Proposed Hybrid Approach of Stream Cipher Base on Selector of Encryption operation and Key Symmetric Translate

Dr. Alaa Kadim Farhan*

Received on:2/2/2011
Accepted on:20/6/2011

Abstract

A stream cipher is a method of encrypting text (to produce cipher text) in which a cryptographic key and algorithm are applied to each binary digit in a data stream, one bit at a time. This method is not much used in modern cryptography. The main alternative method is the block cipher in which a key and algorithm are applied to blocks of data rather than individual bits in a stream. One of the weaknesses in the stream cipher is to analyze the text through the use of encryption algorithms standard in encryption operations.

In this paper description a hybrid structure of encryption algorithm for stream cipher, this algorithm depends on a specific elements for the selection of encryption process (logical operation (XOR, AND) between the secret key and the plain text through encryption, decryption process. the specific intelligence elements choose from key. The secret key generated from random generator will be encrypted by RSA algorithm and sends with the encrypted text inside the packet

Keywords: random generator, intelligence selector, encryption, decryption.

اقتراح نموذج هجين للتشفير الانسيابي بالاعتماد على محدد عملية التشفير وتناقل المفتاح المتناظر

الخلاصة

أن التشفير الانسيابي هو أحد طرق تشفير النص الصريح لإنتاج النص المشفر والتي يطبق فيها المفتاح وخوارزمية التشفير لكل بت داخل سلسلة البيانات في وقت واحد. إن هذه الطريقة لا تستخدم كثيرا في التشفير الحديث الأسلوب البديل هو استخدام التشفير الكتلي حيث إن المفتاح وخوارزمية التشفير تطبق على بلوك من البيانات. من نقاط الضعف في التشفير الانسيابي هو قابلية التحليل للنصوص بالاعتماد على خوارزميات التشفير القياسية لعمليات التشفير.

في هذا البحث وصف هيكل مقترح لخوارزمية التشفير الانسيابي .إن هذه الخوارزمية تعتمد على المحدد يحتوي على عناصر خاصة تستخدم لترشيح العمليات المنطقية مابين المفتاح والنص الصريح من خلال عمليات التشفير وفك الشفرة عناصر المحدد الذكي يتم اختيارها من المفتاح المتولد . إن المفتاح السري يشفر بالاعتماد على طريقة(RSA) ويرسل مع النص المشفر داخل حزمه للمستلم.

Introduction

Cryptography and information security considered of important sciences in the world, especially after using the computer in these science. On other hand, the information and technology revolution and the armament race between the big powers give these priority because hese science play a big role in the since more field[1]. on other hand ,in all these cases there is a growing need for protection of information to

* Computer Science Department, University of Technology /Baghdad

https://doi.org/10.30684/etj.29.11.4
2412-0758/University of Technology-Iraq, Baghdad, Iraq

the safeguard economic interests, to prevent fraud and to ensure privacy [2].

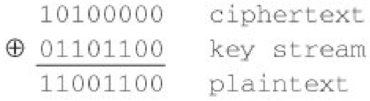
The Cryptographic systems are classified into two cryptosystems, private-key cryptosystem and public-key cryptosystem. Both are based on complex mathematical algorithms and are controlled by keys.

Public key cryptography provides a radical departure from all that has gone before, for thing public key algorithm are based on mathematical function rather than on substitution and permutation .more important public key cryptography is asymmetric involving the use of two separate keys. In contrast to symmetric conventional encryption .which uses only one key. The use of two keys has profound consequences in the area of confidentiality key distribution and authentication [3].

