International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.1, January 2013

A NOVEL STRUCTURE WITH DYNAMIC OPERATION MODE FOR SYMMETRIC-KEY BLOCK CIPHERS

Kuo-Tsang Huang[1] , Jung-Hui Chiu[1] and Sung-Shiou Shen[2]

[1]Department of Electrical Engineering, Chang Gung University, Tao-Yuan, Taiwan
d9221006@gmail.com, jhchiu@mail.cgu.edu.tw
[2]DE LIN Institute of Technology, New Taipei City, Taiwan
shen@dlit.edu.tw

ABSTRACT

Modern Internet protocols support several modes of operation in encryption tasks for data confidentiality to keep up with varied environments and provide the various choices, such as multi-mode IPSec support. To begin with we will provide a brief background on the modes of operation for symmetric-key block ciphers. Different block cipher modes of operation have distinct characteristics. For example, the cipher block chaining (CBC) mode is suitable for operating environments that require self-synchronizing capabilities, and the output feedback (OFB) mode requires encryption modules only. When using symmetric-key block cipher algorithms such as the Advanced Encryption Standard (AES), users performing information encryption often encounter difficulties selecting a suitable mode of operation. This paper describes a structure for analyzing the block operation mode combination. This unified operation structure (UOS) combines existing common and popular block modes of operation. UOS does multi-mode of operation with most existing popular symmetric-key block ciphers and do not only consist of encryption mode such as electronic codebook (ECB) mode, cipher block chaining (CBC) mode, cipher feedback (CFB) mode and output feedback (OFB) mode, that provides confidentiality but also message authentication mode such as the cipher block chaining message authentication code (CBC-MAC) in cryptography. In Cloud Computing, information exchange frequently via the Internet and on-demand. This research provides an overview and information useful for approaching low-resource hardware implementation, which is proper to ubiquitous computing devices such as a sensor mote or an RFID tag. The use of the method is discussed and an example is given. This provides a common solution for multi-mode and this is very suitable for ubiquitous computing with several resources and environments. This study indicates a more effectively organized structure for symmetric-key block ciphers to improve their application scenarios. We can get that it is flexible in modern communication applications.

KEYWORDS

Block Cipher, Mode of Operation, Ubiquitous, Low-Resource

1. INTRODUCTION

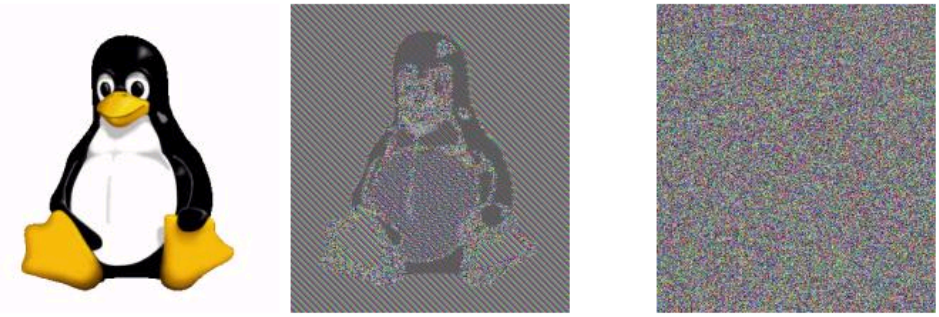
Data confidentiality is one of the security services in cryptography. The major concept in information security today is to continue to improve encryption algorithms. There are two major types of encryption algorithms for cryptography, symmetric-key algorithms and public-key algorithms. Symmetric-key algorithms also referred to as conventional encryption algorithms or single-key encryption algorithms are a class of algorithms that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext. It remains by far the most widely used of the two types of encryption algorithms. Symmetric-key encryption algorithms can use either stream ciphers or block ciphers. Block ciphers take a number of bits and encrypt them as a single unit, padding the plaintext so that it is a multiple of the block size. Blocks of 64 bits have been commonly used. The Advanced Encryption Standard (AES) [1] algorithm approved

DOI : 10.5121/ijnsa.2013.5102

by NIST in December 2001 uses 128-bit blocks. Many modern block ciphers have invertible functions from other functions that are themselves not invertible.

Unfortunately, the non-feedback conventional block ciphers have plaintext-ciphertext pair problem with the disadvantage of limit block region scramble. A disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks. non-feedback Conventional block ciphers do not hide data patterns well. A crack can use it to do known-plaintext attack. In some senses, it doesn't provide serious message confidentiality. A non-feedback conventional block cipher is not recommended for use in cryptographic protocols at all. A striking example of the degree to which non-feedback electronic codebook (ECB) mode can leave plaintext data patterns in the ciphertext can be seen when the electronic codebook mode is used to encrypt a bitmap image which uses large areas of uniform colour. The overall image may still be discerned as the pattern of identically-coloured pixels in the original remains in the encrypted version while the colour of each individual pixel is encrypted. Block cipher modes of encryption beside the electronic codebook mode have been suggested to remedy these drawbacks.

In the following images, a pixel-map version of the image on the left was encrypted with electronic codebook (ECB) mode to create the centre image, versus a non-ECB mode for the right image. The image on the right is how the image might appear encrypted with any of the other more secure modes indistinguishable from random noise. In Figure 1, the random appearance of the image on the right does not ensure that the image has been securely encrypted; many kinds of insecure encryption have been developed which would produce output just as random-looking.



Original Encrypted using ECB mode Modes other than ECB result in pseudo-randomness

Figure 1. A striking example when different modes are used to encrypt a bitmap image [2]

How to make the same plaintext input to transform the different ciphertext output? We can easy to show that the input tuple {M, K, A} and {C, K, A}. If we fix the {M, K}, we just can change the {A}. But the {A} is fixed standard, so we change the parameter of {A}. We change the internal {M} by exclusive-OR any know information. The known information is exposed when decryption or on channel in the transmission interval with time variant character. We suggest every cipher ought to support bit-stream character to output ciphertext.

The standard modes of operation described in the literature [7], such as non-feedback electronic codebook (ECB) mode, cipher block chaining (CBC) mode, output feedback (OFB) mode, and cipher feedback (CFB) mode provide confidentiality. How to choose an appropriate operation mode? The different mode has the different characters. For example, both of CFB and OFB can be design operating without padding with bit-based size keystream output; both of CBC and CFB can self sync to avoid channel noise error propagation; and both of CFB and OFB encryption and decryption applications need an encryption module only to reach both usages. In

addition, only the forward cipher function of the block cipher algorithm is used in both encryption and decryption operations, without the need for the inverse cipher function. Modern protocols support several operation modes and ciphers for suitable various operated environments. For example SSL and IPSec support multi-cipher and multi-mode [3][4][5][6]. However, each encryption operates with one fixed mode during a task session. Although a lot of effort is being spent on improving the plaintext-ciphertext pair problem, the efficient method has yet to be developed. The rest of this paper is organized as follows. Chapter 2 contains the related works of symmetric block ciphers. Chapter 3 contains a design of the unified operation structure; and chapter 4 describes partial operations with standard. Chapter 5 describes a operating simulation with mode selections; and finally section 6 is the conclusion.

2. RELATED WORKS

By far the most important automated tool for network and communications security is encryption [7]. Symmetric-key ciphers are a class of ciphers for cryptography that use trivially related cryptographic secret keys for both encryption of plaintext and decryption of ciphertext. The secret encryption key is trivially related to the secret decryption key, in that they may be identical or there is a simple transformation to go between the two secret keys. In practice, the secret keys represent a shared secret between two or more parties that can be used to maintain a private information link. Symmetric-key cryptography is to be contrasted with asymmetric-key cryptography, and symmetric-key cryptography was the only type of encryption in use prior to the development of asymmetric-key cryptography.

