Design New Block Cipher Algorithm With New Concept

Dr. Saad K. Majeed * & Dr. Maki Mahdi Abdulhasan*

Received on: 21/10/2009
Accepted on: 3 / 6 /2010

Abstract

There are several techniques to safeguard the security of the information stored in the computers or transmitted by networks; the most powerful tool is encryption. Encryption provides confidentiality for information; additionally encryption can be used to achieve integrity and availability. In this paper we design new Block cipher algorithm with new concepts that are " encryption keys updating " where this result a new approach not used in all the known block cipher algorithms which is " The same key, encrypts the same plaintext multiple times and produces different cipher texts ", also user does not inform anything about encryption keys, that giving the proposed algorithm protection from the cheating of user and the secrecy of the encryption keys remain only in Key Management Center (KMC), finally, this algorithm work approximately as one-time pad.

تصميم خوارزمية تشفير كتلي بمبدا جديد

الخلاصة

هناك عِدّة تقنيات لحِماية أمنِ المعلوماتِ المَخْزُونة داخل الحاسباتِ أو المتناقلة عبر الشبكات، لكن الأداة الأقوى في ضمان الأمنِ للمعلومات هي التشفيرُ. حيث يوفرُ التشفيرُ سريّةً للمعلوماتِ؛ إضافة إلى ذلك التشفير يمكن أن يُستَعملَ لتحقيق السلامةِ والاتاحية.

في هذا البحث نحاول تصميم خوارزمية تشفير كتلي بمفهوم جديد "التغير الذاتي لمفاتيح التشفير"حيث هذا المفهوم ينتج عنه مبدأ جديد غير مطروق في كافة خوارزمية التشفير الكتلي المعروفة هذا المبدأ هو "استخدام نفس مفتاح لتشفير نفس النص الواضح عدة مرات لينتج نصوص مشفرة مختلفة", كذلك في هذه الخوارزمية المستخدم لا يملك أية معلومات عن مفاتيح التشفير مما يكسب الخوارزمية حماية ضد خداع المستخدمين

كذلك سرية مفاتيح التشفير تبقى فقط لدى مركز أدارة المفاتيح , أخيرا, هذه الخوارزمية One-Time Pad تعمل تقريبا كخوارزمية.

1- Terminology

A cipher is a pair of algorithms which creates the encryption and the reversing decryption, the detailed operation of a cipher is controlled both by the algorithm and, in each instance, by a **key**. Keys are important, as ciphers without variable keys are trivially breakable and therefore less than useful for most purposes. This is all shown in Figure (1).

2- Block cipher

In cryptography, a block cipher is a symmetric key cipher which operates

*Computer Science Department, University of Technology/Baghdad

https://doi.org/10.30684/etj.28.19.9
2412-0758/University of Technology-Iraq, Baghdad, Iraq

This is an open access article under the CC BY 4.0 license http://creativecommons.org/licenses/by/4.0

on fixed-length groups of bits, termed blocks, with an unvarying transformation. When being encrypted, a block cipher might take (for example) a 128-bit block of plain text as input, and output a corresponding 128-bit block of cipher text. The exact transformation is controlled using a second input-the secret key. Decryption is similar: the decryption algorithm takes, in this example, a 128-bit block of cipher text together with the secret key, and yields the original 128-bit block of plain text. Figure (2) shows the encryption and decryption operations in block cipher.

A block cipher consists of two paired algorithms, one for encryption, **E**, and another for decryption, $\mathbf{E^{-1}}$. Both algorithms accept two inputs: an input block of size n bits and a key of size k bits, yielding an n-bit output block. For any one fixed key, decryption is the inverse function of encryption, so that: $\mathbf{E^{-1}_k (E_k (M)) = M}$ **,** for any block M and key k.

