JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 25, 937-944 (2009)

Short Paper

Structural Binary CBC Encryption Mode

YI-SHIUNG YEH, TING-YU HUANG AND HAN-YU LIN
Department of Computer Science
National Chiao Tung University
Hsinchu, 300 Taiwan

A block cipher is a kind of symmetric encryption algorithm that operates on blocks of fixed length, often 64 or 128 bits. It transforms blocks of plaintext into blocks of ciphertext of the same length under the provided secret key. A common characteristic of currently widely used modes of operation such as CBC, CFB and OFB is the sequential procedure, *i.e.*, the encryption/decryption algorithm can not start to process until the previous operation finished, which is considered to be inefficient in multi-processor structures. In this paper, we combine CBC mode of operation and the binary tree data structure to propose a new structural binary CBC encryption mode allowing parallelized computing. A significant property of the proposed mode of operation is independent branch operations. When applied in multi-processor structures, different branch operations can make effective use of CPUs to perform in parallel, which will lead to shorter computing time and greatly improve the overall performance.

Keywords: block cipher, encryption mode, data structure, binary tree, CBC

1. INTRODUCTION

Conventional symmetric cryptography [1, 2] can be categorized into block ciphers [3, 4] and stream ciphers [5, 6]. They differ in how large a size of the message is encrypted for each round. Block ciphers use a block size of 64 or 128 bits as one encryption unit and produce the corresponding ciphertext block of the same length. Data Encryption Standard (DES) [3] and Advanced Encryption Standard (AES) [4] are two widely used block cipher algorithms with a block size of 64 and 128 bits, respectively. Generally speaking, such block encryption procedures are more complicated and require additional memory space to store temporary outputted data. Stream ciphers such as the well-known RC4 [5] and the one-time pad (OTP) [6] instead encrypt the plaintext one bit at a time. In fact, stream ciphers can be regarded as a block cipher of a really small block size. The advantages of stream ciphers are easy to implement and fast to generate the ciphertext. However, owing to the low diffusion property, a malicious adversary has the possibility to break the confidentiality of the message by taking advantage of language frequency distribution techniques. In addition, the one-time pad requires the provided secret key to be as long as the message length, which is considered to be impractical regardless of its perfect security.

Received July 20, 2007; revised October 24, 2007; accepted November 22, 2007.
Communicated by Chin-Laung Lei.

In 1980, the National Institute of Standards and Technology (NIST) proposed four modes of operation for DES encryption in the FIPS PUB 81 [7, 8]. We briefly describe these four modes as follows:

(1) ECB (Electronic Code Book)

The plaintext is divided into 64-bit blocks and each block operation is independent of the others. That is, for each block, the plaintext encryption can perform solely without having to wait for the finish of its previous round. As a matter of fact, ECB is the easiest and the fastest mode to implement. However, its security is the weakest, since no additional mechanisms are incorporated besides the basic DES algorithm.

(2) CBC (Cipher Block Chaining)

In CBC mode of operation, each plaintext block is XORed with the ciphertext block of its previous round while the first plaintext block is XORed with the initialization vector (*IV*). The *IV* is public information and must be generated randomly for any particular execution of the encryption process. The XORed result is then applied to the encryption algorithm under the provided secret key to get the ciphertext block. It can be seen that all the blocks depend on all the previous blocks. By adopting the extra XOR step, this mode of operation is more secure than ECB.

(3) CFB (Cipher Feedback)

The CFB mode of operation is a close relative of CBC. The first plaintext block is XORed with the result of the encrypted initialization vector (*IV*). For each round, it encrypts the ciphertext block of its previous operation and then the encrypted output is XORed with its current plaintext block to produce the ciphertext block. Since the ciphertext is fed back into next round as the seed, it is called the Cipher Feedback mode of operation.

(4) OFB (Output Feedback)

The OFB mode of operation is similar to CFB mode. The initialization vector (*IV*) is applied to the encryption algorithm and then fed back into next round as the seed. The first ciphertext block is computed by XORing the first plaintext block with the encrypted *IV*, while the *i*th ciphertext block is derived by XORing the *i*th plaintext block with the encrypted output of the $(i - 1)$th encrypted result. Each block operation depends on all previous ones.

