J. of Ultra Comp. and I.T. Vol.1(1), 93-98 (2010). ISSN 2229-3531 (Print)
www.compitjournal.org 2455-9997 (Online)

Implementation of digital logic in generation of symmetric key for cryptography

POONAM RATHEE[1], DAVINDER RATHEE[2] and KEHAR SINGH[3]

[1]Computer Engg. Dept., SIMT, [2]Rewari, Vaish College Of Engineering, Rohtak
[3]Computer Engg. Dept , SIMT, Rewari (INDIA)

Email id: jerry_2c@yahoomail.com,

(Acceptance Date 25th May, 2010)

Abstract

Cryptography is the art of achieving security by encoding messages to keep the information non-readable. Cryptography actually is a mathematical scrambling and unscrambling of data to achieve, Confidentiality (Readable only to sender and receiver); Integrity (not modified by any one to integrity of message); Non repudiation (sender can't refuse the claim of not sending); and Entity authentication. While sending or receiving the data, special keys are used to encrypt/decrypt data to verify the original contents. This key makes the process of cryptography secure. There are two approaches for the generation of encryption/decryption keys *i.e* software and hardware. The purpose of this paper is to generate encryption key using hardware technique. To design electronics circuit basic concept is derived from encoding and decoding of data vis four different channels at different times. Selections of channels will be guided by timer and decoder circuits. one form to another from at transmitter end receiver end respectively. The purpose of this paper is to generate the key code using electronic circuits to make the data secure.

Introduction

Techniques associated with generation of the key by coding data at transmitter end and the same code is used in the receive side to despread the received signal so that the original data sequence may be recovered. The first technique associated with generation of the key is Spread-spectrum in the form of P-N sequence[2] due to its ability to reject interference. In this the data sequence occupies a bandwidth in excess of the minimum bandwidth necessary to send it. It is accomplished before transmission through the use of a code that is independent of the data sequence. The same code is used in the receiver to despread the received signal so that the original data sequence may be recovered.

In addition to key generation technique by Xoring of P-N sequence there are some another alternative logic can also be used to generate key effectively. The alternative logics may be

Xnoring of data with random sequences[4]
2'c complement[6]
Code converters (Binary to Gray)[5]

Before the discussion about key generation, it is important to understand the concepts and terminology used in cryptography techniques. There are two basic types of cryptography:

(1) Symmetric key cryptography.[2,5]
(2) Asymmetric key cryptography or Public key[3].

Some Terminology used for Encryption Process[1]

- M denotes a set called the message space. M consists of strings of symbols from an alphabet. An element of M is called a plaintext message or simply a plaintext. For example, M may consist of binary strings, English text, computer code, etc.
- C denotes a set called the ciphertext space. C consists of value (V) voltage from the definition of input binary data string, which may differ from the alphabet of definition for M. An element of C is called a ciphertext.
- D denotes a finite set called the alphabet. For example, D = f(0,1), the binary alphabet, Note that any alphabet can be encoded in terms of the binary alphabet. For example, since there are 26 binary strings of length eight, each letter of the English alphabet can be assigned a unique binary string of length eight.

Encryption and decryption transformations[8]

- K denotes a set called the key space. An element of K is called a key.
- Each element $e \in K$ uniquely determines a bijections from M to C, denoted by Ee. Ee is called an encryption function. Ee must be a bijections if the process is to be reversed and a unique plaintext message recovered for each distinct Ciphertext.
- For each dK, D_d denotes a bijections from C to M (*i.e.* $D_d: C \rightarrow M$). Dd is called a decryption function
- The process of applying the transformation Ee to a message $m \in M$ is usually referred to as the encryption of m.
- The process of applying the transformation D_d d to a Ciphertext c is usually referred to as decryption of c.
- An encryption scheme consists of a set f(Ee:$e \in K$) of encryption transformations and a corresponding set f(D_d:$d \in K$) of decryption transformations with the property that for each $e \in K$ there is a unique key $d \in K$ such that $D_d = Ee^{-1}$; that is, D_d (Ee(m)) = m for all $m \in M$.

To construct an encryption[9] scheme requires one to select a message space M, a Ciphertext space C, a key space K, a set of encryption transformations f(Ee :$e \in K$), and a corresponding set of decryption transformations f(D_d:$d \in K$). The keys e and d in the above discussion are referred to as a key pair and sometimes denoted by (e; d). It can be concluded that Symmetric or secret key cryptography involves the use of same key (e) for encryption and (d) decryption. Asymmetric or public key cryptography involve the use of one key for encryption (Ee), and another, different key D_d for decryption. No other key can decrypt the message, not even the original key used for encryption. This paper focus on the key generation for symmetric cryptography

Symmetric key generation : 

Firstly a bit explanation about pseudo-noise sequence is a periodic binary sequence generated by means of a feedback shift register, a general block diagram of which is shown in fig 1.

