A Fast Encryption Mode for Block Cipher with Integrity Authentication

Like Chen, Runtong Zhang
Institute of Information Science and Technology
Dalian Maritime University
Dalian, China
chenlike8888@sina.com

Runtong Zhang
School of Economics and Management
Beijing Jiaotong University
Beijing, China
rtzhang@bjtu.edu.cn

***Abstract*—Most important things of secure communications are integrity and confidentiality. However, seldom cipher mode could achieve these two goals simultaneously. So, in this paper, a novel cipher mode is proposed to solve this scheme. The newly proposed mode is based on a previous work PCBC, which has the vulnerability that it permits the modification of swapping two ciphertext, which could pass the integrity authentication. In our algorithm, by adding another XOR operation with a counter to the mode, we successfully solve the vulnerability of PCBC, so we named it C-PCBC. In addition, our algorithm could be more efficient than another modification M-PCBC, thus it will be very useful in the case where confidentiality and integrity are both needed.**

Keywords-Integrity authentication; low cost cipher modes

I. INTRODUCTION

Accompany with the development of wireless communication and internet technology, more and more M-commerce services become part of people's lives. However, for every commerce service, secure problem is always a key point, which has two basic schemes, integrity and confidentiality. Since more and richer data are involved in the commerce information, communication protocol environment which preserve data confidentiality and provide integrity with modest computational costs is urgently required, especially in wireless environment which have limitation in computational capability.

In the designing process of Kerberos Version4[1], a cipher mode, named PCBC(Plaintext Cipher Block Chaining), was created. It was based on the CBC mode, but it added a new XOR operation to combine the previous plaintext block with the current plaintext block and also with the previous cipher text block obtained. So the alteration of the ciphertext would be propagated to the last cipher block. In this way, by adding to the end of the message a fixed last plaintext block, the destination of the communication can verify the integrity of the message simply by decrypting the last block and checking that the fixed plaintext has not changed.

Regrettably, the PCBC mode has not been formally published as standard because there is a vulnerability discovered by Kohl in 1990[2]. This vulnerability permits that if two ciphertext blocks are swapped, then the result of the decryption of the last block still yields the correct fixed plaintext. Though the practical consequences of this flaw are not obvious, PCBC was replaced by CBC-MD5 mode in Kerberos Version5.

In [3], a modified mode based on PCBC, called M-PCBC (Memory-Plaintext Cipher Block Chaining) is designed. This mode use an extra memory of one-block size to store the information of the past cipher blocks. With this modification it is possible to propagate any error to the last block of the message. Nevertheless, there is an increment in the computational cost of the modified mode. In average, M-PCBC add one XOR operation to each block.

In this paper, a novel mode which is also based on PCBC is proposed. In order to solve the flaw in PCBC, a counter is introduced into the new mode to mark each block output, so it is named C-PCBC (Counter – Plaintext Cipher Block Chaining). With this modification, we can not only remove the vulnerability in PCBC, but lower the extra computational cost in M-PCBC. This modification also makes the mode more suitable for parallelization.

This paper is organized as follows: Section 2 briefly reviews some operational modes of block ciphers. Section 3 analyzes the PCBC mode and M-PCBC mode. Section 4 describes the newly proposed cipher mode C-PCBC, and proves its security against the vulnerability in PCBC. The computational cost of C-PCBC with M-PCBC is also compared in this section. We conclude in Section 5.

II. CIPHER MODES

It is well known that, block cipher is the base of security systems, and the confidentiality of the communication is usually implemented through block cipher algorithms. A block cipher algorithm is a type of encryption algorithm where the plaintext is divided into blocks of the same length, which are ciphered independently with the same ciphering key and algorithm. Examples of block ciphers are: the widely known as Data Encryption Standard (DES)[4], which has been the standard encryption algorithm for the United States during more than 30 years, and also its successor Advance Encryption Standard (AES)[5].

Block ciphers usually implement ciphering modes, which combine the blocks with certain basic operations and feedbacks. Operations used must be computational fast and efficient and must increase the security of the ciphertext,

This work was partially supported by the National Science Fund of China under grant number 60773033.