It can be seen that all modes of operation except ECB have a common sequential procedure, *i.e.*, an encryption/decryption algorithm can not start to process until the previous operation finished, which incurs low efficiency in multi-processor structures. So far, several other modes [9-12] have been developed, such as CTR (counter), CCM, EAX, GCM, OCB, LRW, CMC and EME modes. Among them, CTR mode is designed for parallel computation. Yet, it can not maintain the non-linearity of *IV*s as CBC, OFB and CFB modes. In this paper, we combine the CBC mode of operation and the binary tree data structure to propose a new structural binary CBC encryption mode which allows parallelized computing and still maintains the non-linearity of *IV*s. A significant property of the proposed mode is that each branch operation is independent of the others and can perform in parallel without interfering with each other. When applied in multi-

processor structures, the structural binary CBC encryption mode can make effective use of CPUs and gain better performance in terms of the computing time.

The rest of this paper is organized as follows. Section 2 states some preliminaries for facilitating readers with the following contexts. Section 3 introduces the structural binary CBC encryption mode. We analyze its security and performance in sections 4. Finally, a conclusion with the significance of our proposed mode of operation is given in section 5.

2. PRELIMINARIES

This section first gives the complete definitions of symbols, operations, functions, inputs and outputs with respect to the proposed structural binary CBC encryption mode.

2.1 Symbols

$|b|$: The bit length of the block b;
Y_i: The ith chain text block;
P_i: The ith plaintext block;
C_i: The ith ciphertext block;
IV: The initialization vector;
K: The secret key.

2.2 Operations and Functions

$\oplus$: The bitwise exclusive-OR operation;
$RT(X)$: The transformation function that changes the bit sequence of the message block X;
$E_K(P_i)$: The encryption function that encrypts the plaintext block P_i under the secret key K;
$D_K(C_i)$: The decryption function that decrypts the ciphertext block C_i under the secret key K.

2.3 Inputs and Outputs

This mode of operation has three inputs while there is only one output. We define these parameters as follows:

Input (*K, IV, P*)

- A secret key K whose length is appropriate for the underlying block cipher.
- An initialization vector IV with the same length of the block cipher.
- A plaintext P of any length.

Output (*C*)

- A ciphertext C having the same length of the plaintext P.

The inputs and outputs are described in terms of bit strings. Note that the initialization vector IV is public information and it must be generated randomly for any particular

execution of the encryption process. The primary purpose of the *IV* is a kind of nonce, *i.e.*, to be distinct from each encryption algorithm under the fixed secret key, so that encrypting the same plaintext twice will generate different ciphertexts.

3. STRUCTURAL BINARY CBC ENCRYPTION MODE

The structural binary CBC encryption mode has the structure of binary tree, whose encryption process XORs the plaintext block with the previous ciphertext block and the first plaintext block is XORed with the initialization vector *IV*. Details of the encryption/ decryption procedures are stated below:

3.1 Encryption Procedures

From section 2.3, we know the inputs of the proposed mode of operation have three parameters (*K*, *IV*, *P*). First, we compute the *i*th chain text block as Eq. (1).

$$Y_i = \begin{cases} IV & i = 1 \\ C_{\lceil i/2 \rceil} & i > 1 \text{ and } i \equiv 0 \mod 2 \\ RT(C_{\lceil i/2 \rceil}) & i > 1 \text{ and } i \equiv 1 \mod 2 \end{cases} \quad (1)$$

That is, the first chain text block Y_1 is just the initialization vector *IV*, the *i*th even chain text block is $C_{\lceil i/2 \rceil}$ and the *i*th odd chain text block is the transformation function of $C_{\lceil i/2 \rceil}$, denoted as $RT(C_{\lceil i/2 \rceil})$.