Stream Cipher Structure

A typical stream cipher encrypts plaintext one byte at a time; although a stream cipher may be designed to operate on one bit at a time or on units larger than a byte at a time. In the following example is a representative diagram of stream cipher structure. In this structure a key is input to a pseudorandom bit generator that produces a stream of 8-bit numbers that are apparently random. For now, we simply say that a pseudorandom stream is one that is unpredictable without knowledge of the input key. The output of the generator, called a key stream, is combined one byte at a time with the plaintext stream using the bitwise Exclusive-OR (XOR) operation. For example, if the next byte generated by the generator is 01101100 and the next plaintext byte is 11001100, then the resulting cipher text byte is

Decryption requires the use of the same pseudorandom sequence [3]:



```
  10100000   ciphertext
⊕ 01101100   key stream
  11001100   plaintext
```

Important element for design a stream cipher

1. The encryption sequence should have a large period. A pseudorandom number generator uses a function that produces a deterministic stream of bits that eventually repeats. The longer the period of repeat the more difficult it will be to do cryptanalysis. This is essentially the same consideration that was discussed with reference to the Vigenère cipher, namely that the longer the keyword the more difficult the cryptanalysis.
2. The key stream should approximate the properties of a true random number stream as close as possible. For example, there should be an approximately equal number of 1s and 0s. If the key stream is treated as a stream of bytes, then all of the 256 possible byte values should appear approximately equally often. The more random-appearing the key stream is, the more randomized the cipher text is, making cryptanalysis more difficult.
3. The output of the pseudorandom number generator is conditioned on the value of the input key. To guard

against brute-force attacks, the key needs to be sufficiently long. The same considerations as apply for block ciphers are valid here. Thus, with current technology, a key length of at least 128 bits is desirable [**4].**

Types of stream ciphers

A stream cipher generates successive elements of the key stream based on an internal state. This state is updated in essentially two ways: if the state changes independently of the plaintext or cipher text messages, the cipher is classified as a synchronous stream cipher. By contrast, self-synchronizing stream ciphers update their state based on previous cipher text digits [3,8].

Synchronous stream ciphers

A synchronous stream cipher is one in which the key stream are generated independently of the plaintext message and of the cipher text. The encryption process of a synchronous stream cipher can be described by the in the equation (1):

$$\begin{aligned} \sigma_{i+1} &= f(\sigma_i, k), \\ z_i &= g(\sigma_i, k), \qquad (1) \\ c_i &= h(z_i, m_i), \end{aligned}$$

where σ_0 is the initial state where i=0 and may be determined from the key k, is the next-state function, g is the function which produces the key stream z_i, and h is the output function which combines the key stream and plaintext mi to produce cipher text c_i The encryption and decryption processes are depicted in figure(1). The OFB (Output Feedback) mode of a block cipher:

(i) ***Synchronization requirements***. In a synchronous stream cipher, both the sender and receiver must be *synchronized* – using the same key and operating at the same position (state) within that key – to allow for proper decryption. If synchronization is lost due to cipher text digits being inserted or deleted during transmission, then decryption fails and can only be restored through additional techniques for re-synchronization. Techniques for re-synchronization include re-initialization, placing special markers

at regular intervals in the cipher text, or, if the plaintext contains enough redundancy, trying all possible key stream offsets.

(ii) ***No error propagation.*** A cipher text digit that is modified (but not deleted) during transmission does not affect the decryption of other cipher text digits.

(iii) ***Active attacks.*** As a consequence of property (i), the insertion, deletion, or replay of cipher text digits by an active adversary causes immediate loss of synchronization, and hence might possibly be detected by the decrypted.

Self-synchronizing stream ciphers

A self-synchronizing or asynchronous stream cipher is one in which the keystream is generated as a function of the key and a fixed number of previous ciphertext digits. The encryption function of a self-synchronizing stream cipher can be described by the equation(2):

$$\begin{aligned} \sigma_i &= (c_{i-t}, c_{i-t+1}, \ldots, c_{i-1}), \\ z_i &= g(\sigma_i, k), \\ c_i &= h(z_i, m_i), \end{aligned} \qquad (2)$$

where σ_0 = (c_{-t}; c_{-t+1}; …. ; $c_{-}1$) is the (non-secret) *initial state*, k is the *key*, g is the function which produces the *key stream* z_i, and h is the *output function* which combines the key stream and plaintext mi to produce cipher text c_i. The encryption and decryption processes are depicted in Figure(2). The most common presently-used self synchronizing stream ciphers are based on block ciphers in 1-bit cipher feedback mode