2-1 Block cipher properties

- **Iterated block ciphers**

Most block ciphers are constructed by repeatedly applying a simpler function. This approach is known as iterated block cipher. Each iteration is termed a round, and the repeated function is termed the round function; anywhere between 4 to 32 rounds are typical. [2,7]

- **Feistel cipher (feistel network)**

A large proportion of block ciphers use schemes, including Data Encryption Standard (DES). The Feistel structure has the advantage that encryption and decryption operations are very similar, even identical in some cases, requiring only a reversal of key schedule.[

- **Substitution Boxes (S-Boxes)**

The S-Box is simply a substitution: mapping of m-bit inputs to n-bit outputs, this S-Box is called "m n S-Box".

S-Boxes are generally the only nonlinear step in an algorithm, they are what give a block cipher its security. The bigger they are the better.

- **Permutation Boxes(P-Boxes)**

The permutation is required to provide the necessary diffusion of the outputs from the S-Boxes over as many of the S-Boxes inputs in the next layer as possible. Any uniquely reversible transformation from any set into itself is called Permutation (or sometimes P-Box). This permutation maps each input bit to an output position, no bits are used twice and no bits are ignored.

2-2 Cryptanalytic techniques against block cipher:

Analyzing block cipher is often harder than that of stream cipher. Since the attacks against block cipher are largely theoretical. [3, 6]

- **Differential cryptanalysis**

Differential cryptanalysis looks specifically at cipher text pairs: pairs of cipher texts whose plain texts have particular differences. It analyzes the evolution of these differences as the plain texts propagate through the rounds. [3,4,5]

Linear cryptanalysis

- This attack uses linear approximations to describe the action of a block cipher. This means that if we (**XOR**) some of the plain text bits together, (**XOR**)

PDF created with pdfFactory Pro trial version www.pdffactory.com

some cipher text bits together, and then (**XOR**) the result a single bit will be obtained that is the (**XOR**) of some of key bits. This is linear approximation that is used to guess the values of the key bits.

- **Related-key cryptanalysis**

A related key attack is one where the attacker learns the encryption of certain plain text not only under the original (unknown) key (**K),** but also under some derived keys [**K' = f(K)**].

3- Proposed Block cipher algorithm

An algorithm is unconditionally secure if, no matter how much cipher text a cryptanalyst has there is not enough information to recover the plain text from that cipher text. The primary objective of designing the proposed Block cipher algorithm (***MAK-128***) is to ensure that a cryptanalyst cannot obtain sufficient information to succeed in an attack in significant time than the expected cost of an exhaustive key search.

3-1 MAK-128 algorithm description

Figure (3) shows the high level structure of proposed Block cipher algorithm (MAK-128), this algorithm of Feistel network encrypts and decrypts 128-bit data with variable keys of size 192 bytes (1536 bits). the former consists of 16 cycle iterations of (substitution and permutation). MAK-128 algorithm has Encryption keys file contains 128 keys of length of 197 bytes. The first five bytes from each key called ***Address key*** (**AK**) are the first group in the encrypted message, and in the decryption process, using the (AK) as pointer to the key that is used in encryption process and the remaining 192 bytes called ***Basic key*** (**BK**) are used by MAK-128 algorithm to encryption and decryption processes. In the encryption process, the user chooses a number (0 through 127) randomly, this number determines the encryption key that will be used by MAK-128 algorithm, after ending each encryption and decryption process this key is remedied and returns to the same position in encryption keys file.

The MAK-128 algorithm has the following features:

a- It is 16 round cipher with block length of 128 bits and variable keys of length 192 bytes.

b- Identical process for encryption and decryption with the same key.

c- The same key, encrypts the same plaintext multiple times and produces different cipher texts. This will increase the length of life-cycle of cryptographic keys.

d- Substitution (in S-Box) consists of random numbers, not fixed as DES algorithm.

e- Permutation is key-dependent such that the MAK-128 can be protected against linear and differential cryptanalysis.

The encryption process of the MAK-128 algorithm can be described as:

The input block (128-bit) divided into four chunks each chunk 32-bit.