In the root round, the input block of the encryption function *E* is formed by XORing the first plaintext block P_1 with the *IV*. The encryption function *E* processes under the provided secret key *K* and its corresponding outputs constitute the ciphertext block C_1. In the second round, the ciphertext C_1 is XORed with the second plaintext block P_2 to produce the input block of the encryption function for the even branch tree. The odd branch tree starts with the third round which takes the transformed ciphertext C_1 XORed with the third plaintext block P_3 as the input block of the encryption function, so as to output the resulting ciphertext block C_3. Note that both the second and the third rounds are in the same level. It can be seen that for the *i*th even branch tree, the input block of the encryption function *E* is the outputted ciphertext block $C_{\lceil i/2 \rceil}$ of its parent node XORed with the plaintext block P_i. Whereas the odd branch tree takes $RT(C_{\lceil i/2 \rceil})$, the transformed ciphertext block $C_{\lceil i/2 \rceil}$ of its parent node, XORed with the plaintext block P_i as the input block of the encryption function. Here, each encryption function *E* of either even or odd branch tree processes under the fixed secret key *K* and all generated ciphertext blocks C_i's will constitute the whole ciphertext *C*. More precisely, we can derive the equality of C_i as Eq. (2).

$$C_i = E_K(P_i \oplus Y_i) \quad (2)$$

The encryption diagram of the structural binary CBC encryption mode is shown as Fig. 1.

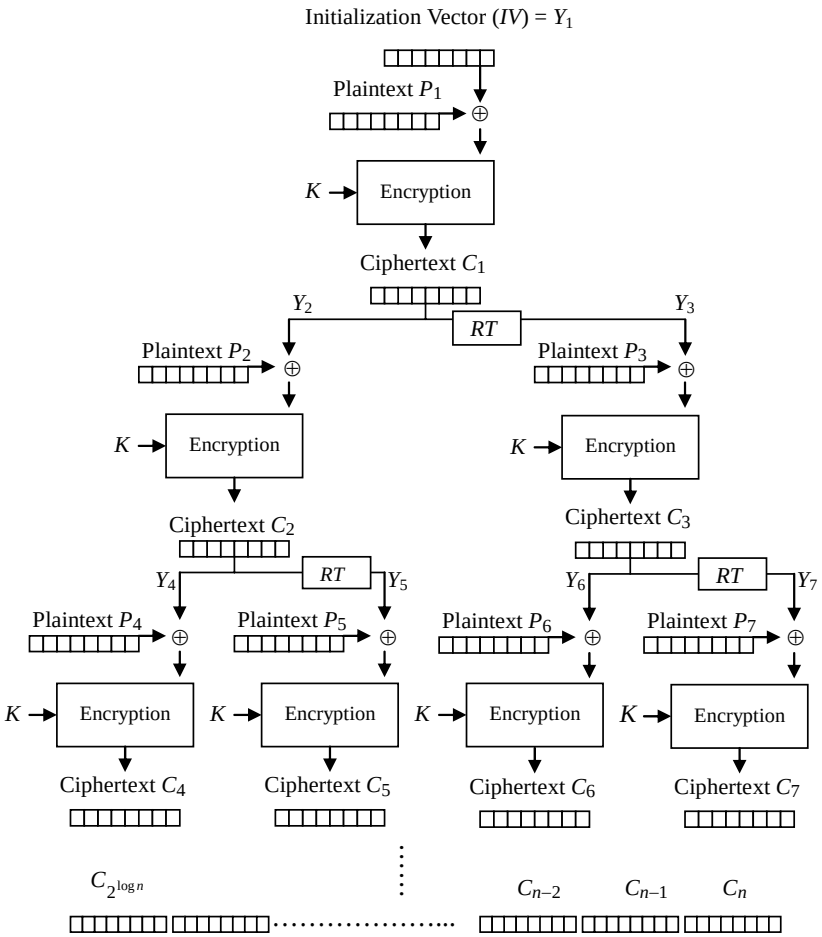

Fig. 1. Encryption diagram of the structural binary CBC encryption mode.

