FPGA Implementation of WG Stream Cipher

Anna Johnson

Assistant Professor,ECE Department, Jyothi Engineering College,Thrissur

***Abstract*—** Cryptography is the technique of providing security to a network. The term Cryptography is derived from two Greek words crypto means hiding and graphy means way of writing. The hiding of information is done through encryption algorithms. Cryptography is the fundamental of authentication process, it has two protocols namely, encryption and authentication protocols.In the encryption we have two types of ciphers, stream ciphers and block ciphers. In stream ciphers we send data in the form of bits or bytes. The example of stream ciphers are RC4 and WG stream ciphers respectively. These two are key generation algorithm.This paper deals with WG stream cipher.This is done using verilog programming and implemented in FPGA using Spartan 3 kit.

Keywords—WG,LFSR,IV,GF

1. Introduction

In this section we discuss the importance of securing valuable information in our day to day life. We introduce cryptography and terms that should be familiar. Types of ciphers used to encrypt our data. And also a brief description about basic types of attacks that are possible on our ciphers. The requirements of information security within an organization have undergone two major changes in the last several decades. Before the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means. An example of the former is the use of rugged filing cabinets with a combination lock for storing sensitive documents. An example of the latter is personnel screening procedures used during the hiring process. Network security measures are needed to protect data during their transmission. In fact, the term network security is somewhat misleading, because virtually all business, government, and academic organizations interconnect their data processing equipment with a collection of interconnected networks. Such a collection is often referred to as an internet and the term **internet security** is used. If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.

A. Stream ciphers:

In stream cipher the plaintext is converted to cipher text by **one bit at a time.** It generates arbitrarily long stream of key material (bits) known as keystream. The generation of keystream output is based up on the internal state which is usually hidden inside the cipher and changes frequently as cipher operates. During encryption the keystream is XOR'ed (exclusive-or operation) with each plaintext one bit at a time. Some of the examples of stream ciphers are Welch-Gong (WG) cipher, RC4, grain, trivium, A5/1 and so on.

B. Block ciphers:

Block cipher operates on the **fixed length blocks** (i.e. group of bits) of plaintext or ciphertext. The encryption operation is an unvarying transformation, which is controlled by using the secret key. For example, a block cipher might take 128-bit block of plaintext as an input and generate 128-bit block of ciphertext. Examples of block ciphers are Data Encryption Standard (DES), Advanced Encryption Standard (AES) and so on.

2. LITERATURE SURVEY

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.Stream ciphers are more preferred for communication since they can be built using simple devices and also more immune to error propagation. Synchronous stream ciphers consist of a keystream generator which produces a sequence of binary digits. The generated sequence is called as keystream. 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. A Welch-Gong (WG) (29, 11)^3 stream cipher consists of a WG key-stream generator which produces a long pseudo-random key-stream. The keystream is XOR'ed with the plain-text to produce the cipher text. The WG keystream 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 keystream bits with good balance property, ideal tuple distribution, large linear complexity etc. Hence it has a potential to be adopted in practical application. The direct design using optimal normal basis (ONB) [7]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 .

3. Mathematical background

Finite Fields [6] play a major role in some of the most interesting applications of modern algebra to the real world. In particular, the applications related to the data communication is a vital concern in our information friendly society. In today's technological advancements in the areas of space and satellite communications, protecting the privacy of information involve the use of finite fields in one way or the other. This chapter begins with a brief overview of concepts of field and group. Next, we will explain the concepts and types of finite fields of the form GF(p), where p is a prime number. Before going to the details of finite field extension of the form GF (p^n), where n is a positive integer, it needs to be discussed some of the elementary background in polynomial arithmetic operations. Finally, we briefly discuss about the normal basis and types of optimal normal basis.

Modular Arithmetic

Modular arithmetic has gained importance in the area of cryptography. In Public Key Cryptosystem algorithms such as RSA and Diffie-Hellman algorithm uses the theory of modular arithmetic including, symmetric key algorithms such as AES, IDEA and RC4. The major advantage of using modular arithmetic is that it allows us to do faster multiplication operations. For example, in any complex operations such as polynomial greatest common divisor calculation where we come across large number of integers to perform number of multiplication operations. The use of modular arithmetic reduces the computing times of these large operations. In one of the applications like error correcting codes each digit of the code is related to the elements of the finite field by using the modular arithmetic theory. The modular operator (mod n) maps all the integers within the confined set of integers {0,1,2,....(n-1)} and all the arithmetic operations are performed within this set. This techniques is called as modular arithmetic. The set of integers and nonzero integers of mod n are denoted by Zn and Zn^* respectively.