1- Plain text M(128-bit) = (left L_{11}(32-bit) , L_{21}(32-bit) ; right R_{11}(32-bit) ,R_{21}(32-bit))

2- For i= 1,2, …. 16 do

PDF created with pdfFactory Pro trial version www.pdffactory.com

2-1 $R1_{i+1} = L_{1i}$
$R_{2\,i+1} = L_{2i}$

3-2 $L1_{i+1} = R_{1i}$ $(F\,[L \oplus 1_i\; K_{1i}]\,, K\oplus)$
$L_{2\,i+1} = R_{2i}$ $(F\,[L2 \oplus_i\; K_{2i}]\,, K_2\oplus)$

4- Cipher text C (128-bit) ($L_{1\,16}$(32-bit) , $L_{2\,16}$(32-bit) ,$R_{1\,16}$(32-bit) ,$R_{2\,16}$(32-bit))

The decryption process is the same as the encryption process, except the round sub keys are applied in reverse order.

After each encryption and decryption process the sub keys in key scheduling shift vertically and horizontally to the right one step, and then return to the same position in the Encryption keys file.

F-function

F-function consists of three parts: Substitution Boxes (***S-Boxes***), Permutation Boxes (***P-Boxes***) and **XOR-net**. This function has 8 S-Boxes and 8 P-Boxes with 2 32-bit input (In_1 , In_2), 2 32-bit output (Ou_1 , Ou_2). There are two sets of S-Boxes (S_1 , S_2), each set consists of 4 S-Boxes, the input In_1 (32-bit) is broken into four 8-bit chunks, and each chunk becomes the input to different S-Box, the first 8 bits go into the first S-Box, the second 8 bits go into the second S-Box and so on, so the input In_2 (32-bit) is also broken into four 8-bit chunks and each chunk becomes the input to 4 S-Boxes in the second set of S-Boxes (see Figure (4)).

There are 8 S-Boxes inthe two sets of S-Boxes each S-Box is a random number (0 through 255), the random numbers are generated according to specific properties and for each application there are different S-Boxes. Table (1) shows contents of S-Boxes.

The second part of F-function is P-Boxes, which are made to shift the output of S-Boxes one step depending on sub key (per-key). The main purpose of the P-Boxes is to guarantee the avalanche effect at only one round and prevent any attack on this function especially the related key attack that is described in chapter two.

P-Boxes part consists of two sets of P-Boxes (P_1 , P_2), each set consists of four P-Boxes, each one has 8 8-bit input that is output of S-Boxes and 2 32-bit output (Ou_1 , Ou_2). P-Boxes are permutation key-dependent has 32-bit sub key (per-key), the first P-Box in the first set of P-Boxes receive the 8-bit output of the first S-Box in the first set of S-Boxes and the second P-Box in the same set of P-Boxes receives the 8-bit output of the first S-Box in the first set of S-Boxes and so on. 32-bit per-key is divided into 2 16-bit (Rperkey1 , Rperkey2), Rperkey1 is for P_1 broken into four 4-bit chunks, the first chunk goes to the first P-Box in the P_1 and the second chunk goes to the second P-Box in the P_1, and so on, so 16-bit Rperker2 is also broken into four 4-bit chunks, each chunk goes to the corresponding P-Box in the P_2. The first bit of each 4-bit chunk determines that the input of P-Box is shifted to left or to right (0 to left and 1 to right), and the remainder three bits determine the number of steps that the input of P-Box are shifted to the right (000 no shift, 001 shift one step, …, 111 shift seven steps).

PDF created with pdfFactory Pro trial version www.pdffactory.com

The third part of F-function is a ***XOR-net*** that is network of Xor's (modulo-two addition), the output of each set of P-Boxes becomes the input to the XOR-net, and the outputs of each network are recombined into a 32-bit. The main purpose of the XOR-net is to guarantee the avalanche effect at only one round and prevent any attack on this function especially the related key attack (see Figure (5)). The XOR-net can be described as:

$$\mathbf{W} = \mathbf{X}_1 \qquad \mathbf{X}_4 \; \oplus$$

$$\mathbf{Y}_1 = \mathbf{W} \qquad \mathbf{X}_2 \oplus$$

$$\mathbf{Y}_3 = \mathbf{Y}_1 \qquad \mathbf{X}_3 \oplus$$

$$\mathbf{Y}_4 = \mathbf{Y}_3 \qquad \mathbf{X}_4 \; \oplus$$

$$\mathbf{Y}_2 = \mathbf{Y}_4 \qquad \mathbf{X}_2 \; \oplus$$

Example 1:

Let the input of F-function be the following:

In_1 : 00000000 00000000 00000000 00000000
In_2 : 00000000 00000000 00000000 00000000

And the per-key is:

Per-key: 0010 1101 0101 1100 1001 0100 1011 0001

The output of F-function after XOR-net according to Figure (4-5) is:

Ou $_1$: 10101100110011100000011110010110
Ou $_2$: 10110010000101110010111001010010

3-1-1 The Key Scheduling

In the MAK-128 algorithm, the key schedule as shown in Table (2) consists of look-up table which contains sixteen 96-bit sub keys (extracted from encryption keys file depending on random number (0 through 127) where the user chooses it in the encryption process and depending on AK in the decryption process), and shift function.

Each 96 bits sub key is used for one round, the first sub key to the first round, and the second sub key to the second round and so on.

Each sub key is divided into 3 32-bit k_{ij} , where i= 1,2,3 , and j= 1,2,3,…,16.

After ending each encryption and decryption process, the shift function is performed where the contents of look-up table is shifted vertically and shifted horizontally to the right one step then returns to the SSMSI library in the same position.

New Approach

Imagine an encrypted data link where the user wants to change keys daily. Sometimes it's a pain to distribute new keys daily. An easier solution is to generate new keys from the old keys; this is called ***keys updating***.

Key updating works, but remember that the new keys are only as secure as the old keys were. As stated in the abstract of this paper, that MAK-128 must come out as a new approach (Encryption keys updating) that combines between the characteristics of the Block ciphers and the conditions of the perfect secrecy, the Block cipher algorithms operate on scheme that is the same plain text produce as the

PDF created with pdfFactory Pro trial version www.pdffactory.com

cipher text using the same key multiple times.

MAK-128 operates on approach that is the same plain text that has produced different cipher texts using the same key that is acquired from Encryption keys file multiple times.

3-1-3 The Key Scheduling Complexity

In any good cryptosystem the security of the algorithm must reside in the key, and the security should not depend on the secrecy of the algorithm.

For example DES algorithm has a 56-bit key, calculating the complexity of a brute-force attack is easy, because there are 2^{56} possible keys in DES.

In MAK-128 algorithm, look-up table contains sixteen 96-bit sub-keys then the exhaustive key search requires

$$\mathbf{16 \times 2^{96} = 2^{4} \times 2^{96} = 2^{100}}$$

3-2 Analysis of MAK-128 algorithm

This section is for assessing whether the algorithm is according to the design criteria, and then analyze the immunity against various possible attacks is analyzed.

- **F-function analysis**

F-function in MAK-128 algorithm has two sets of S-Boxes, 4 S-Boxes in each set, with two 32-bit inputs and two 32-bit outputs. The content of S-Boxes is random numbers and changed from application to another. Once two sets of S-Boxes have been constructed for a given implementation they are fixed all the time.

F-function also has two sets of P-Boxes, 4 P-Boxes in each set. P-Boxes are key-dependent with two 32-bit input and two 32-bit output under effect 32-bit per-key. P-Boxes are permutations that map two 32-bit inputs to two 32-bit outputs under the influence of the 32-bit per-key.

- **Weak keys**

MAK-128 algorithm does not contain any weak keys, all keys in Encryption keys file are generated by KMC, assuredly, they are equally strong and random , besides, the user does not inform any thing about these keys, only he chooses a random number (0 through 127) every time he wants to encrypt a message.

- **Differential cryptanalysis**

In MAK-128 algorithm the changing of one bit input of F-function results in changing all output bits (as shown in example 2), besides MAK-128 algorithm can not obtain the characteristics whereas the basic key that is used in the each encryption process is completely different from the other, therefore, the MAK-128 algorithm is resistant to differential cryptanalysis.