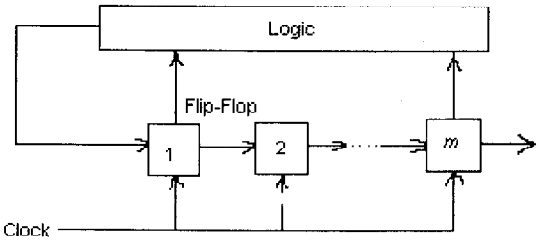
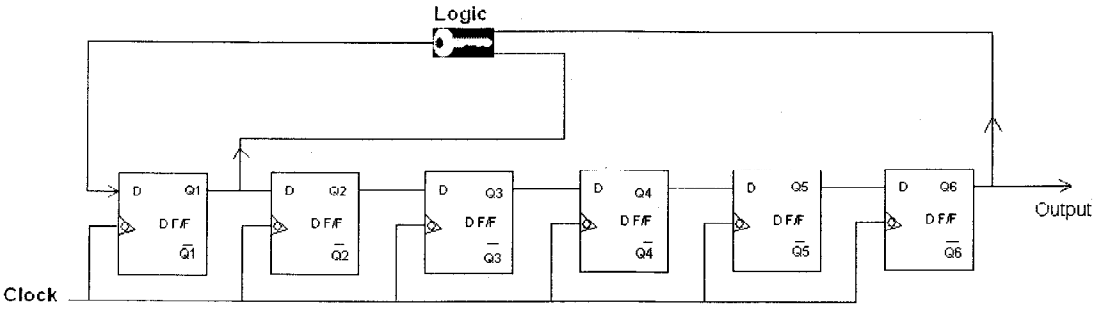


Figure 1 Feedback shift register

A feedback shift register consist of an ordinary shift register made up of m flip-flops and a logic circuit that are interconnected to form a multiloop feedback circuit. The flip-flops in the shift register are regulated by a single timing clock. At each pulse of the clock, the state of each flip-flop is shifted to the next one down the line. With each clock pulse the logic circuit computes a Boolean function of the states of the flip flops. The result is then fed back as the input to the first flip-flop, thereby preventing the shift register from emptying. The PN sequence so generated is determined by the length m of the shift register, its initial stage, and the feedback logic. With a total number of m flip-flops, the number of possible states of the shift register is at most 2^m. A feedback shift register is said to be linear when the feedback logic consists entirely of modulo-2 adders. In such a case, the zero state (e.g., the state for which all the flip-flops are in state 0) is not permitted. We say so because for a zero state, the input produced by the feedback logic would be 0, the shift register would then continue to remain in the zero state, and the output would therefore consist entirely of 0s. Consequently, the period of a
PN sequence produced by a linear feedback shift register with m flip-flops cannot exceed 2^m -1. When the period is exactly 2^m -1, the PN sequence is called a maximal-length-sequence or simply m sequence.

Maximal-Length Sequences :

Consider a maximal-length sequence requiring the use of a linear feedback shift register of length m = 6. For feedback taps we select the set (6, 1). The corresponding configuration of the code generation is shown in figure 2.



64-bit feedback shift register Fig. 2

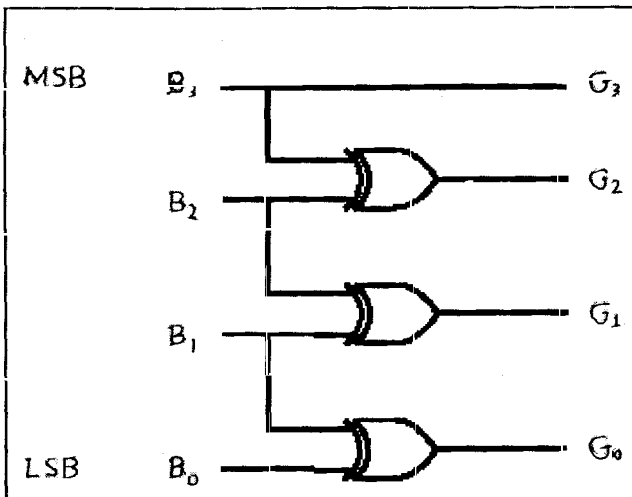
Assuming that the initial state is 100000, the evolution of maximal length sequence generated by this scheme is shown in table 1.where we see that the generator returns to the initial 100000 after 63 iterations, that is the period is 63.Cleary, the code generator of fig.2 having feedback taps (6, 1) has an advantage of having fewer feedback connections[10]. But the iteration pattern will be changed with different logic as show below

To generating the key following operations can be implemented in logic box shown in fig. 1 *i.e* Xoring, Xnoring, by using these operations inside the logic box different codes can be generated without loss of any information.

When 

=(Q1 X Q6') + (Q1' X Q6) =Q1 **XOR**Q6
= (Q1 X Q6') + (Q1' X Q6) =Q1 **XNOR** Q6
=(Q1 X Q6') + (Q1' X Q6) =Q1 **(2's complement)** Q6

The code converter method is also fast and reliable. Diagram shown in fig. 3 is Just for thew concept about 4 bit biary to Gray code conversion. But in lab 8 bit code converter is analyzed successfully. For 8 bit converter coding is shown as below in fig. 3.




```
library ieee;
use ieee.std_logic_1164.all
use ieee.std_logic_airth.all
entity b2g is
port(b: in bit_vector(0 to 7);
g: out bit_vector(0 to 7));
end b2g;
architecture b2g of b2d is
begin
g(0)<=b0 xor b1;
g(1)<=b1 xor b2;
g(2)<=b2 xor b3;
g(3)<=b3 xor b4;
g(4)<=b4 xor b5;
g(5)<=b5 xor b6;
g(6)<=b6 xor b7;
g(7)<=b7;
end b2g;
```

Figure 3.