3.2 Decryption Procedures

In the decryption procedure, the first plaintext block P_1 is computed by XORing the initialization vector IV with $D_K(C_1)$, the decrypted ciphertext block C_1. For the even branch tree, the ith ciphertext block C_i is first applied to the decryption function D under the fixed secret key K and then the corresponding output is XORed with the input $C_{\lceil i/2 \rceil}$ of its parent node to recover the plaintext block P_i. Whereas the odd branch tree XORs the ith ciphertext block C_i with $RT(C_{\lceil i/2 \rceil})$, the transformed ciphertext block $C_{\lceil i/2 \rceil}$ of its parent node to recover the plaintext block P_i. After all decryption function $D_k(\cdot)$'s finish, these recovered plaintext P_i's will constitute the original plaintext P. We can generalize the equality of P_i as Eq. (3).

$$P_i = D_K(C_i) \oplus Y_i \tag{3}$$

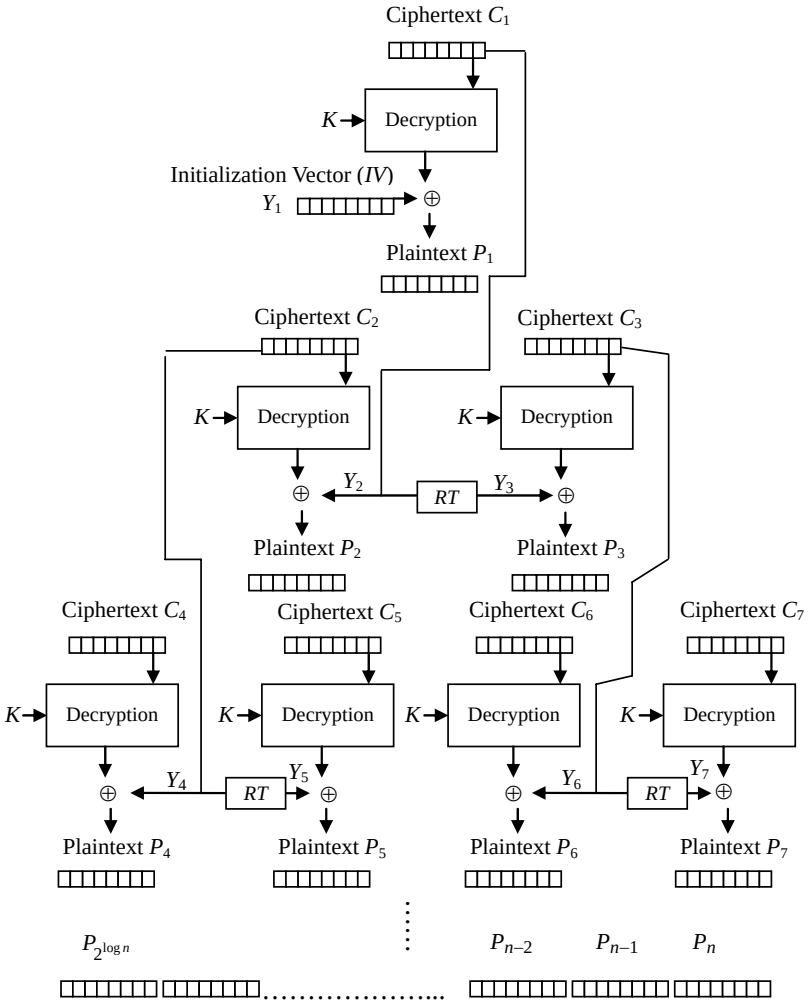

Fig. 2. Decryption diagram of the structural binary CBC encryption mode.

The decryption diagram of the structural binary CBC encryption mode is demonstrated as Fig. 2.

In the above encryption/decryption procedures, each round only depends on the output/input of its parent node and thus procedures of the even branch tree will not affect those of odd branch tree and vice versa. In other words, both even and odd branch operations can perform in parallel in multi-processor structures, so as to greatly improve the overall performance.

4. SECURITY AND PERFORMANCE ANALYSES

The underlying building block of the structural binary CBC encryption mode is ac-

tually CBC mode of operation. If we start with the root and always choose its left successor as the next node, such sequential encryption mode will turn out to be the CBC block cipher. Consequently, the security considerations of CBC can be directly applied to those of the proposed structural binary CBC encryption mode. Its security relies on the fact that each ciphertext block is indistinguishable from a random permutation and the unpredictability of the initialization vector (*IV*). Since the CBC block cipher is shown to be secure against various attacks such as ciphertext only, known plaintext and chosen ciphertext attacks, we can claim that the proposed structural binary CBC encryption mode is also secure against these attacks. Table 1 evaluates the performance of our scheme with original CBC and CTR modes for encrypting/decrypting n message blocks assuming n processors can be used together.