Example 2

If the eighth bit in the first chunk in (In_1) is changed, and the eighth bit in the first chunk in (In_2) is changed in example (1) as :

In_1 : 00000001 00000000 00000000 00000000

In_2 : 00000001 00000000 00000000 00000000

And the same per-key is used:

0010 1101 0101 1100 1001 0100 1011 0001

The output of F-function after the XOR-net and according to Figure (5) is:

PDF created with pdfFactory Pro trial version www.pdffactory.com

Ou 1 : 000101110111010110111100001011010

Ou 2 : 101110010001110000100101010110010

- **Linear cryptanalysis**

The MAK-128 algorithm has a large number of encryption keys in Encryption keys file and a large number of sub keys and per-keys in look-up table, where each round requires two 32-bit sub keys and 32-bit per-key, besides, these keys are changed after ending each encryption and decryption process, therefore it is computationally infeasible to find the sub keys and per-keys and to decide which sub key or per-key is right to find any linear relationship, then MAK-128 algorithm is resistant to linear cryptanalysis.

- **Related-key cryptanalysis**

In MAK-128 algorithm the key schedule is one-way where attacker cannot gain any information about the keys in SSMSI library or other round keys if he recovers a few round keys, therefore the MAK-128 algorithm is resistant to any related-key attack.

The table (3) showed some of the characteristics of some block cipher algorithms already found and the cryptanalysis applied to each algorithm:

4- Conclusions

- This paper includes new cryptographic algorithm (Block cipher called MAK-128) with new concepts are differ from the known algorithms.
- The proposed Block cipher algorithm protected against the cheating of the authorized user by the secrecy of the encryption keys saved only by Key Management Center (KMC).

References

[1]Wikipedia the free encyclopedia: retrieved on 7/10/2008 from Http://en.wikipedia.org/wiki/Cryptoghrapy.

[2]O. Goldreich, "Foundations of Cryptography", Department of Computer Science and Applied Mathematics, Weizmann Institute, 1995.

[3]C. H. Lee, "Cryptographic Techniques of E-commerce", BY City University of Hong Kong, 1999.

[4]Dr. Adam L. Young, Dr. Moti Yung, "Malicious Cryptography", Wiley publishing, Inc., Indianapolis, Indiana, 2004.

[5]S. Bono, M. Green, "Security Analysis of a cryptographically", In proceedings of the USENIX security symposium, August 2005.

[6]Dr. Adam L. Young, Dr. Moti Yung, "Malicious Cryptography", Wiley publishing, Inc., Indianapolis, Indiana, 2004.

[7]R. Oppliger, "Contemporary Cryptography", Artech House, Inc., 2005.

[8]Wikipedia the free encyclopedia: retrieved on 7/10/2008 from Http://en.wikipedia.org/wiki/Cryptoghrapy.

[9]M. Sudul,"Modern Cryptography: Theory and Practice", Prentice **Hall,** Inc. , 2004.

PDF created with pdfFactory Pro trial version www.pdffactory.com

[10]J. Talbot and Dominic Welsh, "Complexity and Cryptography", Cambridge University press 2006.

Table (1) the content of S-boxes (S_1,S_2) of MAK-128 algorithm (S_1)

address	S-Box 1	S-Box 2	S-Box 3	S-Box 4
0	01011001	00011011	01011101	00100011
1	10110111	10100011	01110100	01011110
….	….	….	….	….
255	01001101	00000010	11111010	11011110

(S_2)

address	S-Box 1	S-Box 2	S-Box 3	S-Box 4
0	00010111	01010100	11100100	00111110
1	10000000	01000001	11100010	00110001
….	….	….	….	….
255	11010100	10001110	00100001	11001100