Following the discussion of the symmetric-key cryptography, modes of operation is the technique of making the repeated and secure use of a block cipher under a secret key. A block cipher allows encryption only of one single data block of the cipher's block length. When targeting a variable-length message, the data must be partitioned into separate cipher blocks. Typically, the last block must be extended to match the cipher's block length using a suitable padding scheme. A mode of operation describes the technique of encrypting each of these data blocks, and generally uses randomization based on an additional input value, often called an initialization vector (IV) , to allow doing so safely [8][9].

An initialization vector is a block of bits or a binary sequence. This binary sequence is used by several modes to randomize the encryption and hence to produce distinct ciphertexts without the need for a slower re-keying process. Even if the same plaintext is encrypted multiple times, it can produce distinct ciphertexts. An initialization vector has different security requirements than a key, so the initialization vector usually does not need to be secret and be transmitted on a channel in public. However, it is important that an initialization vector is never reused under the same key. For cipher block chaining (CBC) mode and cipher feedback (CFB) mode, reusing an IV leaks some information about the first block of plaintext. For output feedback (OFB) mode, reusing an IV completely destroys security.

Although a block cipher works on units of a fixed block size. Messages come in a variety of lengths. Therefore some modes of operation require that the final block be padded before encryption. Cipher block chaining (CBC) mode is such an operation mode with padding. Several padding schemes exist. The simplest is to add null bytes to the plaintext to bring its length up to a multiple of the block size, but care must be taken that the original length of the plaintext can be recovered. Cipher feedback (CFB) and output feedback (OFB) modes do not require any special measures to handle messages whose lengths are not multiples of the block size.

Block encryption may be vulnerable to ciphertext searching, replay, insertion and deletion because it encrypts each block independently. Several well-known techniques have been suggested to remedy these drawbacks. They are self-synchronizing stream cipher in cipher block chaining (CBC) mode and in cipher feedback (CFB) mode. However, self-synchronizing

stream ciphers, such as cipher block chaining (CBC) mode and cipher feedback (CFB) mode, exist one disadvantage with these techniques when they are applied to disk encryption: any modification in encrypted files requires the re-encryption of all succeeding plaintexts after the place of modification. Therefore, block cipher encryption modes of operation have been studied extensively in regard to their error propagation properties under various scenarios of data modification and so on. The earliest modes of operation, electronic codebook (ECB) mode, cipher block chaining (CBC) mode, output feedback (OFB) mode, and cipher feedback (CFB) mode, date back to 1981 and were specified in FIPS 81, DES Modes of Operation [2].

There are two cascade types for mode of operation. In the recent research called multi-mode, one type multi-mode cascades operations with the same data block [10], and another type cascades operations in the same session [11]. The related works about multi-mode and multi-cipher, we explore Crypto-coprocessors and multi-cipher cryptosystem [12][13][14][15]. NOP-cycle-padding algorithm (NCPA) [15] is one of those re-configurable cryptosystems used for hardware acceleration. In other words, [15] enables crypto-coprocessors reconfigured with diverse encrypting bursts to be pipeline scheduled. Crypto-coprocessor like CryptoManiac (CM) [12] processor is a flexible crypto-coprocessor. CryptoManiac supports multiple cipher algorithms and multi-mode operations. [11] introduces a multi-cipher cryptosystem (MCC) which enables a cryptosystem to use multiple cipher algorithms concurrently of fixed sequence in a session of data communication. The implementation of a sample MCC is introduced in [11] using Field Programmable Gate Array. Here refers to a multi-mode of operation, the kind of the internal structure of the cryptographic module design considerations for a specific circuit. One of the multi-mode operations Field Programmable Gate Array (FPGA) implementation [14] is a fast pipelined DES architecture operating in IP representation.

3. THE UNIFIED OPERATION STRUCTURE

A symmetric-key block cipher by itself allows operation only of a single data block of the cipher's block length. For a variable-length input message, the operating data must first be partitioned into separate cipher blocks. Known as the electronic codebook (ECB) mode is in the simplest case, and each block is encrypted and decrypted independently.

The standard modes of operation described in the literature [8], such as non-feedback electronic codebook (ECB) mode, cipher block chaining (CBC) mode, output feedback (OFB) mode, and cipher feedback (CFB) mode provide confidentiality. Many modern block ciphers have invertible functions from other functions that are themselves not invertible. Some symmetric ciphers use the same function to perform encryption and decryption applications, some does not. Because of some modes decrypt message using the same algorithm as encrypting, i.e. CFB and OFB. We define three levels about **module**, **function** and **application** to distinguish different usage situations. A crypto cipher **module** of any one symmetric cipher can perform both the encryption and decryption **applications**. This module has two **functions**, an encryption function and a decryption function. In some ciphers the two functions are the same as one, some are not. **E-application** is a practical application in message encryption, and **D-application** is a practical application in message decryption. **E-function**, that we denote $E_K[\ ]$, is a mapping function of transforming information (referred to as plaintext) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. **D-function**, that we denote $D_K[\ ]$, is the inverse one of E-function.

In the popular cipher block chaining (CBC) mode, for encryption to be secure the initialization vector passed along with the plaintext message must be a random or pseudo-random value, which is added in an exclusive-or manner to the first plaintext block before it is being encrypted. The resultant ciphertext block is then used as the new initialization vector for the next plaintext block. In the cipher feedback (CFB) mode, which emulates a self-synchronizing stream cipher, the initialization vector is first encrypted and then added to the plaintext block. The output

feedback (OFB) mode repeatedly encrypts the initialization vector to create a key stream for the emulation of a synchronous stream cipher.