Table 1. Performance evaluation of the proposed scheme with other modes.

Item \ Scheme	Proposed scheme	Original CBC	Original CTR
Waiting time	$O(\log n)$	$O(n)$	$O(1)$
Completion time	$O(\log n)$	$O(n)$	$O(1)$
Relation between *IV*	Non-Linear	Non-Linear	Linear

As shown in Table 1, the proposed scheme outperforms the original CBC mode in terms of waiting and completion time under multi-processor structures. Compared with the CTR mode which is designed for parallel computation and has been widely used in high-speed network standards, the proposed scheme has lower performance. However, our scheme does not have to implement the successive counter circuit and the non-linearity of *IV*s provides stronger security than the CTR mode.

5. CONCLUSIONS

To overcome the low efficiency of traditional modes of operation due to the sequential procedure, in this paper, we proposed a new structural binary CBC encryption mode combining the original CBC mode of operation and the binary tree data structure. The encryption/decryption procedures can be separated into two parts of even and odd branch operations, respectively. Without interfering with each other, each branch operations can perform independently and parallelized computing is feasible in multi-processor structures, which will greatly improve the overall performance in terms of the computing time. Further, since the proposed encryption mode is based on the operation of CBC, we conclude that the structural binary CBC encryption mode is as secure as CBC.

REFERENCES

1. A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Inc., 1997.
2. W. Stallings, *Cryptography and Network Security: Principles and Practices*, 3rd ed.,

Prentice Hall, 2002.
3. American National Standards Institute, "Data encryption algorithm, ANSI X3.92-1981," 1980.
4. National Institute of Standards and Technology, "Federal information processing standards publication 197, Advanced Encryption Standard (AES)," 2001.
5. R. L. Rivest, "The RC4 encryption algorithm," RSA Data Security, Inc., 1992.
6. G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *Journal of American Institution of Electronic Engineering*, Vol. 45, 1926, pp. 252-259.
7. American National Standards Institute, "Data encryption algorithm – Modes of operation, ANSI X3.106-1983," 1983.
8. National Bureau of Standards, "DES modes of operation, federal information processing standards publication (FIPS PUB) 81," U.S. Department of Commerce, 1980.
9. M. Dworkin, "Recommendation for block cipher modes of operation: Methods and techniques," NIST Special Publication 800-38A, 2001.
10. National Institute of Standards and Technology, "Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) for confidentiality and authentication," draft Special Publication 800-38D, 2007.
11. P. Rogaway, M. Bellare, and J. Black, "OCB: A block-cipher mode of operation for efficient authenticated encryption," *ACM Transactions on Information and System Security*, Vol. 6, 2003, pp. 365-403.
12. B. Schneier, *Applied Cryptography*, 2nd ed., John Wiley & Sons, 1996.

Yi-Shiung Yeh (葉義雄) received a M.S. and a Ph.D. degrees in Department of Electrical Engineering and Computer Science from University of Wisconsin-Milwaukee, U.S.A. in 1980 and 1986, respectively. He died as a professor in the department of Computer Science, National Chiao Tung University in July 2007. His research interests include data security and privacy, information and coding theory, game theory, reliability, and performance.

Ting-Yu Huang (黃定宇) received his B.S. degree in electrical engineering from National Central University, Taiwan in 1990, and his M.S. degree in computer science and information engineering from National Chiao Tung University, Taiwan in 2001. Now he is a doctoral student in the Department of Computer Science of National Chiao Tung University, Taiwan. His research interests include cryptanalysis and communication protocols.

Han-Yu Lin (林韓禹) received his B.A. degree in economics from the Fu-Jen University, Taiwan in 2001, and his M.S. degree in information management from the Huafan University, Taiwan in 2003. Now he is a doctoral student in the Department of Computer Science of National Chiao Tung University, Taiwan. His research interests include cryptography and network security.