Groups and fields

Fields and groups are the well known algebraic structures of the abstract or modern algebra. In abstract algebra, we work with sets on which elements can be operated algebraically. For instance, we can say that by combining two elements of a set in several different ways the third element of the set can be obtained. All these operations will follow certain specific rules [6] which will define the nature of the set. The notation followed for operations on set of elements is usually same as the notation for ordinary addition and multiplication.

Finite Fields

Most of the cryptographic algorithms such as the Digital Signature Standard (DSS), the El Gamal public key encryption, elliptic curve public key cryptography are heavily depend on the properties of finite fields and it is also used in Advanced Encryption Standard (AES) cryptography. The order of the finite field must be a power of prime p^n, where n is a positive integer. Here two cases exits: for n = 1, the finite field is of the form[6] GF(p) where GF stands for Galois Field and for n > 1, the finite field is of the form GF(p^n). The finite field GF (p) has different structure compared to the finite field GF (p^n).

4. KEY INITIALIZATION

Linear feedback shift register (LFSR)[2] have been widely used in keystream generators in stream ciphers, random number generators in most of the cryptographic algorithms. The n binary storage units[6] are called as the stages of the shift register and their contents are in the form of n bits in length, which is called as the internal state of the shift register.

(4.3)

Let (a_0, a_1, a_2..... an-1) ϵ GF(2^n) be the initial state of the LFSR and f(x_0,x_1, x_2.... x_{n-1}) be the feedback function or feedback polynomial, as shown in figure 4.1. If the feedback function is a linear function then it can be expressed as

f(x0, x1, x2.... x n-1) = c0x0 + c1x1 + c2x2 + ...+ cn-1x n-1;
ci ϵ GF(2)

After each consecutive clock pluses the LFSR will generate a output binary sequence b of the form b = a0, a1....

The output sequence of LFSR satisfies the following recursive relation

a k+n =Σci ak+i; k = 0,1.......

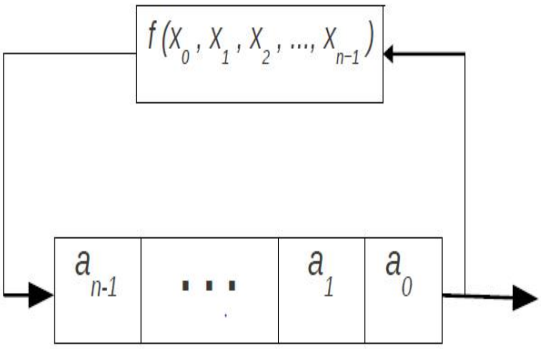


Fig.1.Block diagram of LFSR[1]

Initial Vector(IV)

The ideal IV is a random number that is made known to the destination computer to facilitate decryption of the data when it is received. The IV[1] can be agreed on in advance, transmitted independently or included as part of the session setup prior to exchange of the message data. The length of the IV (the number of bits or bytes it contains) depends on the method of encryption. The IV length is usually comparable to the length of the encryption key or block of the cipher in use. The use of an IV[7] prevents repetition in data encryption, making it more difficult for a hacker using a dictionary attack to find patterns and break a cipher. For example, a sequence might appear twice or more within the body of a message. If there are repeated sequences in encrypted data, an attacker could assume that the corresponding sequences in the message were also identical.

Key Initialization Of WG Cipher

Consider the stages of LFSR namely S(1),S(2),S(3),….S(11) ϵ $F_2^{29.}$Each stage S(i) ϵ F_2^{29},is represented as $S_{1,….,29}$(i) where 1≤i≤11.Similarly we represent the key bits as $K_{1,….,j}$,1≤j≤128 and IV bits as $IV_{1,….,m,}$1≤m ≤128[1].

