 and the simulation of the design is performed using Xilinx ISim to verify the functionality of the design. The key-stream is generated using WG key-stream generator and the various modules such as FSM, WG transformation block, LFSR and SKIM module is simulated using Xilinx ISim. The RTL schematic view of the WG keystream generator is shown in Fig. 4.

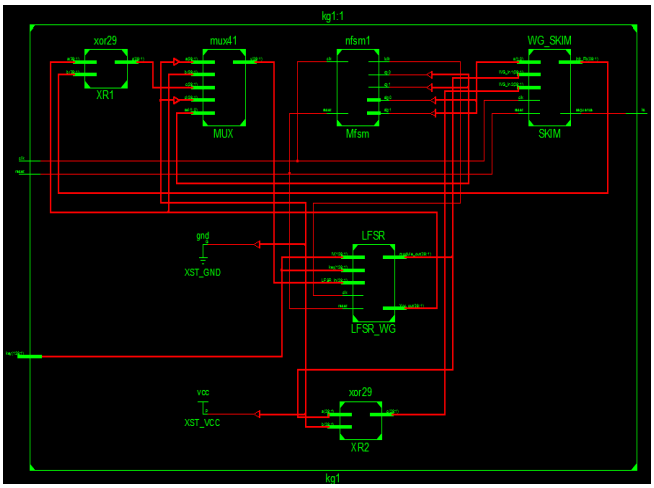

Figure 4: RTL-Schematic of WG Keystream Generator

The simulation result of the WG keystream generator is shown in Fig. 5. The inputs to the keystream generator is initial vector, clk and reset signal. When reset is high keystream is generated.

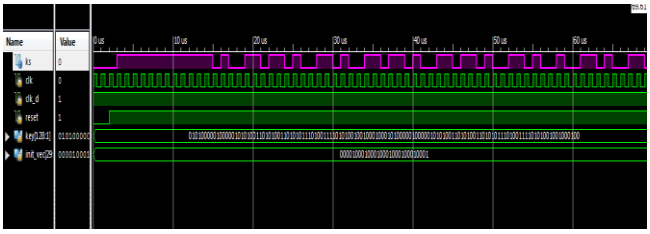

Figure 5: Simulation Result of WG keystream generator

5. Conclusion

Many hardware oriented stream ciphers that are built using linear feedback shift registers are very hard to analyze. The arrival of 4G mobile technology led to the design of WG stream cipher. WG stream cipher is very secure as it satisfies all the cryptographic properties such as long period, large linear span, cross correlation, balance property etc. WG stream cipher is designed on the basis of WG transformations to produce key-stream bits which are found to be more secure for communication purposes. The design of WG stream cipher is mainly concentrated in reducing the number of multipliers used. The design of WG stream cipher uses Optimal Normal Basis (ONB) to reduce the computational complexity. It is obtained through using the properties of ONB accompanied by the serialized computation of the

initial feedback signal during the key initialization phase. The work also focuses in increasing the performance by incorporating the Vedic Multiplier in the integrated blocks of WG transformation block and SKIM module.

References

[1] Hayssam El-Razouk and Arash Reyhani Masoleh, Guang Gong "New Implementation of the WG stream cipher "IEEE Transactions on VLSI Systems,,Vol 22 No.9, Sep.2014.
[2] H. Wu, T. Huang, P. Nguyen, H. Wang, and S. Ling, "Differential attacks against stream cipher ZUC" ,in Advances in Cryptology—ASIACRYPT (Lecture Notes in Computer Science), vol. 7658, X. Wang and K. Sako, Eds. Berlin Heidelberg, Germany: Springer-Verlag, 2012.
[3] Y. Luo, Q. Chai, G. Gong, and X. Lai"A lightweight stream cipher WG-7 for RFID encryption and authentication," in Proc. IEEE Global Telecommun. Conf., pp. 1–6,Dec. 2010.
[4] C. Lam, M. Aagaard, and G. Gong, "Hardware implementations of multi-output Welch-Gong ciphers," Dept. Department of Electr. Comput. Eng., Univ. Waterloo, Waterloo, ON, Canada, Tech. Rep. CACR 2011-01, 2009.
[5] E. Krengel, "Fast WG Stream Cipher," in Proc. IEEE Region 8 Int. Conf. Comput. Technol. Electr. Electron. Eng., pp. 31–35,Jul. 2008.
[6] Y. Nawaz and G. Gong, "WG: A family of stream ciphers with designed randomness properties," Inf. Sci., vol. 178, no. 7, pp. 1903–1916, 2008.
[7] N. Courtois, "Algebraic attacks on combiners with memory and several outputs," in Information Security and Cryptology—ICISC (Lecture Notes in Computer Science), vol. 3506, C.-S. Park and S. Chee, Eds. NewYork, NY, USA: Springer-Verlag, 2005, pp. 3–20.
[8] G. Gong and Y. Nawaz. (2005). The WG Stream Cipher[Online]
[9] N. Courtois, "Fast algebraic attacks on stream ciphers with linear feedback," in Proc. Advances in Cryptology—CRYPTO (Lecture Notes in Computer Science), vol. 2729. New York, NY, USA: Springer-Verlag, 2003, pp. 176–194.
[10] G. Gong and A. Youssef, "Cryptographic properties of the Welch-Gong transformation sequence generators," IEEE Trans. Inf. Theory, vol. 48, no. 11, pp. 2837–2846, Nov. 2002.
[11] R. C. Mullin, I. M. Onyszchuk, S. A. Vanstone, and R. M. Wilson, "Optimal normal bases in GF(p^n)," Discrete Appl. Math., vol. 22, no. 2, pp. 149–161, Feb. 1989.