PDF created with pdfFactory Pro trial version www.pdffactory.com

Table (2) sub keys schedule of MAK-128 algorithm

address	96-bit sub key
0	01001000111010…………………..0001010
1	01101001011110…………………..1101101
…	…………………………………………….
15	11101001111110………………….1000100

algorithm	Block size	Key size	N# of rounds	S-box	Sub keys derivation	Permeation form	cryptanalysis
DES [1,2]	64-bit	56-bit	16 round	8 6x4 bit S-box fixed table	Direct from main key and fixed	fixed	Broken by differential, linear and related key attack
NEWDES [5,6]	64-bit	120-bit	17 round	nothing	Direct from main key and fixed	fixed	Broken by related key attack
GOST [3,4]	64-bit	256-bit	32 round	8 4x4 bit S-box random number (0-15)	Direct from main key and fixed	fixed	Broken by related key attack
AES [9,10]	128-bit	128, 192 or 256-bit	10, 12 or 14 depend on key size	Byte multiplication and XOR table	Direct from main key and fixed	fixed	Related key attack can break up to 5 rounds of 256-bit, a chosen plaintext attack can break 8 rounds of 192-bit and 256-bit, and 7 rounds of 128-bit
MAK-128	128-bit	1536-bit	16 round	8 8x8 bit S-box random number (0-255)	Separated from main key and modified	Key-dependent	unknown