(i) ***Self-synchronization***. Self-synchronization is possible if cipher text digits are deleted or inserted, because the decryption mapping depends only on a fixed number of preceding cipher text characters. Such ciphers are capable of re-establishing proper decryption automatically after loss of synchronization, with only a fixed number of plaintext characters unrecoverable.

(ii) ***Limited error propagation.*** Suppose that the state of a self-synchronization stream cipher depends on t previous cipher text digits. If a single cipher text digit is modified (or even deleted or inserted) during transmission, then decryption of up to t subsequent cipher text digits may be incorrect, after which correct decryption resumes.

(iii) ***Active attacks.*** Property (ii) implies that any modification of cipher text digits by an active adversary causes several other cipher text digits to be decrypted incorrectly, thereby improving (compared to synchronous stream ciphers) the likelihood of being detected by the decrypted. As a consequence of property (i), it is more difficult (than for synchronous stream ciphers) to detect insertion, deletion, or replay of cipher text digits by an active adversary. This illustrates that additional mechanisms must be employed in order to provide data origin authentication and data integrity guarantees.

(iv) ***Diffusion of plaintext statistics.*** Since each plaintext digit influences the entire following cipher text, the statistical properties of the plaintext are dispersed through the cipher text. Hence, self-synchronizing stream ciphers may be more resistant than synchronous stream ciphers against attacks based on plaintext redundancy [5, 7].

Proposed Hybrid Approach of Stream Cipher Structure

The mean idea in the proposed approach is send the encryption key (**Synchronization**) with the encryption message session can increase the difficult analysis of cipher text. the secret key used in encryption method (stream cipher) was encrypted by RSA method depend on the public key of the receive ,when the receiver part can decryption by RSA to get the secret key, after get the secret key used it to decryption cipher message. .

The structure used in proposed algorithm is difference from stream cipher structure, that is, a bit-stream is divided into several fixed units called selector.

The selector are select the appropriate cipher operation (XOR, AND) between plain text and the key. So this increase the complexity of code analyzing, the selector is a stream of bits (0, 1) that are generated depend on (elements selector algorithms), where the bit (0) value refer to AND operation and bit (1) value to the XOR operation.

The generated of the random key by random generation algorithm that contains two elements (ADD, CARRY), which have more than one registers with different length to get a largest number of attempts. The results of proposed algorithm have two sections the first one is a cipher text and the second is cipher key. Where the original key is encrypted by receiving public key depends on RSA algorithm.

Algorithms of Hybrid Approach of Stream Cipher

The Hybrid approach has number of algorithms can be description in the following:

1-Instillation Algorithm

- Select the secret key from Random key Generator (used to stream cipher).
- Select the keys (E,D,N) for sender and receiver to encryption and decryption secrete key(the secrete used in stream cipher) where E=encryption key ,D=decryption Key and N=mod.

2-Random Key Generator Algorithm

Input: registers and initial values

Output: secret key random

1. Select two registers m, n and the GCD(M,N)=1 to max priories.
2. Select two LFSRs function must primitive polynomial.
3. By the in the equation(3) to calculate output and carry:

   $Z_i=A_i+B_i+C_{i-1}$

   $C_i= A_i+B_i+(A_i+B_i) C_{i-1}$ When $C_{i-1}=0$

3- Elements Selector Algorithm

Input: random key from generator (k)

Output: array of elements selector to select the type encryption operation

1. Copy the original key into key1.
2. Merge between two positions by XOR operator.
3. Convert the result from 1- D array to 2-D array.
4. Rotate the result array (2-D).
5. Convert from 2-D to 1-D array.
6. Use as Selector.