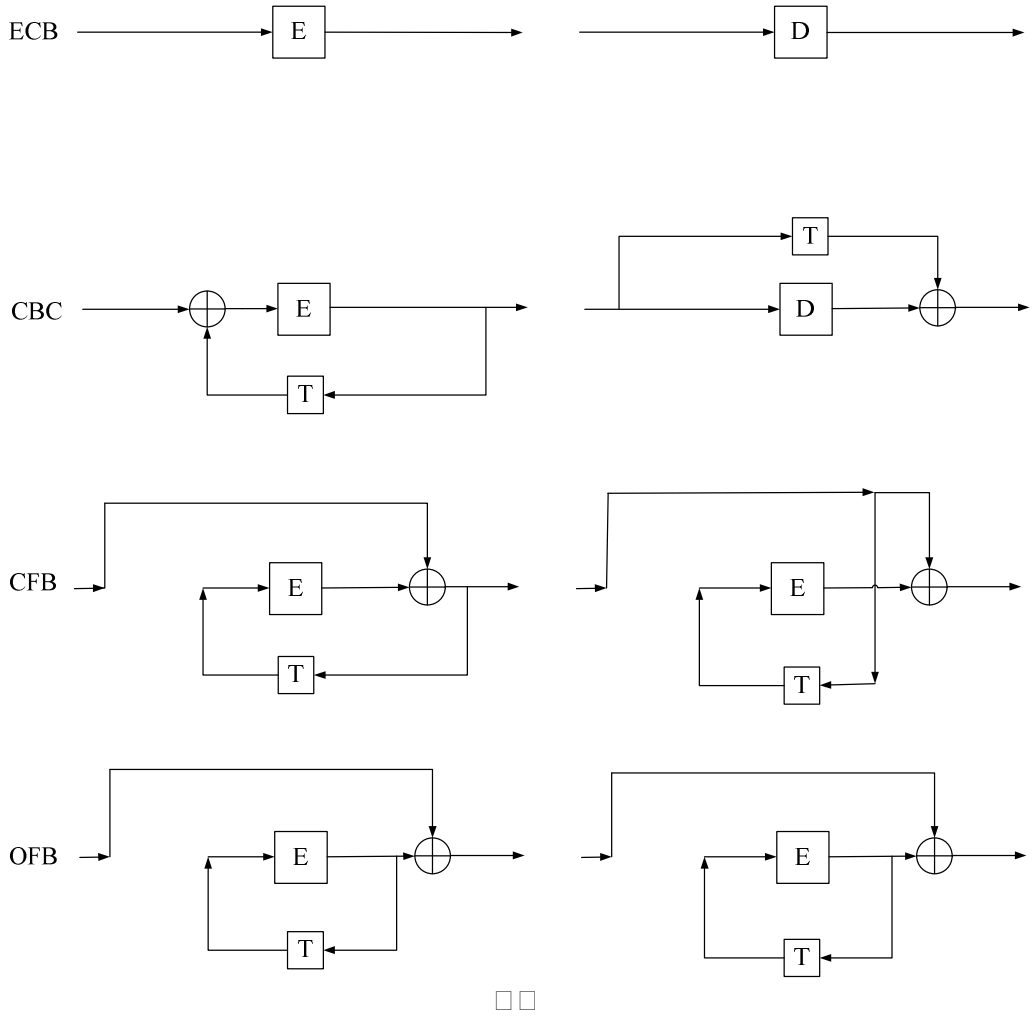


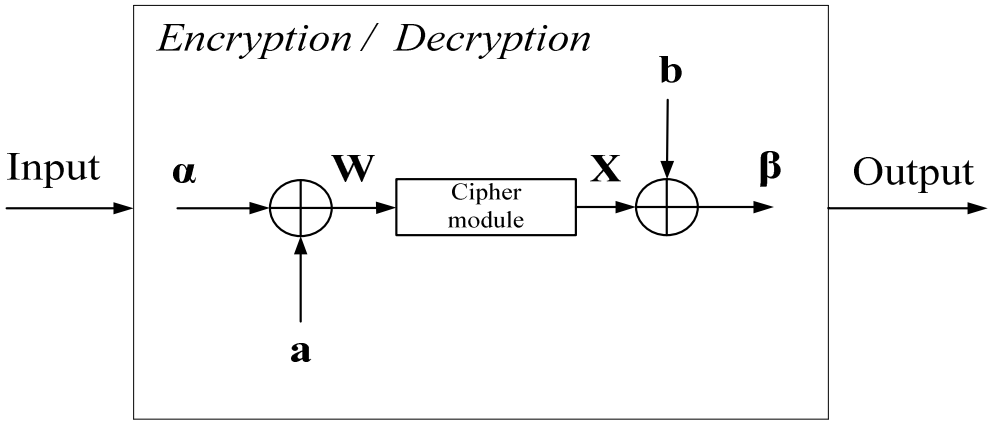
Figure 2. Input data path flow diagrams of ECB, CBC, CFB and OFB

If users want to combine one more mode to archive multi-mode performance, they must prepare several feedbacks to suit any one of the target mode needed. This is because people must hold several data for a proper feedback usage. For example, a user must prepare the current ciphertext data block feedbacks in operating the CBC mode.

How about solving the feedback problem? We use four buffers to hold the previous period, i.e. block cycle, parameters: previous input α of UOS, previous output β of UOS, previous inter-input W of cipher, previous inter-output X of a cipher. Especially in encryption, the previous output of UOS can be retrieved from ciphertext sequence, so that we do not buffer it. Therefore all the four buffers can reduce to three buffers. Because of the three buffers, we solved the feedback problem.

The W is inter-input of cipher module and the X is an inter-output of cipher module. The previous output of UOS is ciphertext in encryption used for CBC and CFB. The previous inter-output of a cipher is au output feedback used for OFB.

Suppose we want to encrypt a long input message sequence. First, we divide the sequence into several blocks, Pi, i=1,2..., each Pi with 128 bit length. Then, the encryption operations and the decryption operations are performed in following figures. Where Xi is the output from cipher module; and each Xi, for i=1,2..., is a 128 bit sequence. The four modes of operation (ECB, CBC, CFB and OFB) are presented in chapter 4.



$$\beta = \mathrm{f}(\alpha, \mathrm{a}, \mathrm{b}) = \mathrm{E}_K(\alpha \oplus \mathrm{a}) \oplus \mathrm{b}$$

When ECB/CBC/CFB/OFB encryption
CFB/OFB decryption
CBC-MAC

$$\beta = \mathrm{f}^{-1}(\alpha, \mathrm{a}, \mathrm{b}) = \mathrm{D}_K(\alpha \oplus \mathrm{a}) \oplus \mathrm{b}$$

When ECB/CBC decryption

holding the previous period information parameters:

α previous input of UOS

β previous output of UOS

W previous inter-iuput of cipher

X previous inter-output of cipher

□□Figure 3. The proposed novel multi-mode structure

Figure 3. The proposed novel multi-mode structure

Table 1. The Parameters for Proposed UOS

Operation $Mode_{App}$	Cipher module	**α**	**β**	**a**	**b**	Schematic diagram
ECB_e	$Cipher_e$	P_i	C_i	0	0	Fig 4.(a)
ECB_d	$Cipher_d$	C_i	P_i	0	0	Fig 4.(b)
CBC_e	$Cipher_e$	P_i	C_i	C_{i-1}	0	Fig 5.(a)
CBC_d	$Cipher_d$	C_i	P_i	0	C_{i-1}	Fig 5.(b)
CFB_e	$Cipher_e$	0	C_i	C_{i-1}	P_i	Fig 6.(a)
CFB_d	$Cipher_e$	0	P_i	C_{i-1}	C_i	Fig 6.(b)
OFB_e	$Cipher_e$	0	C_i	X_{i-1}	P_i	Fig 7.(a)
OFB_d	$Cipher_e$	0	P_i	X_{i-1} / W_{i-1}*	C_i	Fig 7.(b)
CBC-MAC	$Cipher_e$	P_n	*tag*	C_{n-1}	0	Fig 8.

P.S. The mask means ciphers with *decryption function* especially

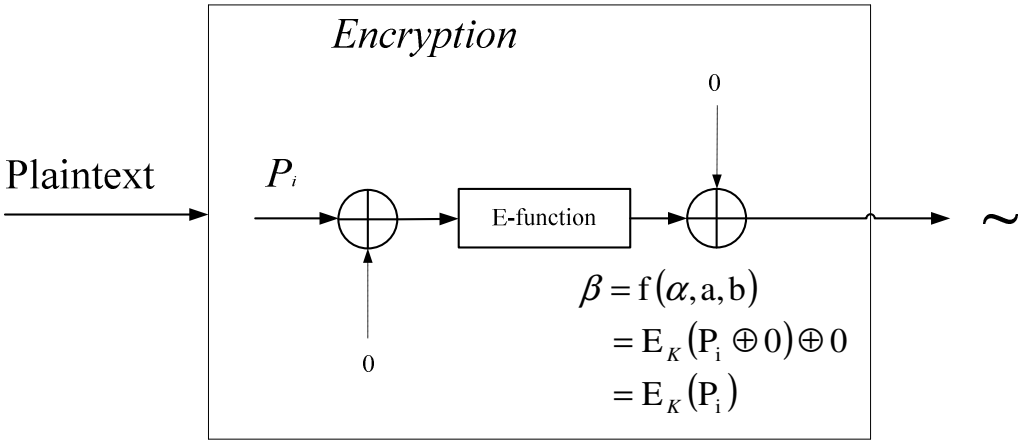
4. PARTIAL OPERATIONS WITH STANDARD

Several so-called block cipher modes of operation have been designed and specified in national recommendations such as NIST 800-38A and international standards such as ISO/IEC 10116 [8][16].