Table (3) characteristics of some block cipher algorithms

PDF created with pdfFactory Pro trial version www.pdffactory.com

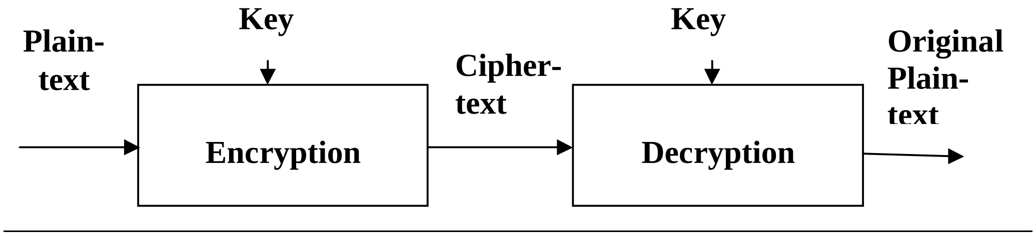


Figure (1) the encryption and decryption operations

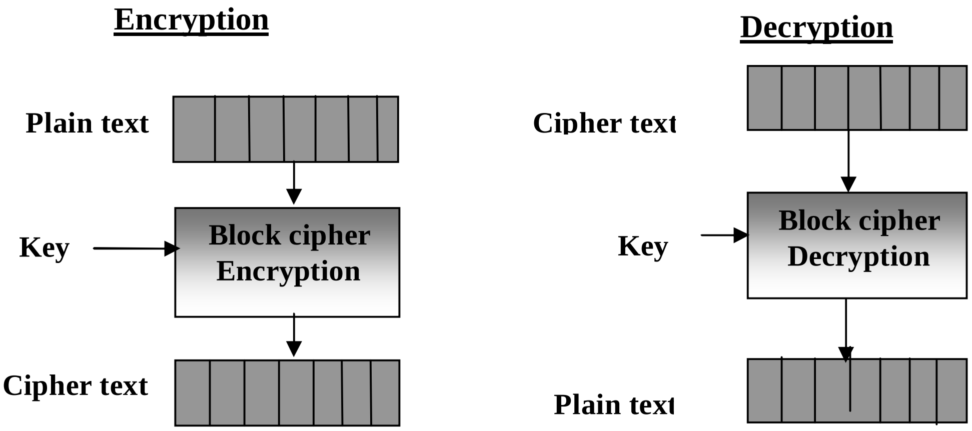


Figure (2) the encryption and decryption operations in blok cipher algorithm

PDF created with pdfFactory Pro trial version www.pdffactory.com

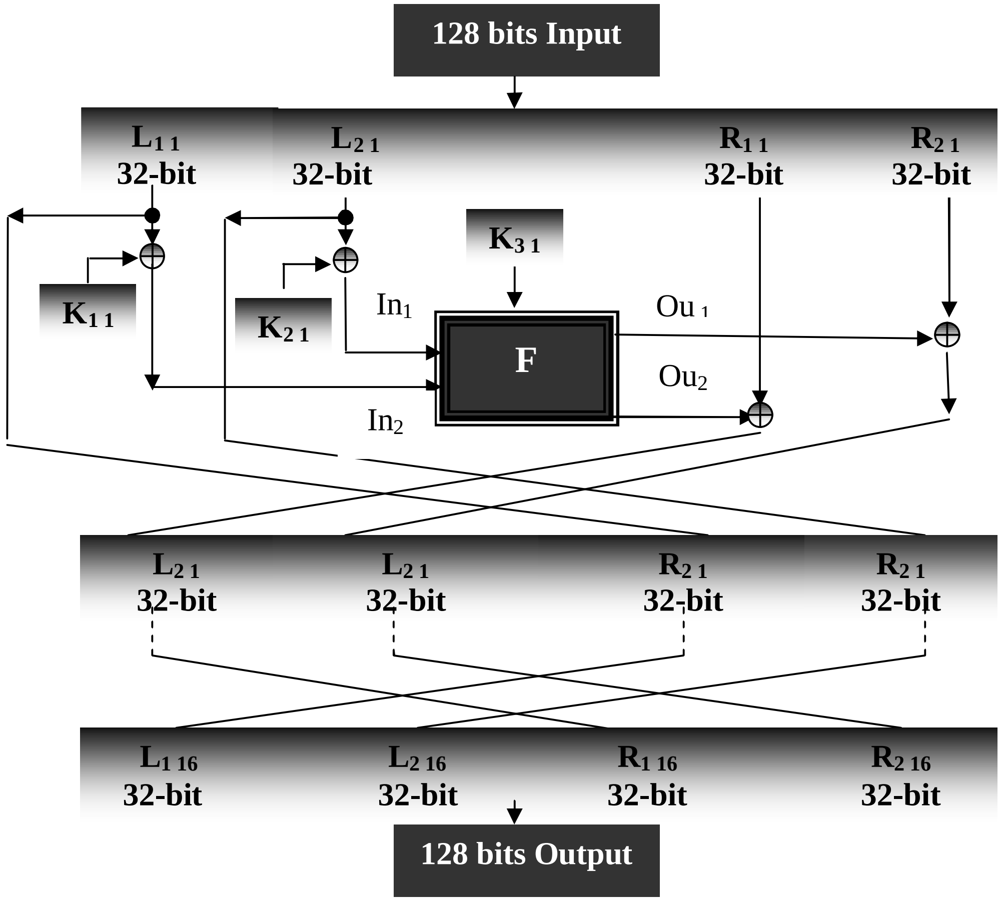
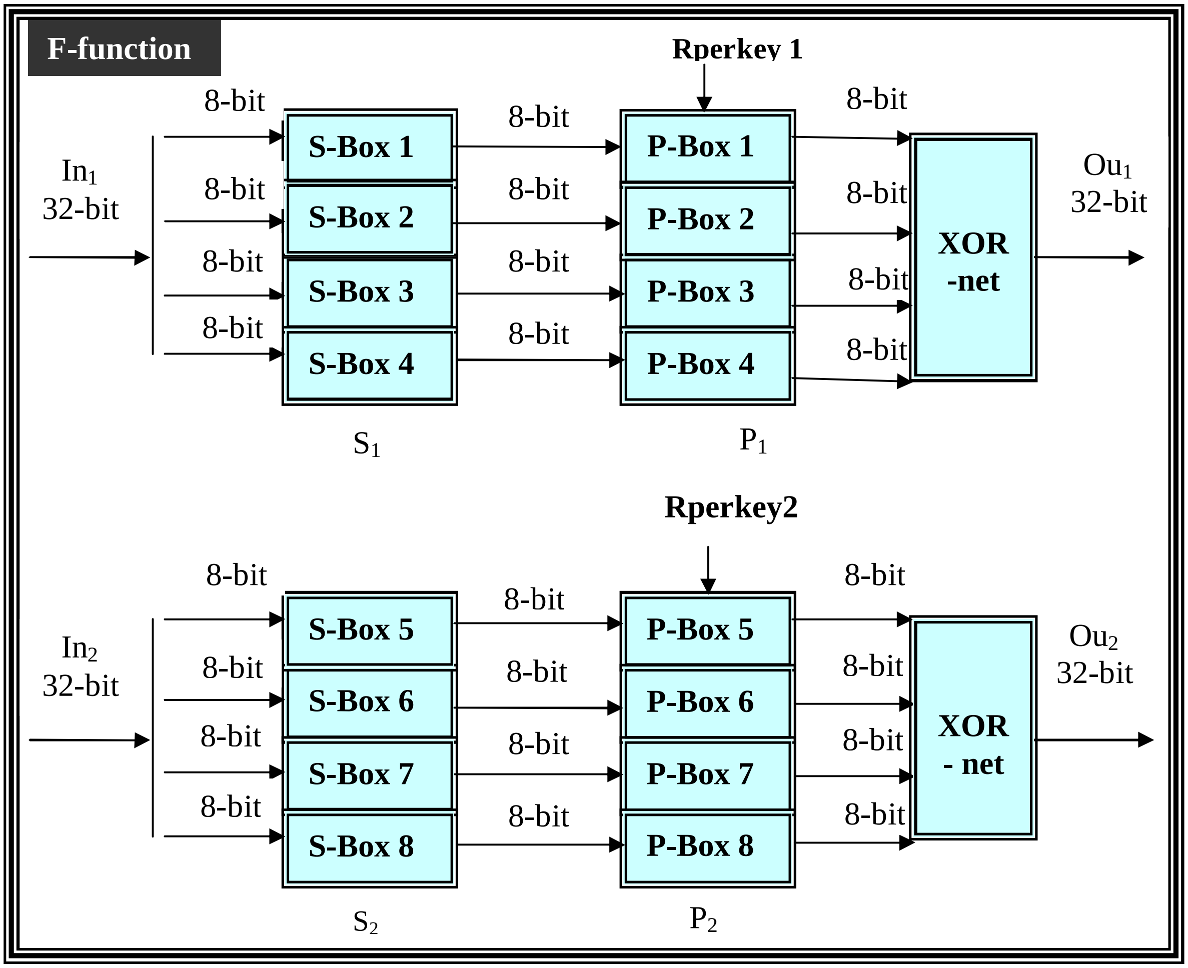