978-1-4244-2013-1/08/$25.00 ©2008 IEEE

making its cryptanalysis more difficult. In any case, confidentially of the ciphertext is only based on the robustness of the cipher algorithm but never in the ciphering mode.

The simplest cipher mode is the Electronic Code Book (ECB) that only consist on ciphering each plaintext block independently. Although this mode is very simple, its simplicity allows the parallelization of the ciphering process, which is very useful in terms of computational cost associated to the encryption. Obviously the ECB does not supply any integrity validation, and hence does not supply any integrity validation, and hence modifications, intentional or not, of the ciphertext.

Although the ECB mode is the simplest cipher mode, it is not frequently used in practice. The most widely used cipher mode is the Cipher Block Chaining (CBC). In this mode, each plaintext block is operated, by the XOR function, with the ciphertext produced in the previous block. The result of this operation is ciphered as the result of this block (the process is shown in fig.1-2). By only adding one additional XOR operation to each block, two identical plaintext blocks produce different ciphertext blocks, and this fact makes the cryptanalysis harder. To decrypt a ciphered message, the process must be inverted.

CBC mode has many virtues and is used as default in many communication protocols, but from Fig.1, we see that one bit alteration in certain ciphertext block will only produce errors in two plaintext blocks, when the message is decrypted. So, CBC mode can not supply any mechanism for the integrity verification either.

Our problem is not only to find out a mechanism for the verification of the integrity of the message, but to find an efficient mechanism that does not delay the encryption process. With this in view the Massachusetts Institute of Technology designed a new cipher mode, which is based on CBC mode, to solve this problem. The new cipher mode is called Plaintext Cipher Block Chaining (PCBC). This mode makes possible to detect integrity alterations in ciphertext. In the next section, we will cover the PCBC mode and its modification in [3].

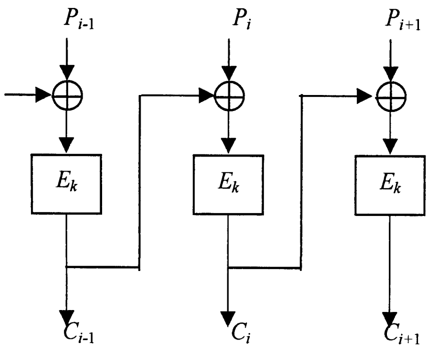

Figure 1. CBC Encryption.

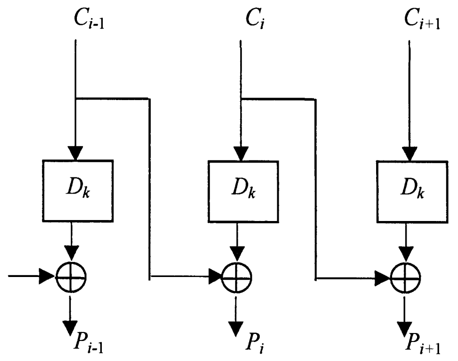

Figure 2. CBC Decryption.

III. PCBC AND M-PCBC

The PCBC mode was created for the Kerberos Version 4[1]. It was based on the CBC mode, but added a new XOR operation. In fact, the PCBC combine the previous plaintext block with the current plaintext block and also with the previous ciphertext block obtained. Fig.3 shows the process for encryption and decryption of PCBC mode.

From fig.3, it can be seen that PCBC can propagate any alteration of the ciphertext to the last block. In this way, by adding to the end of the message a fixed plaintext block, the destination can verify the integrity of the message, simply by decrypting the last block and checking that the fixed plaintext has not changed.

Regrettably, the PCBC mode has not been formally published as standard because there is a vulnerability discovered by Kohl in 1990 [2]. This vulnerability permits that if two ciphertext blocks are swapped, the result of the decryption of the last block still yields the correct fixed plaintext. PCBC was replaced by CBC-MD5 mode in Kerberos Version 5.