– 80 bits key and IV are loaded as

S 1,....,16(1) = *k* 1,....,16	*S* 17,..,24(1) = *IV* 1,..,8	*S* 1,..,8(2) = *k* 17...,24
S 9,....,24(2) = *IV* 9,..,24	*S* 1,..,16(3) = *k* 25,..,40	*S* 17,...,24(3) = *IV* 25,..,32
S 1,....,8(4) = *k* 41,....,48	*S* 9,..,24(4) = *IV* 33,..,48	*S* 1,..,16(5) = *k* 49,..,64
S 17,..,24(5) = *IV* 49,..,56	*S* 1,..,8(6) = *k* 65,..,72	*S* 9,..,24(6) = *IV* 57,..,72
S 1,....,8(7) = *k* 73,....,80	*S* 17,..,24(7) = *IV* 73,..,80	

The remaining bits in the LFSR are all set to zero. Once the key and the IV have been loaded, the keystream generator is run for 22 clock cycles.. During this phase the 29 bit vector,given by

keyinitvec =~ (q1 ^ (q2 ^ (q3 ^ q4)))
in Figure 1, is added to the feedback of the LFSR which is then used to update the LFSR. The key initialization process is shown in Figure 4.2. Once the key hasbeen initialized the LFSR is clocked once and the 1 bit output of the WG transformation gives the first bit of the running keystream. (4.4)

Since the linear complexity of the keystream is slightly more than 245 the maximum length of the keystream allowed to be generated with a single key and IV is 245. After this the cipher must be reinitialized with a new IV or a new key or both.

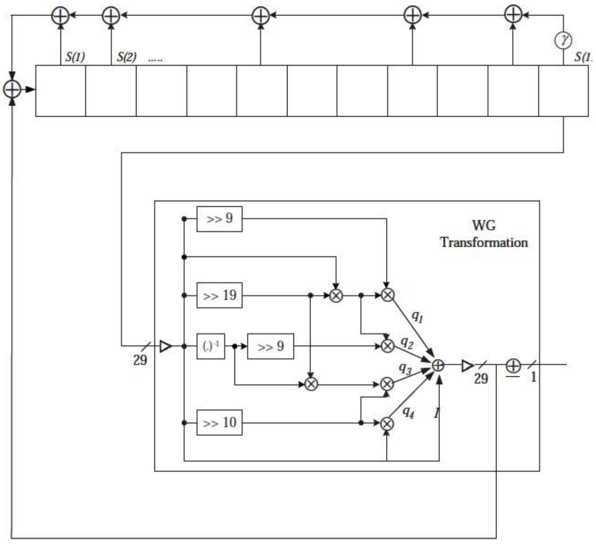


Fig 2.Key initialization of WG Cipher[1]

5. WG TRANSFORMATION

The cipher is based on WG (Welch-Gong) transformations which have proven cryptographic properties. The cipher has been designed to produce a keystream which has all the cryptographic properties of WG transformation sequences, is resistant to Time/Memory/Data tradeoff attacks, algebraic attacks and correlation attacks, and can also be implemented in hardware efficiently.