The following figure (4) describes the algorithm:

And, the work of elements selector can describe by in the figure (5):

4- Encryption Algorithm

Input: Plain Text, Keys (RSA, Stream cipher)

Output: Package (Cipher Text, Cipher Key).

1. Through the secret key choose the selector values (sequence of bit from key) used to operation (XOR-AND) between key and

Plain text in encryption or decryption operation.
2. Encryption plain bits depend on selector (used XOR-AND) and make cipher text.
3. Encryption the secrete key by used encryption key to receive.
4. Cipher Key=RSA(Key, E-receive) Mod N
5. Configure the package (Cipher Text, Cipher Key).

Figure (6) shows the main processes:

5- Decryption Algorithm

Input: Cipher Text, Keys (RSA, Stream Cipher)

Output: Plain Text

1. Select the part (Cipher Text)
2. Decryption the secrete key Key=RSA(Cipher Key, D-receiver)
3. Through the secret key choose the selector values (sequence of bit from key) used to operation (XOR-AND) between key and Plain text in encryption or decryption operation.

Decrypt the Cipher text by secret key and get Plain text. As illustrate in figure (7)

7. Cryptanalytic Tools for Stream Cipher

In our treatment of cryptanalysis of stream ciphers in the section, one important cryptanalytic tool techniques will be frequently used. This technique is based on a combination of algebra, statistics, and numerical techniques. This section, will describe the basic technique that have had the greatest influence in the cryptanalysis of stream ciphers.

7.1 Matrix Method

Meyer et al [10], have demonstrated a method called Matrix method (or Algebraic method) of breaking an “N-stage” LFSR given (2N) consecutive bits of known plaintext. The basic method is to set up the matrix equation: K = MC, where,

M: the matrix of successive shifts of the first n bits of plaintext.

C: the unknown matrix of switch states.

K: the key matrix.

Thus the switch values can be obtained by inverting M and solving: $C = M^{-1}K$

Therefore, the entire key will be known completely. Although finding the inverse of matrix is not trivial (the time taken to find the inverse matrix over GF(2) is proportional to O (N3)), it is a straightforward Process. But this method is weak, since if less than 2N consecutive bits are known this may not be enough to determine the entire sequence. Beker and Piper, showed two methods of breaking an “N-stage” LFSR given (2N) bits of the known plaintext sequence where the bits are not necessarily consecutive. These methods are based on trying every possibility for filling unknown entries in the sequence, and for each one they merely use the techniques of the matrix equation, so that, they involve a mixture of guessing and solving simultaneous equations. The choice between these two methods is usually made by seeing which will require fewer trials.

These methods apply to proposed hybrid method and the result difficult to break the proposed method.

Conclusions

Several conclusions are reached through the working the system steps. The following items represent the important conclusions which are drawn from the proposed system:

Research is focused on two main points the first point, the system selector will exchange the process of generating the encrypted message and the second point is encrypted stream cipher key depends on public key methods the complexity of stream cipher key using public key method with the message.

The two logical operations AND, and OR provides a good random operation that will increase the complexity of proposed system analyzing.

Tests conducted on the system are proved the power of the system in terms of security, as well as random operation will adding to the strength of the proposed system compared to the as for the time it was very close to the traditional system, this is useful in relation to the strength of the proposed system

References

1- Baker H.& piper F.,"***Cipher System:The Protection Of Communication***", Northwood Publiction,U.K,1982.

2- Stalling W.,"***Cryptography and Network Security , Principle and Practice***", Addenison Wesley,1999.

3- Oorschot M. P. van, and Vanstone S., "***Handbook of Applied Cryptography***", CRC Press, 1996.

4- Beker, H & Piper, F," ***The Protection of Communications***". Wiley-Interscience. p. 212, 1982.