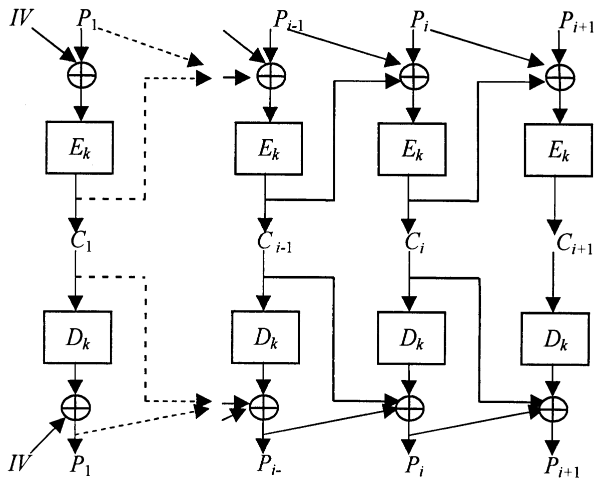

Figure 3. Encryption and decryption of PCBC mode.

The following algebraic expressions describe the operation process of the PCBC mode. Let us consider P_i the ith plaintext block, C_i the ith ciphertext block, E_K and D_K the block encryption and decryption algorithm, which use the key K, respectively.

$$\begin{aligned}
C_{i-1} &= E_k(P_{i-1} \oplus P_{i-2} \oplus C_{i-2}) \\
C_i &= E_k(P_i \oplus P_{i-1} \oplus C_{i-1}) \\
C_{i+1} &= E_k(P_{i+1} \oplus P_i \oplus C_i) \\
P_{i-1} &= D_k(C_{i-1}) \oplus P_{i-2} \oplus C_{i-2} \\
P_i &= D_k(C_i) \oplus P_{i-1} \oplus C_{i-1} \\
P_{i+1} &= D_k(C_{i+1}) \oplus P_i \oplus C_i
\end{aligned} \tag{1}$$

From (1), it is easy to obtain:

$$\begin{aligned}
P_{i+1} &= D_k(C_{i+1}) \oplus D_k(C_i) \oplus P_{i-1} \oplus C_{i-1} \oplus C_i \\
P_{i+1} &= D_k(C_{i+1}) \oplus D_k(C_i) \oplus D_k(C_{i-1}) \oplus P_{i-2} \oplus C_{i-2} \oplus C_{i-1} \oplus C_i
\end{aligned} \tag{2}$$

If we swap the ciphertext block C_{i-1} and C_i, then the plaintext is:

$$\begin{aligned}
P_{i-1}' &= D_k(C_i) \oplus P_{i-2} \oplus C_{i-2} \\
P_i' &= D_k(C_{i-1}) \oplus P_{i-1} \oplus C_i \\
P_{i+1}' &= D_k(C_{i+1}) \oplus P_i \oplus C_{i-1}
\end{aligned} \tag{3}$$

Operating with the above expressions, it is possible to obtain:

$$\begin{aligned}
P_{i+1}' &= D_k(C_{i+1}) \oplus D_k(C_{i-1}) \oplus P_{i-1} \oplus C_i \oplus C_{i-1} \\
P_{i+1}' &= D_k(C_{i+1}) \oplus D_k(C_{i-1}) \oplus D_k(C_i) \oplus P_{i-2} \oplus C_{i-2} \oplus C_i \oplus C_{i-1}
\end{aligned} \tag{4}$$

It is clear that expressions (2) and (4) are identical. This proves that when swapping two ciphered blocks, the following ones will not suffer any modification.

Though the PCBC has this weakness, the idea of this mode is still meaningful. So, José et al modified PCBC in [3]. As shown in fig.4, the new mode is based on adding a memory to the original PCBC that stores information of the past ciphertext blocks.