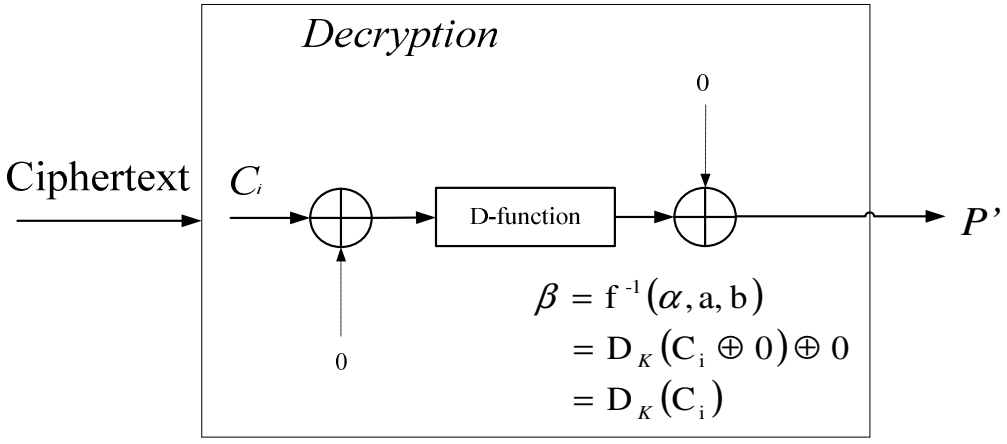
4.1. Electronic CodeBook (ECB) mode

The simplest of the encryption modes is the electronic codebook mode. The message is divided into blocks and each block is encrypted separately. The disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks; thus, it does not hide data patterns well. In some senses, it doesn't provide serious message confidentiality, and it is not recommended for use in cryptographic protocols at all.

ECB mode can also make protocols without integrity protection even more susceptible to replay attacks, since each block gets decrypted in exactly the same way. For example, the Phantasy Star Online: Blue Burst online video game uses Blowfish in ECB mode. Before the key exchange system was cracked leading to even easier methods, cheaters repeated encrypted "monster killed" message packets, each an encrypted Blowfish block, to illegitimately gain experience points quickly.



(a) Encryption application



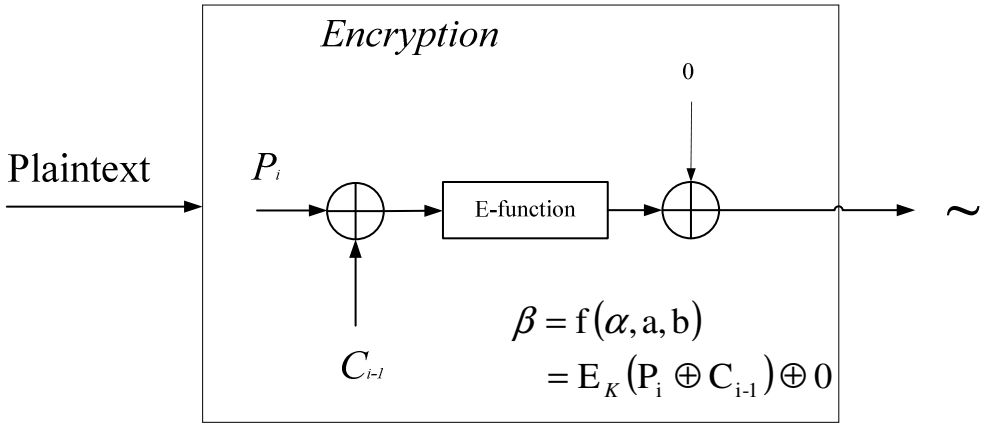
(b) Decryption application

Figure 4. Electronic CodeBook mode

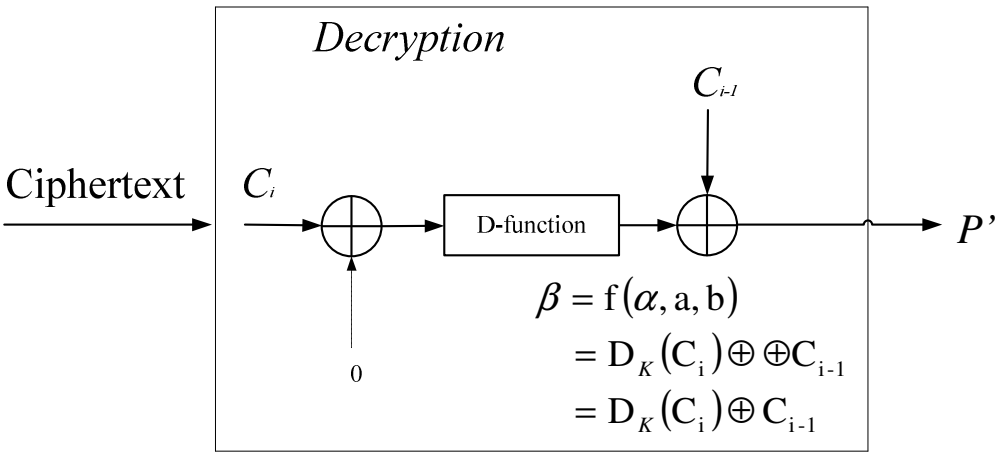
4.2. Cipher Block Chaining (CBC) mode

Cipher block chaining is a block cipher mode that provides confidentiality but not message integrity in cryptography. Cipher block chaining mode of operation was invented by IBM in 1976 [17]. In CBC mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block is dependent on all plaintext blocks processed up to that point. Also, to make each message unique, an initialization vector must be used in the first block.

CBC has been the most commonly used mode of operation. Its main drawbacks are that encryption is sequential (i.e., it cannot be parallelized), and that the message must be padded to a multiple of the cipher block size. One way to handle this last issue is through the method known as ciphertext stealing. Note that a one-bit change in a plaintext affects all following ciphertext blocks.