5- Babbage S. & Dodd M., "***The MICKEY Stream Ciphers***". Lecture Notes in Computer Science, 4986:191,209, 2008. http://www.ecrypt.eu.org/stream mickeypf.html

6- Ismail K,"***Intelligence cryptanalysis tool practical swarm***", PhD Thesis, Computer Science, University of Technology,2009

7- Bernstein D." ***The Salsa20 family of stream ciphers***". Lecture Notes in Computer Science, 4986:84,97, 2008 http://www.ecrypt.eu.org/stream/salsa20pf.html.

Schneier B., "***Applied Cryptography, Protocols, Algorithms, and***

8- ***Source Code .C Language***", John Wily and Sons, Inc, U.S.A, 1996..

Meyer C, Tuchmab W., "***Pseudo-Random Codes Can be Cracked***

9- ***Electronic Design***", Vol. 23, pp. 74-76, 1972..

Meyer C, Tuchmab W., "***Pseudo-Random Codes Can be Cracked***",

10- Electronic Design, Vol. 23, pp. 74-76, 1972

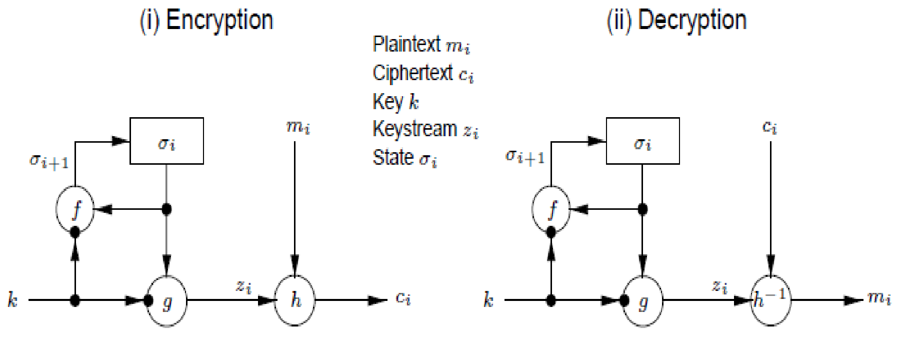


Figure (1) Synchronization Stream cipher

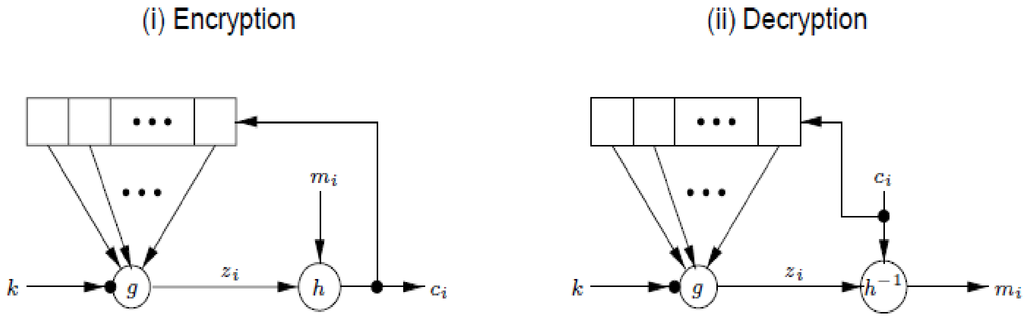


Figure (2) Self-Synchronization Stream Cipher

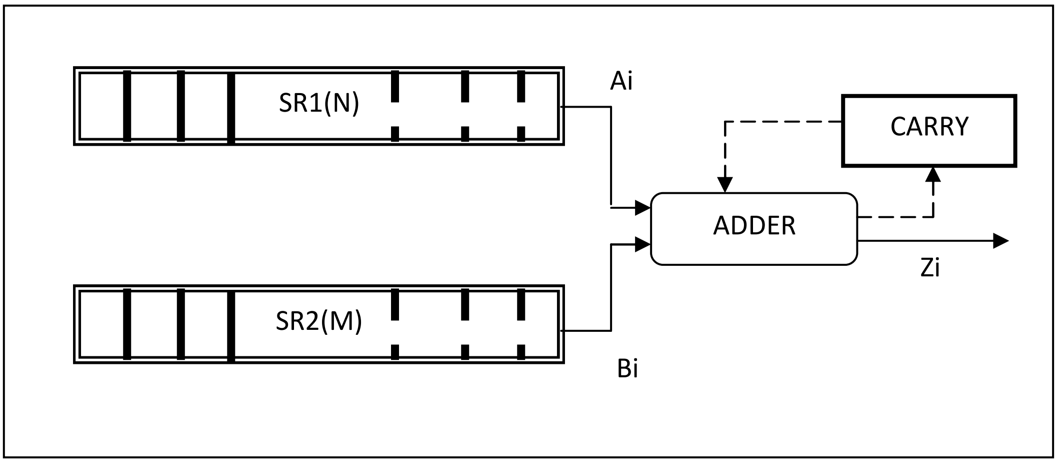