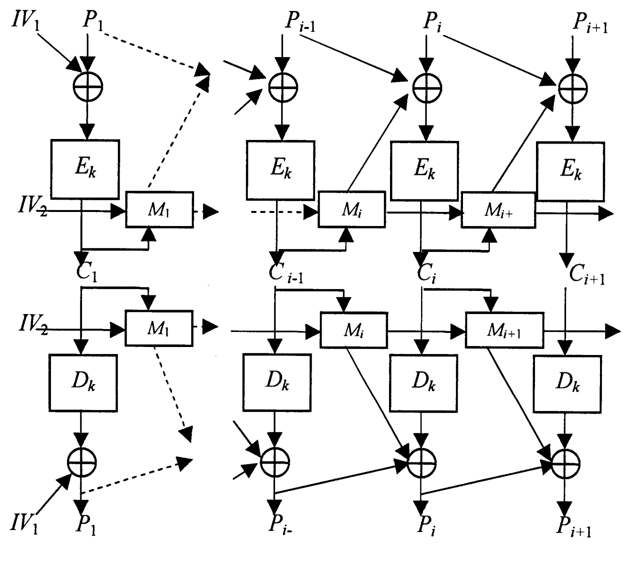

Figure 4. Encryption and decryption of M-PCBC mode.

Here, the length of the memory M, is equal to the length of the cipher block size n. Then, M is divided into two variables ML and MR, represent the left and right part of the memory respectively, both of length n/2. The M will be initialized with the IV_2, and every block M_i will be mixed according to the following expressions:

$$ML_i = ML_{i-1} \oplus MR_{i-1}, \quad MR_i = CL_{i-1} \oplus CR_{i-1} \tag{5}$$

The M-PCBC is defined as the following algebraic expressions.

$$\begin{aligned}
C_{i-1} &= E_k(P_{i-1} \oplus P_{i-2} \oplus M_{i-2}) \\
C_i &= E_k(P_i \oplus P_{i-1} \oplus M_{i-1}) \\
C_{i+1} &= E_k(P_{i+1} \oplus P_i \oplus M_i) \\
P_{i-1} &= D_k(C_{i-1}) \oplus P_{i-2} \oplus M_{i-2} \\
P_i &= D_k(C_i) \oplus P_{i-1} \oplus M_{i-1} \\
P_{i+1} &= D_k(C_{i+1}) \oplus P_i \oplus M_i
\end{aligned} \tag{6}$$

From (1), it is easy to obtain:

$$\begin{aligned}
P_{i+1} = P_{i-2} \oplus D_k(C_{i-1}) \oplus D_k(C_i) \oplus D_k(C_{i+1}) \oplus \\
\lfloor ML_{i-2} \oplus MR_{i-2} \oplus CR_{i-2} \oplus CL_{i-2} \oplus CL_{i-1} \oplus CR_{i-1} \,\|\, CR_i \oplus CL_i \rfloor
\end{aligned} \tag{7}$$

Then, if we swap the ciphertext block C_{i-1} and C_i in the modified mode, then the plaintext is:

$$\begin{aligned}
P_{i+1}' = P_{i-2} \oplus D_k(C_{i-1}) \oplus D_k(C_i) \oplus D_k(C_{i+1}) \oplus \\
\lfloor ML_{i-2} \oplus MR_{i-2} \oplus CR_{i-2} \oplus CL_{i-2} \oplus CL_i \oplus CR_i \,\|\, CR_{i-1} \oplus CL_{i-1} \rfloor
\end{aligned} \tag{8}$$

Obviously, expression (7) and (8) are not equal because both parts of the memory contents are different. Though the modification solved the vulnerability of PCBC, averagely, it adds one XOR operation to each data word, the word size depending on different processors. And it is our belief that the efficiency of M-PCBC could still be improved. So, we did a new modification of PCBC, which saves part of computational cost in comparison with M-PCBC.

IV. C-PCBC MODE

In this paper, we also present a new modification of the PCBC algorithm, which solved the commented flaw and supplies efficient integrity validation. Our new algorithm is based on adding a counter to differentiate the output of the block cipher and so we name the new mode C-PCBC. With this information it is possible to propagate the error to the last block of the message, using less computational cost than the M-PCBC.

Fig 5-6 show the encryption and decryption stage of C-PCBC respectively. From the figures, it can be seen that C-PCBC is very similar with PCBC. The only difference between PCBC and C-PCBC is that a counter is XORed with the output of block encryption in the encryption stage, and the result is the C-PCBC ciphertext. In the decryption stage, the ciphertext should be first XORed with the counter, and then inputed into the decryption function of the block cipher. The counter could be initialized by a random vector, determined by different applications. The C-PCBC is defined by the following algebraic expressions.