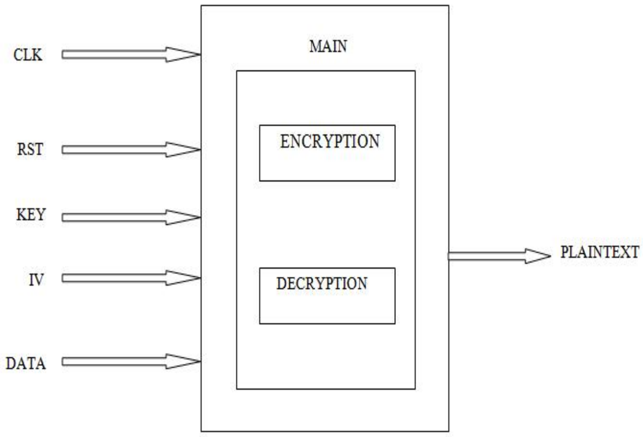


Fig 3. I/O Diagram of cryptography

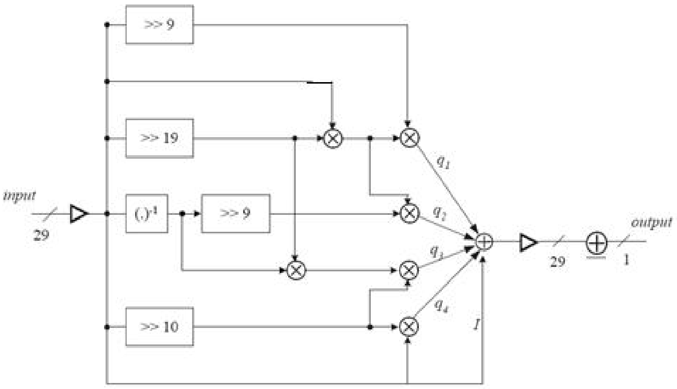


Fig 4 Block Diagram of Implementation of WG transformation:$F_2^{29} \longrightarrow F_2$ [1]

The WG transformation form $F_2^{29} \rightarrow F_2$ can be regarded as a boolean function in 29 variables. The exact boolean representation depends on the basis used for computation in F_2^{29} [1]. It has degree 11 and its nonlinearity is 2^{28}-2^{14}=268419072. The 29 bit input to the WG transformation function is regarded as an element of F_2^{29} represented in normal basis. From the figure the output of the WG transformation can be written as

$$output = \underline{\bigoplus}(\triangleright(q_1 \oplus (q_2 \oplus (q_3 \oplus (q_4 \oplus I)))))$$

where

$$q_1 = (I \gg 9) \otimes ((I \gg 19) \otimes I)$$

$$q_2 = (I^{-1} \gg 9) \otimes ((I \gg 19) \otimes I)$$

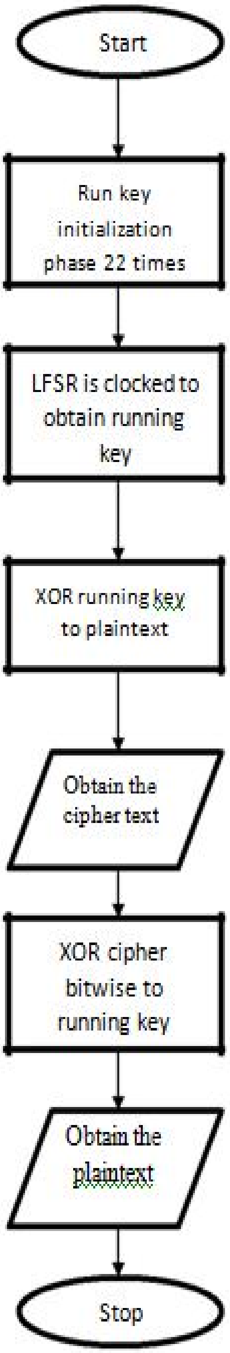
$$q_3 = (I^{-1} \otimes (I \gg 19)) \otimes (I \gg 10)$$

$$q_4 = (I \gg 10) \otimes I$$

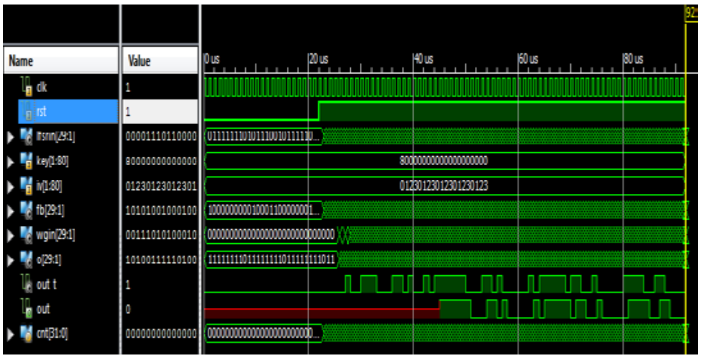
$$I = \triangleright(input).$$

The notation x ⊗ y means normal basis multiplication of x and y in $F_{2^{29}}$ defined by g(x). Similarly $(x)_{-1}$ means the normal basis inversion of x in $F_{2^{29}}$ defined by g(x). x ⊕ y represents the bitwise addition (XOR) of words x and y, and x À c represents the cyclic shift of x, c stages to the right where c is a positive integer. The symbol B(x) means all the 29 bits of x are complemented and L(x) means the addition of the 29 bits of x over F_2 (XOR) i.e., for x = (x_0, .., x_{28}), L(x) = $\mathrm{P}_{28i=0}$ x_i mod 2.