(a) Encryption application



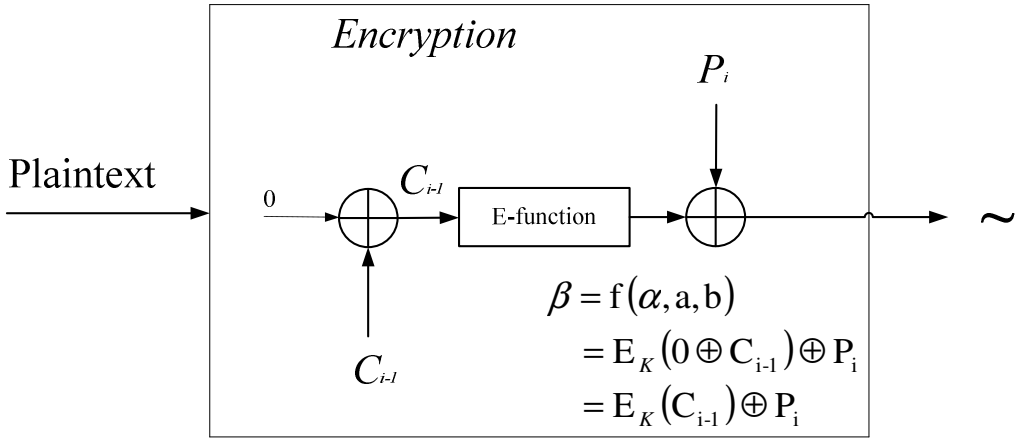
(b) Decryption application

Figure 5. Cipher Block Chaining mode

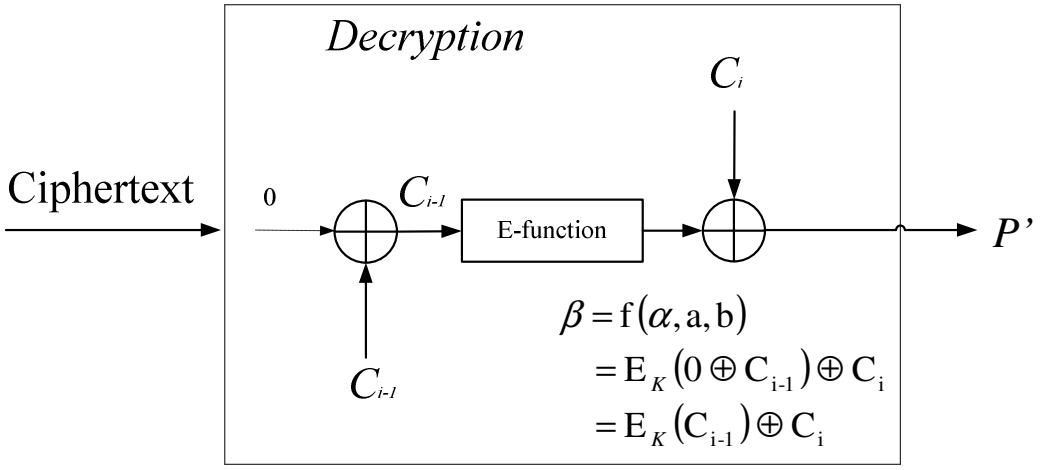
Decrypting with the incorrect previous ciphertext block causes the current block of plaintext to be corrupt but subsequent plaintext blocks will be correct. This is because a plaintext block can be recovered from two adjacent blocks of ciphertext. As a consequence, decryption can be parallelized. Note that a one-bit change to the ciphertext causes complete corruption of the corresponding block of plaintext, and inverts the corresponding bit in the following block of plaintext, but the rest of the blocks remain intact.

4.3. Cipher FeedBack (CFB) mode

The cipher feedback mode is a confidentiality mode that features the feedback of successive ciphertext segments into the input blocks of the forward cipher to generate output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa.



(a) Encryption application



(b) Decryption application

Figure 6. Cipher FeedBack mode

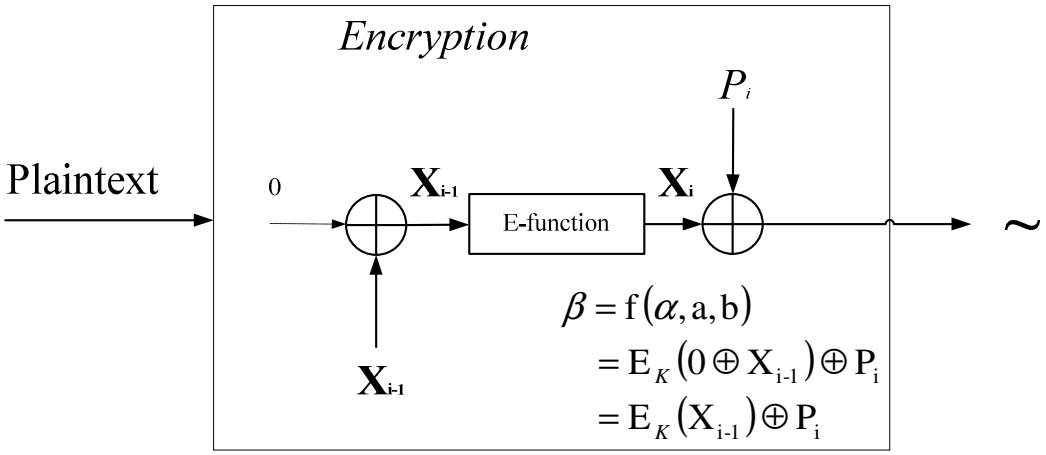
Here, a plaintext block is enciphered by being XORed to the encryption module of the previous ciphertext block. The process is repeated with the successive input blocks until a ciphertext segment is produced from every plaintext segment. In general, each successive input block is enciphered to produce an output block.

In CFB encryption, like CBC encryption, the input block to each forward cipher function depends on the result of the previous forward cipher function; therefore, multiple forward cipher operations cannot be performed in parallel. In CFB decryption, the required forward cipher operations can be performed in parallel if the input blocks are first constructed (in series) from the ciphertext [18].

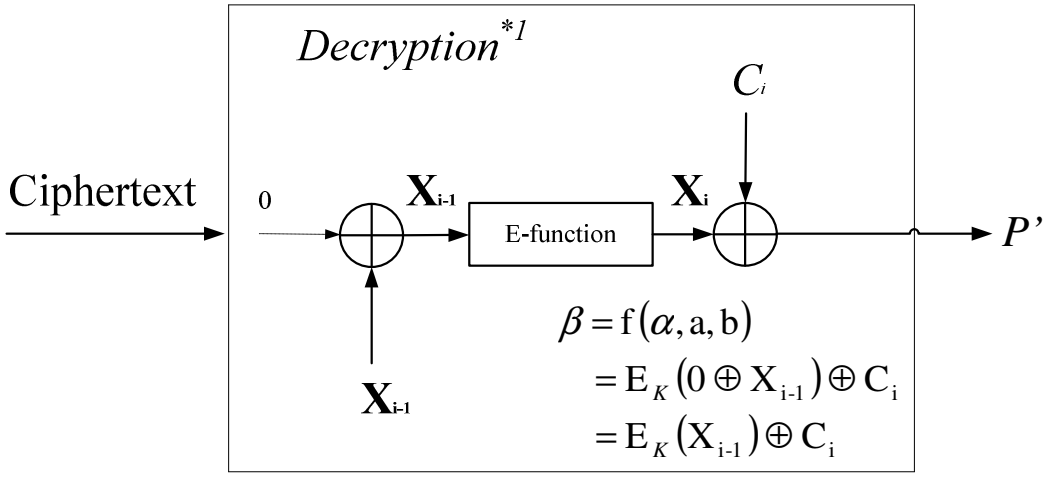
4.4. Output Feedback (OFB) Mode

One other mode among those originally suggested for use was Output Feedback Mode: this mode encrypted an initial value with DES, and then the result of the encryption was encrypted again repeatedly. The resulting values were used as a keystream to XOR with messages.

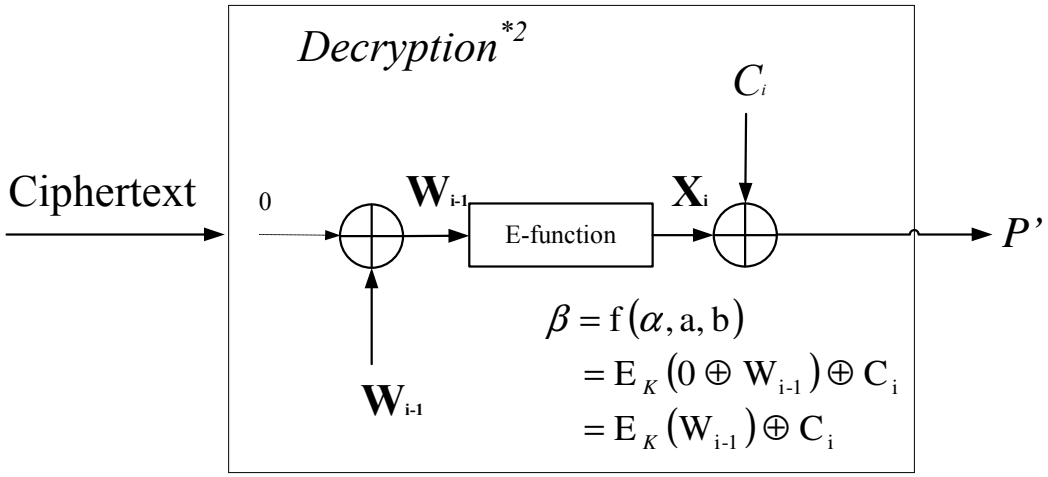
The Output Feedback (OFB) mode is a confidentiality mode that features the iteration of the forward cipher on an IV to generate a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The OFB mode requires that the IV is a nonce, i.e., the IV must be unique for each execution of the mode under the given key; In OFB encryption, the IV is transformed by the forward cipher function to produce the first output block. The first output block is exclusive-ORed with the first plaintext block to produce the first ciphertext block. The forward cipher function is then invoked on the first output block to produce the second output block. The second output block is exclusive-ORed with the second plaintext block to produce the second ciphertext block, and the forward cipher function is invoked on the second output block to produce the third output block. Thus, the successive output blocks are produced by applying the forward cipher function to the previous output blocks, and the output blocks are exclusive-ORed with the corresponding plaintext blocks to produce the ciphertext blocks.