Figure (3) Random Generator

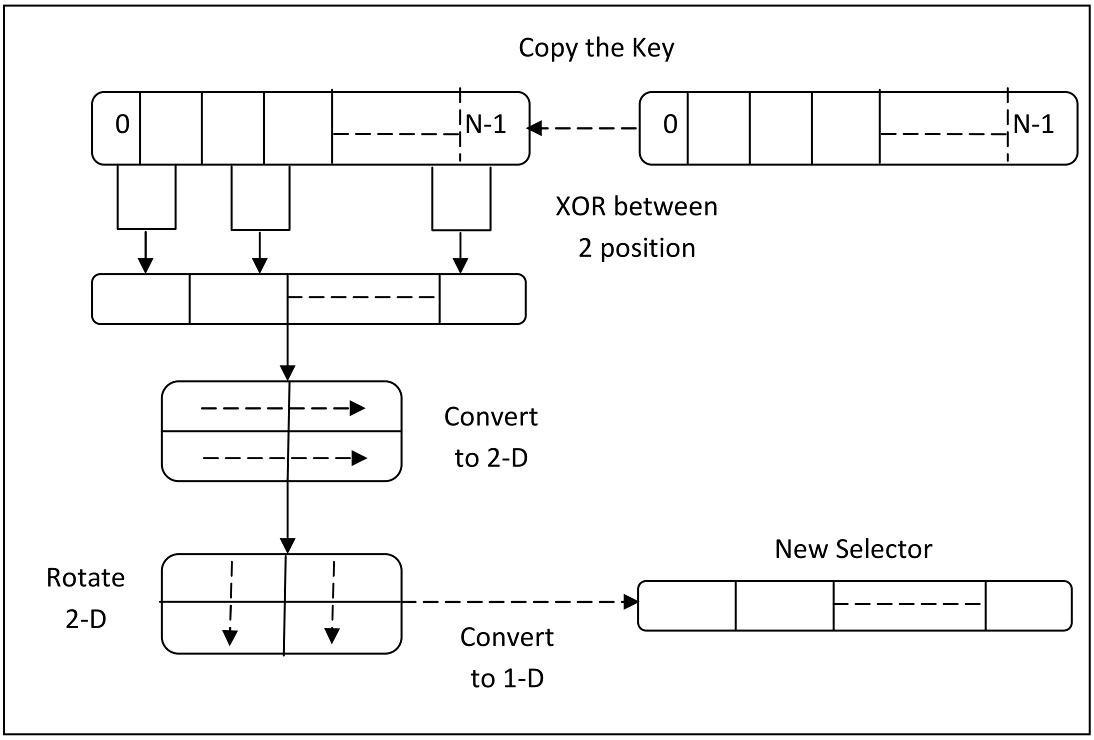


Figure (4) Elements Selector Algorithm

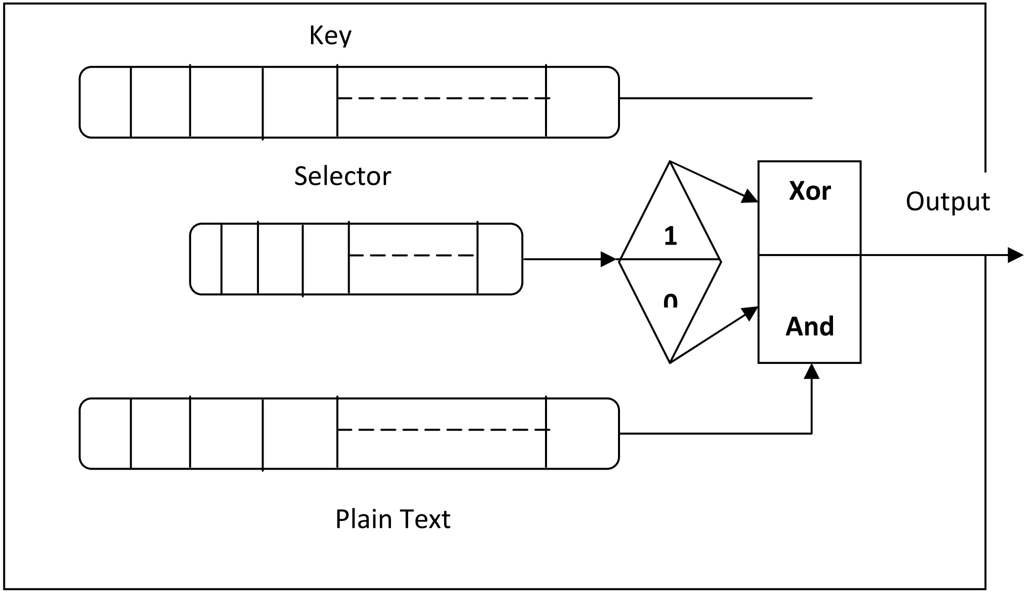


Figure (5) Operation By elements selector

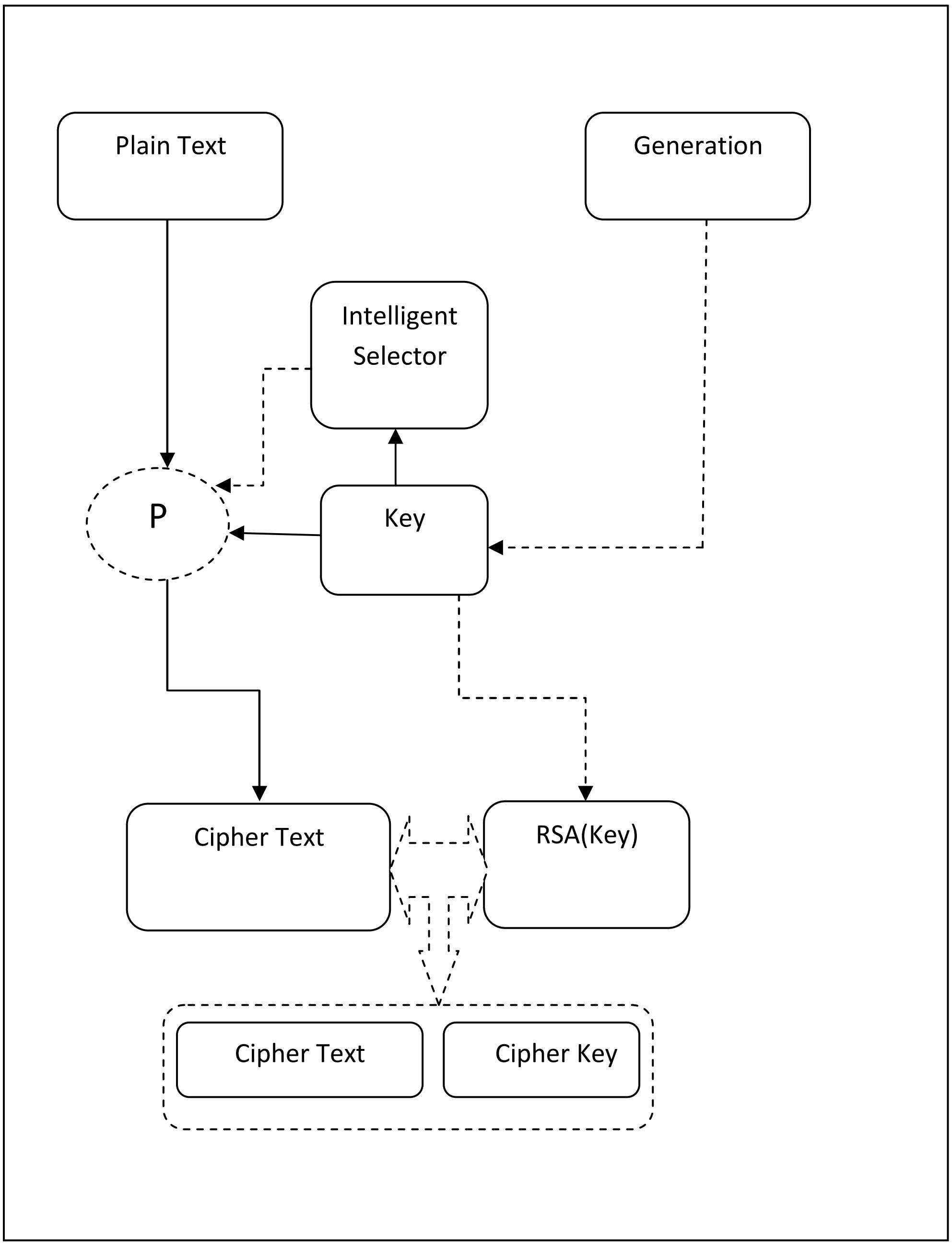


Figure (6) Encryption Stream Cipher

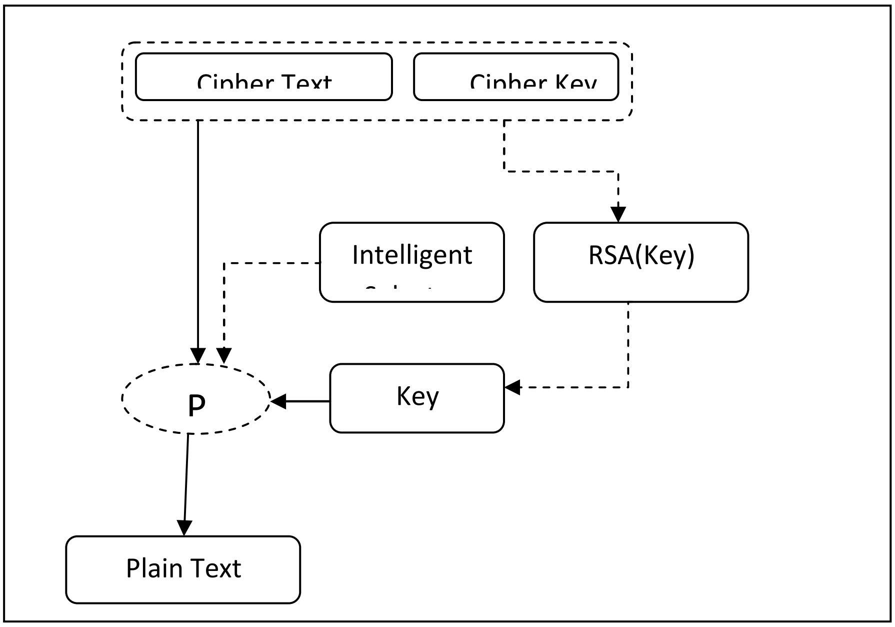


Figure (7) Decryption Stream Cipher