Figure (3) MAK-128 algorithm structur

PDF created with pdfFactory Pro trial version www.pdffactory.com



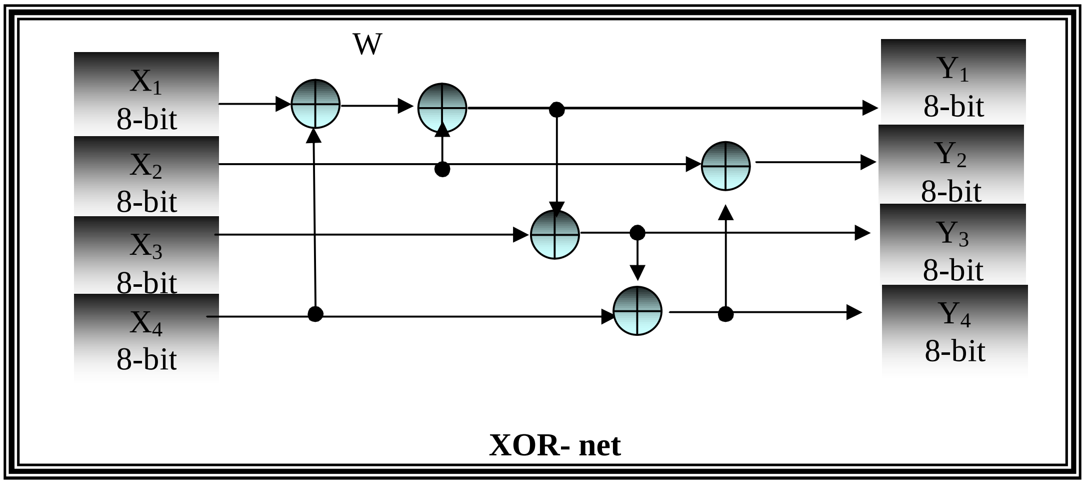


Figure (5) XOR-net structure of MAK-128 algorithm

PDF created with pdfFactory Pro trial version www.pdffactory.com