(a) Encryption application



*1: When previous mode change is OFB or CFB



*2: When previous mode change is ECB or CBC

(b) Decryption application

Figure 7. Output FeedBack mode

In OFB decryption, the IV is transformed by the forward cipher function to produce the first output block. The first output block is exclusive-ORed with the first ciphertext block to recover the first plaintext block. The first output block is then transformed by the forward cipher function to produce the second output block. The second output block is exclusive-ORed with the second ciphertext block to produce the second plaintext block, and the second output block is also transformed by the forward cipher function to produce the third output block. Thus, the successive output blocks are produced by applying the forward cipher function to the previous output blocks, and the output blocks are exclusive-ORed with the corresponding ciphertext blocks to recover the plaintext blocks.

4.5. Cipher Block Chaining Message Authentication Code (CBC-MAC)

A cipher block chaining message authentication code (CBC-MAC) is a technique for constructing a message authentication code from a block cipher. The message is encrypted with some block cipher algorithm in CBC mode to create a chain of blocks such that each block depends on the proper encryption of the previous block. This interdependence ensures that a change to any of the plaintext bits will cause the final encrypted block to change in a way that cannot be predicted or counteracted without knowing the key to the block cipher.

To calculate the CBC-MAC of message one encrypts in CBC mode with zero initialization vector. The following figure sketches the computation of the CBC-MAC of a message comprising $P_1||P_2||P_3||..||P_n$ using a secret key K and a block cipher E():

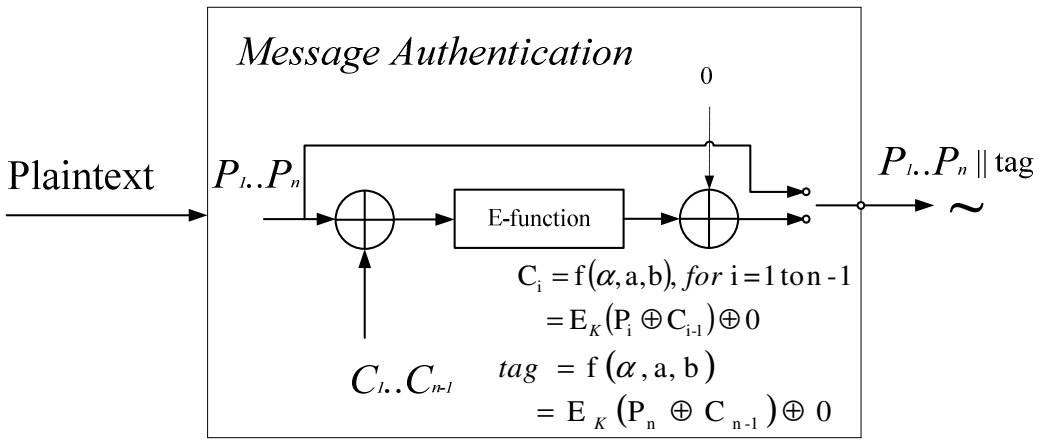


Figure 8. Cipher Block Chaining Message Authentication Code

The simple CBC-MAC operation uses CBC encryption, just CBC-MAC outputs of UOS are the through passed plaintext block from the first divided message block to the end. A tag only goes behind the whole message with C_n as the message authentication code, i.e. an integrity check value.

5. Mode Selection and Operating Simulation

The mode selection could be built into the unified operation structure to make it more powerful. We design three generators for mode selection. The first one, easy generator, performs an operational simulation as follows. To change the operation mode, a mode selection bit sequence is proposed for being used to change the operation mode. Assumed that the number of modes of operation is 4 by ECB, CBC, CFB and OFB. The mode selection parameter may be generated by a mode selection generator.

5.1. A Simple Case with 4-bit Test-Cipher

Here we use a 4-bit Test-Cipher operation to show the example. Here the simple case uses the 2's complement as the sample 4-bit Test-Cipher encryption with special target key. The 4-bit Test-Cipher operating of encryption with special target key from input to output is shown in following table. It is the same as the ECB mode operation, and the operating of decryption is the backward transformation.

Table 2. The mapping table of a 4-bit Test-Cipher

input of encryption	output of encryption	input of decryption	output of decryption
0000	0000	0000	0000
0001	1111	1111	0001
0010	1110	1110	0010
0011	0011	0011	0011
0100	1001	1001	0100
0101	0010	0010	0101
0110	1000	1000	0110
0111	0101	0101	0111
1000	0110	0110	1000
1001	0111	0111	1001

We define that current mode exchange is depend on the middle two bits of last plaintext. For example, if the middle two bits of last plaintext are 01_2, then we choose the current mode exchange to CBC mode. Therefore 00_2 means ECB, 01_2 means CBC, 10_2 means CFB and 11_2 means OFB choice. When the first block of plaintext is coming, we define the ECB as the default operation mode to handle it.

There is an example: a 40-bit plaintext is 0000000100100011010001010110011110001001. Every 4-bit divide to one block. The sample block size is 4-bit because of easy trace. If we prepare to encrypt it by a 4-bit Test-Cipher, the plaintext can be shown as 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001. Including showing the mode choice, the plaintext can be shown as 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001. We can see that the example of mode from 00_2 to 00_2 then 01_2 then 01_2 then 10_2 then 10_2 then 11_2 then 11_2 then 00_2, and finally with 00_2. The first block of plaintext is operating with the default operation mode ECB, and the after mode sequence is from ECB to ECB then CBC then CBC then CFB then CFB then OFB then OFB then ECB, and the last with ECB.

Step1: a 40-bit plaintext

0000000100100011010001010110011110001001

Step2: every 4-bit divide to one block

0000 0001 0010 0011 0100 0101 0110 0111 1000 1001

Step3: including showing the transform choice

3-1. 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001

3-2. 00_2 → 00_2 → 01_2 → 01_2 → 10_2 → 10_2 → 11_2 → 11_2 → 00_2 → 00_2

3-3. ECB → ECB → CBC → CBC → CFB → CFB → OFB → OFB → ECB → ECB

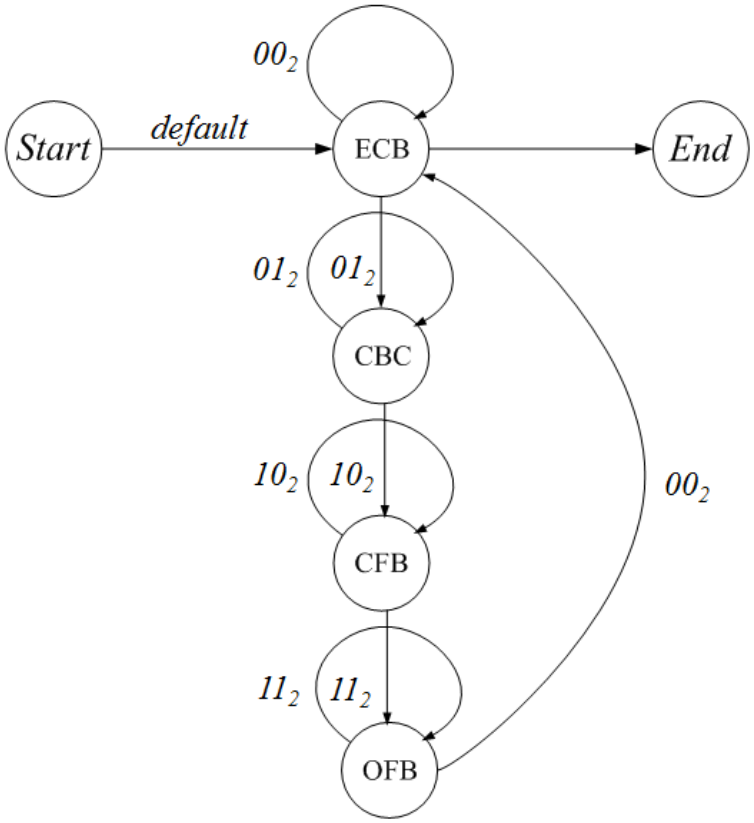


Figure 9. Example of ECB → CBC→CFB→OFB→ECB

Figure 9 is an example of mode transform from ECB to CBC then CFB then OFB, and back to ECB. We use Finite State Machine (FSM) to explain the state between block operations.