6. FLOWCHART

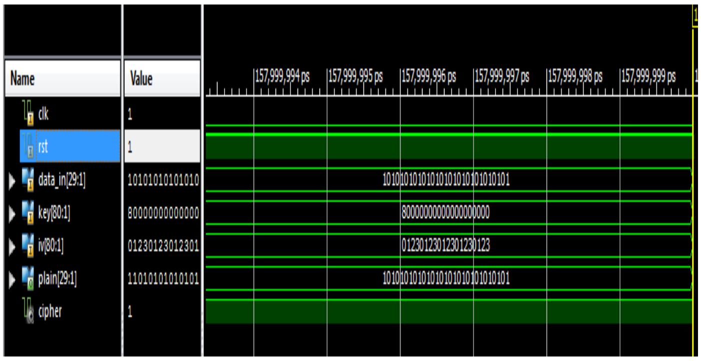

Start
Run key initialization phase 22 times
LFSR is clocked to obtain running key
XOR running key to plaintext
Obtain the cipher text
XOR cipher bitwise to running key
Obtain the plaintext
Stop

7. SIMULATION RESULT



Name	Value
clk	1
rst	1
ltsnn[29:1]	00001110110000
key[1:80]	80000000000000
iv[1:80]	01230123012301
fb[29:1]	10101001000100
wgin[29:1]	00111010100010
o[29:1]	10100111110100
out t	1
out	0
cnt[31:0]	00000000000000

Fig 5.Simulation result of WG keystream generator



Name	Value
clk	1
rst	1
data_in[29:1]	10101010101010
key[80:1]	80000000000000
iv[80:1]	01230123012301
plain[29:1]	11010101010101
cipher	1

Fig 6.Simulation result of cryptography

8. SYNTHESIS REPORT OF WG KEYSTREAM GENERATOR

- FPGA REPORT

NUMBER OF SLICES	POWER(mw)	DELAY(ns)
741	20.44	28.163

- ASIC REPORT

AREA	POWER(nw)	DELAY(ps)
17741	1036323.693	4

9. CONCLUSION

Thus proposed a new stream cipher, WG cipher, suitable for hardware implementations. The architecture is designed in such a way to reduce the computational complexity by reducing the number of multipliers.The cipher generates a keystream with guaranteed randomness properties and offers high level of security. Hence it has a potential to be adopted in practical application. The cipher can be implemented with relatively small amount of hardware. We believe that exhaustive key search is the most efficient way to recover the secret key or internal state of the cipher. We claim that we have not inserted any hidden weakness in the design of the WG cipher.

10. REFERENCES

[1]The WG Stream Cipher,Yassir Nawaz and Guang Gong,Department of Electrical and Computer Engineering,University of Waterloo,Waterloo, ON, N2L 3G1, CANADA,2004

[2]D. Watanabe, S. Furuya, H. Yoshida, and B. Preneel, A New Keystream Generator MUGI, Fast Software Encryption 2002, LNCS 2365, pp. 179-194, Springer-Verlag, 2002.

[3]M. Briceno, I. Goldberg, and D. Wagner, A Pedagogical Implementation of A5/1,http://www.scard.org, May 1999.

[4]G. Gong, and A. Youssef, Cryptographic Properties of the Welch-Gong Transformation Sequence Generators, *IEEE Transactions on Information Theory*, vol. 48, No. 11, pp. 2837-2846,Nov. 2002.

[5] A. Biryukov, and A. Shamir, Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers, Asiacrypt *2000*, LNCS 1976, pp. 113, Springer-Verlag, 2000.

[6] L. Massey, and J. Omura, Computational Method and Apparatus for Finite Field Arithmetic,*US Patent No. 4,587,627*, 1986.

[7] B. Sunar, and C. Koc, An Efficient Optimal Normal Basis Type II Multiplier, *IEEE Transactions on Computers*, vol. 50, No. 1, pp. 83-88, Jan. 2001.

[8] N. Courtois, Fast Algebraic Attacks on Stream Ciphers with Linear Feedback, *Advances in Cryptology-CRYPTO 2003*, LNCS 2729, pp. 176-194, Springer-Verlag, 2003.