KEY CODE GENERATED FOR SYMMETRIC KEY CRYPTOGRAPHY

0000011111101010110011011101101001001110001011110010100011000010

Security:

A fundamental premise in cryptography is that the sets M, C, K, f(Ee :e∈K), f(Dd :d∈K) are public knowledge. When two parties wish to communicate securely using an encryption scheme, the only thing that they keep secret is the particular key pair (e; d) which they are using, and which they must select. One can gain additional security by keeping the class of encryption and decryption transformations secret but one should not base the security of the entire scheme on this approach. History has shown that maintaining the secrecy of the transformations is very difficult indeed. Some security parameters can help us to generate more secure codes by considering that an encryption scheme can be broken by trying all possible keys to see which one the communicating parties are using (assuming that the class of encryption functions is public knowledge). This is called an exhaustive search of the key space. It follows then that the number of keys (*i.e.*, the size of the key space) should be large enough to make this approach computationally infeasible. It is the objective of a designer of an encryption scheme.

Ex. Let M = f(M1,M2,M3)and C = f(C1,C2,C3) There are precisely 3! = 6 bijections from M to C. The key space K = f(1, 2, 3, 4, 5, 6) has six elements in it, each specifying one of the transformations. Similarly in our case there are 256 messages and 256 Ciphertext. So there will be key space of 256! $=1.2 \times 10^{77}$ So it is follows that the number of keys *i.e.* the size of the key space should be large enough to make this approach computationally infeasible. Even more complexity can be increased by using permutation and combination during the processing of data.

Advantages :

(i) Symmetric-key algorithms are generally much less computationally intensive than asymmetric key algorithms. In practice, asymmetric key algorithm are typically hundreds to thousands times slower than a symmetric key algorithm

(ii) Once a symmetric key is known to all parties of the session, faster symmetric-key algorithms using that key can be used to encrypt the remainder of the session. This simplifies the key distribution problem, because asymmetric keys only have to be distributed authentically, whereas symmetric keys need to be distributed in an authentic and confidential manner.

Conclusion

The data could be sending confidentially using above techniques. Since data encryption is frequently the most time consuming part of the encryption process, the symmetric key scheme for key establishment is a small fraction of the total encryption process between A and B. Depending on the mode of usage, a private key/public key primary remains unchanged for considerable periods of time, *e.g.*, many sessions (even several years). Throughput rates for the public-key encryption methods are several orders of magnitude slower than the best known symmetric-key schemes. Key sizes are typically much larger than those required for symmetric-key encryption No public-key scheme has been proven to be secure (the same can be said for block ciphers). As the shift register length m, or equivalently, the period N of the maximal-

length sequence is increased, the maximal-length sequence become increasingly similar to the random binary sequence. Indeed, in a limit, the two sequences become identical when N is made infinitely large. However the price paid for making N large is an increasing storage requirement, which imposes a practical limit on how large N can actually be made.In code conversion no such limitation even size is incresed though it is fast and reliable process.

References

1. Handbook of Applied Cryptography by A. Menezes, P. van Oorschot and S. Vanstone.
2. Simon Haykins, Communication System, 4th ed., John Wiley & Sons (2003).
3. Charanjit S. Jutla, "Encryption Modes with Almost Free Message Integrity", Proc. Eurocrypt 2001, LNCS 2045, May (2001).
4. Simpson, R. E. "The Exclusive NOR (XNOR) Gate." §12.5.7 in Introductory Electronics for Scientists and Engineers, 2nd ed. Boston, MA: Allyn and Bacon, pp. 539 and 554 (1987)
5. "Origins of the Binary Code," F.G. Hearth, Scientific American, August 1972, pp. 76-83.
6. Fast twoapos;s complement VLSI adder desigr, Dobson, J.M.; Blair, G.M., *Electronics Letters*, Volume *31*, Issue 20, 28 Sep 1995 Page(s): 1721–1722.
7. Niels Ferguson; Bruce Schneier (2003). "Introduction to Cryptography: Attacks", in Carol A. Long: Practical Cryptography, Hardcover, Wiley Publishing Inc, 30-32.
8. CC Lo, YJ Chen - ... and Signal Processing, 1999 IEEE Pacific Rim Conference on, 1999 - ieeexplore.ieee.org
9. Cryptography—SAC 2003, 2003, to. appear. ELECTRONICS LETTERS 8th July 2004 Vol. 40 No. 14.
10. Simpson, R. E. "The Exclusive NOR (XNOR) Gate." §12.5.7 in Introductory Electronics for Scientists and Engineers, 2nd ed. Boston, MA: Allyn and Bacon, pp. 539 and 554 (1987).