The above example demos the eight types of transforms. All the sixteen types of mode changing are shown in following table.

Table 3. All types of mode changing with 2-bit of mode choices

	previous		next		previous		next
type 1	ECB	→	ECB	type 9	CFB	→	ECB
type 2	ECB	→	CBC	type 10	CFB	→	CBC
type 3	ECB	→	CFB	type 11	CFB	→	CFB
type 4	ECB	→	OFB	type 12	CFB	→	OFB
Type 5	CBC	→	ECB	type 13	OFB	→	ECB
Type 6	CBC	→	CBC	type 14	OFB	→	CBC
type 7	CBC	→	CFB	type 15	OFB	→	CFB
type 8	CBC	→	OFB	type 16	OFB	→	OFB

5.2. Mode Selection Bits

We define that current mode exchange depends on the two choice bits of last plaintext. For example, if the choice bits are 01_2, then we choose the current mode exchange to CBC mode. Therefore 00_2 means ECB, 01_2 means CBC, 10_2 means CFB and 11_2 means OFB choice.

Each proposed change mechanism is using 2 bits choice related from the last block plaintext message before current block operating. The mode change depends on 2-bit choice S, i.e. S_0S_1.

5.2.1. Easy Generator

Easy change is using 2 bits plaintext from the previous block plaintext message before current block operating. We define that current mode exchange depends on the msb./lsb./middle two bits of last plaintext. The mode change is depended on partial 2-bits message.

$$S=(S_0S1)=filter(P_{i-1})$$
$$=MSB^{2\text{-}bit}(P_{i-1})\ or\ LSB^{2\text{-}bit}(P_{i-1})\ or\ MID^{2\text{-}bit}(P_{i-1}) \quad (1)$$

5.2.2. Normal Generator

This generator uses two parity check bits, one is from all odd positions sequence and the other is from all even positions sequence. It can make a simple related effect. If changing any one bit then infecting effect the current block and behind operating.

$$S=(S_0S_1)$$
$$S_0=f^{odd}(P_{i-1})=Parity\ (P_{i-1}{}^{odd})$$
$$S_1=f^{even}(P_{i-1})=Parity\ (P_{i-1}{}^{even}) \quad (2)$$

5.2.3. Hash Generator

We improve the normal change by hash functions to instead of parity check functions. This brings hard scrambled performance but an extra cost of the resource.

$$S=(S_0S1)=f(Pi\text{-}1)=hash^{2\text{-}bit}(Pi\text{-}1)$$
$$=LSB^{2\text{-}bit}(\ MD5(Pi\text{-}1)\ )or\ LSB^{2\text{-}bit}(\ SHA\text{-}1(Pi\text{-}1)\ ) \quad (3)$$

5.3. Operating Simulation

According to the low-resource environment, we suggest using the easy generator for ubiquitous computing. Here we perform an operating simulation with the easy generator in the following table. This simulation is marking OFB and then CBC especially. The detail descriptions of one-by-one steps are in the appendix.

People can download the simulation programs with the link http://dl.dropbox.com/u/54967925/UOS_Win32.exe and http://dl.dropbox.com/u/54967925/UOS_x64.exe to verify the results. Those programs are suitable for OS: Windows 2000/XP Pro./Vista/7 but not Windows XP Home Edition.

Table 4. The Parameters for a Simulation

Cipher: AES
Key: 12121212121212123434343434343434
IVs: 00000000000000000000000000000000
Plaintext: 00000000000000000000000000000010 00000000000000000000000000000011 00000000000000000000000000000001 11111111111111111111111111111111 ……
Ciphertext: 1F8300022FAD7840E51D265C9A1B663F 8EC78F4182557DEE3461681A3061D901 644A48DCC8CB017482399212A5164471 4B095F7288862F4FD4D8F7BFDD18131B ……

6. CONCLUSIONS

Modern block cipher protocols support several modes of operation to provide the confidentiality for the requirements of different applications. To begin with we provide a brief background on the modes of operation for symmetric-key block ciphers. In this paper, a novel structure, called unified operation structure (UOS), is proposed. Our contribution of this paper can be summarized as follows; UOS uses possible options to satisfy any next one of all feedbacks so that the technique can perform several modes of operation. It is a common solution for multi-mode, easier to provide multiple modes of operation and suit for any kinds of block ciphers. This study proposes an integrated operational structure capable of providing dozens of standard and non-standard modes of operation. The use of the method is discussed and an example is given. We employed standard encryption as well as decryption modes in NIST SP800-38A as illustrative examples. We also design three generators for mode selection and then use the easy generator to perform an operating simulation according to the low-resource environment in cloud computing. It reduces the cost of resource for multi-mode implementation, supports continuous mode change application, and provides low-resource hardware implementation of a common solution for multi-mode. It is proper to ubiquitous computing devices such as a sensor mote with several resources and environments.

REFERENCES

[1] J. Daemen and V. Rijmen (2002). *The Design of Rijndael: AES - the advanced encryption standard.* Springer Verlag.

[2] Block cipher modes of operation http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

[3] IPSec Working Group. http://www.ietf.org/html.charters/ipseccharter.html

[4] OpenSSL. http://etutorials.org/Programming/secure+programming/Chapter+5.+Symmetric+Encryption/5.17+Performing+Block+Cipher+Setup+for+CBC+CFB+OFB+and+ECB+Modes+in+OpenSSL/

[5] SSL 3.0 Specification. http://wp.netscape.com/eng/ssl3/

[6] Lan Luo, ZhiGuang Qin, and Juan Wang, (2009) "The Intelligent Conversion for Different Layers' Block Ciphers," *ICIC Express Letters*, Vol. 3, No. 1, pp.73–77.

[7] William Stallings (2003). *Cryptography and Network Security: Principles and Practices 3rd Edition*. Pearson Education. ISBN 0-13-111502-2.

[8] National Institute of Standards and Technology (NIST), NIST. gov - Computer Security Division - Computer Security Resource Center, "Recommendation of block cipher security methods and Techniques," NIST SP800-38.

[9] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone (1996). *Handbook of Applied Cryptography*. CRC Press. ISBN 0-8493-8523-7.

[10] E. Biham, (1998) "Cryptanalysis of multiple modes of operation," *J. Cryptology*, Vol. 11, No. 1, pp. 45-58.

[11] Chung-Ping Young, Yen-Bor Lin & Chung-Chu Chia, (2009) "Software and Hardware Design of a Multi-cipher Cryptosystem," *Proc. IEEE TENCON 2009*, Singapore.

[12] Lisa Wu, Chris Weaver, and Todd Austin, (2001) "CryptoManiac: A Fast Flexible Architecture for Secure Communication," *Proc. IEEE Int. Symp. Comput. Archit.*, pp. 110–119.

[13] S. Laovs, A. Priftis, P. Kitsos, and O. Koufopavlou, (2003) "Reconfigurable crypto process design of encryption algorithms operation modes methods and FPGA integration," *Proc. IEEE Int. Conf. MWSCAS*, pp. 811–814.

[14] S. Guilley, P. Hoogvorst, and R. Pacalet, (2007) "A Fast Pipelined Multi-Mode DES Architecture Operating in IP Representation," *The VLSI Journal*, Vol. 40, No. 4, pp.479-489.

[15] Young, C.-P., (2008) "NCPA: A Scheduling Algorithm for Multi-cipher and Multi-mode Reconfigurable Cryptosystem," *Proc. IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing*), Harbin, China.

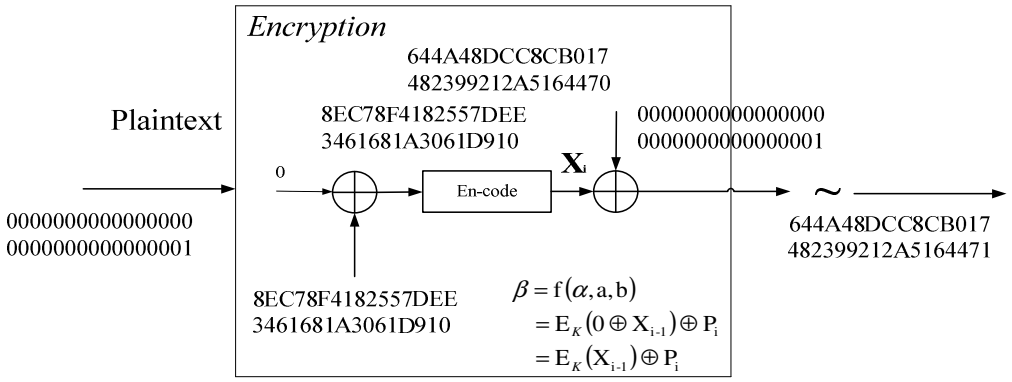
[16] International Organization for Standardization (ISO), "Information Technology-Security Techniques-Modes of Operation for. an n-bit Block Cipher," ISO/IEC 10116, 1997.

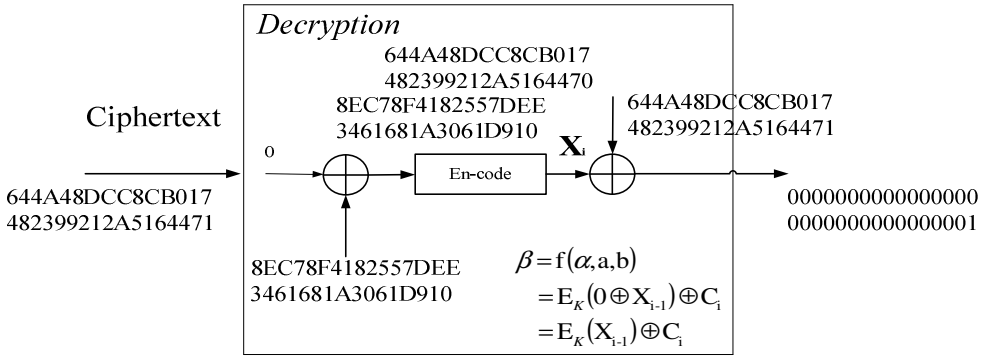
[17] William F. Ehrsam, Carl H. W. Meyer, John L. Smith, Walter L. Tuchman, (1976) "Message verification and transmission error detection by block chaining," US Patent 4074066.

[18] H. M. Heys, (2003) "Analysis of the Statistical Cipher Feedback Mode of Block Ciphers," *IEEE Transactions on Compuiters,* Vol. 52, No. 1.

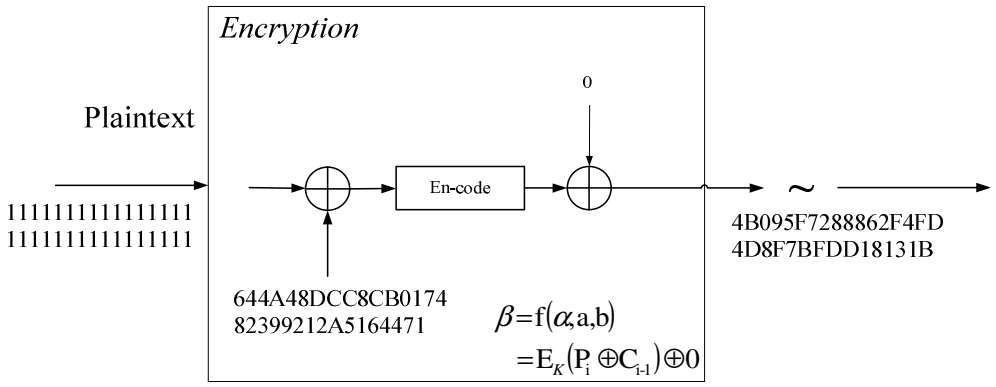
APPENDIX

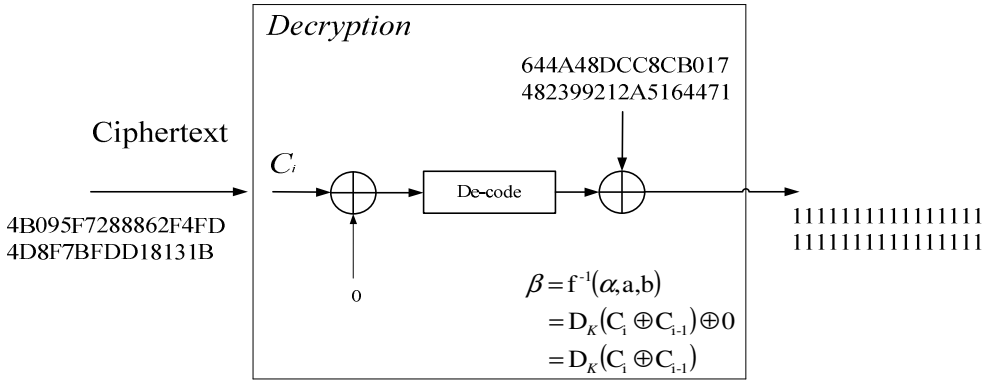
A. (1) OFB Encryption and Decryption Applications





A. (2) CBC Encryption and Decryption Applications





Authors

Mr. Kuo-Tsang Huang received B.Sc. from Chung Hua University in 2001 and M.Sc. from Aletheia University in 2003. He is currently studying for the Ph.D. degree in Department of Electrical Engineering of Chang Gung University, Taiwan. He is a member of the International Collaboration for Advancing Security Technology (iCAST). His research interests include wireless network, information security, cryptography, computer architecture issues and technology.

Dr. Jung-Hui Chiu received B.S.E.E. from Tatung University in 1971, M.S.E.E. in signal processing and Ph.D. in communication from National Taiwan University in 1973 and 1986 respectively. From 1975 to 1981, he was a research staff with Chunghwa Telecom Labs where he was involved in the research of fiber communications and the microwave systems. During 1981–1986, he was an institutor for the Electronic Department, National Taiwan University of Science and Technology, and was associate professor from 1986 to 2003. He is currently an associate professor in the Department of Electrical Engineering of Chang Gung University, Taiwan. He is a member of IEEE Communications Society, the Chinese Cryptology and Information Security Association (CCISA), and the International Collaboration for Advancing Security Technology (iCAST). His research interests include digital communication systems, wireless communication systems, information security, RFID, hardware security, smart card, and cryptography.

Dr. Sung-Shiou Shen received B.S.E.E. and M.S.E.E from National Taiwan University of Science and Technology in 1990 and in 1992, respectively. Since 2000, he has been a lecturer at the Department of Electronic Engineering, De Lin Institute of Technology in Taiwan. He is a member of IEEE Communications Society and the International Collaboration for Advancing Security Technology (iCAST). His research interests include digital communication systems, wireless communication systems, information security, hardware security, RFID, smart card, and cryptography.