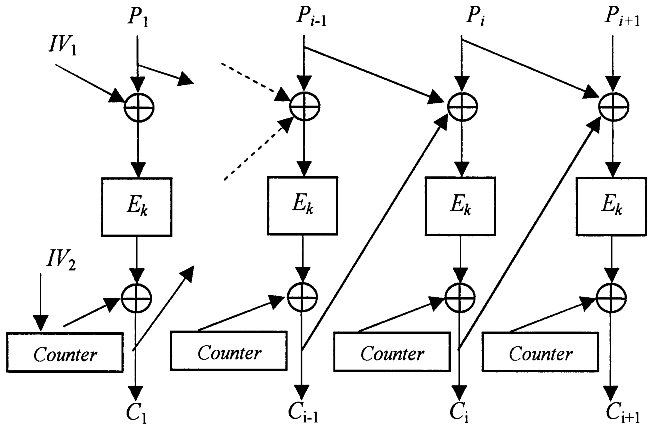

Figure 5. Encryption of C-PCBC

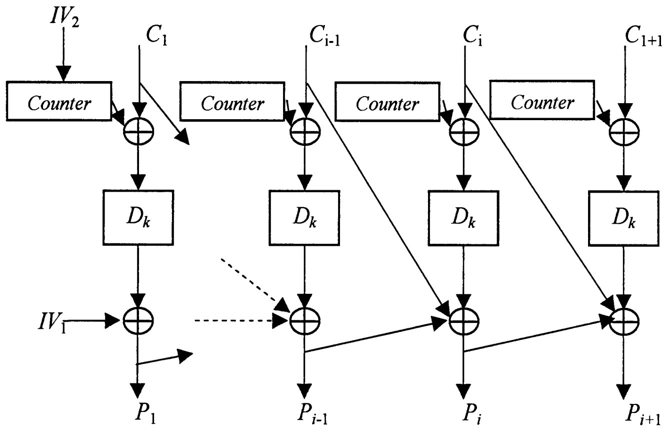

Figure 6. Decryption of C-PCBC

$$
\begin{aligned}
C_{i-1} &= E_k(P_{i-1} \oplus P_{i-2} \oplus C_{i-2}) \oplus Counter(i-1) \\
C_i &= E_k(P_i \oplus P_{i-1} \oplus C_{i-1}) \oplus Counter(i) \\
C_{i+1} &= E_k(P_{i+1} \oplus P_i \oplus C_i) \oplus Counter(i+1) \\
P_{i-1} &= D_k(C_{i-1} \oplus Counter(i-1)) \oplus P_{i-2} \oplus C_{i-2} \\
P_i &= D_k(C_i \oplus Counter(i)) \oplus P_{i-1} \oplus C_{i-1} \\
P_{i+1} &= D_k(C_{i+1} \oplus Counter(i+1)) \oplus P_i \oplus C_i
\end{aligned} \tag{9}
$$

From (9), we obtain

$$
\begin{aligned}
P_{i+1} = P_{i-2} \oplus C_{i-2} \oplus C_{i-1} \oplus C_i \oplus D_k(C_{i+1} \oplus Counter(i+1)) \\
\oplus D_k(C_i \oplus Counter(i)) \oplus D_k(C_{i-1} \oplus Counter(i-1))
\end{aligned} \tag{10}
$$

Then, if we swap the ciphertext block C_{i-1} and C_i in C-PCBC mode, the decrypted plaintext is:

$$
\begin{aligned}
P_{i-1}' &= D_k(C_i \oplus Counter(i-1)) \oplus P_{i-2} \oplus C_{i-2} \\
P_i' &= D_k(C_{i-1} \oplus Counter(i)) \oplus P_{i-1}' \oplus C_i \\
P_{i+1}' &= D_k(C_{i+1} \oplus Counter(i+1)) \oplus P_i' \oplus C_{i-1}
\end{aligned} \tag{11}
$$

Operating with (11), it is possible to obtain:

$$
\begin{aligned}
P_{i+1}' = P_{i-2} \oplus C_{i-2} \oplus C_i \oplus C_{i-1} \oplus D_k(C_{i+1} \oplus Counter(i+1)) \\
\oplus D_k(C_{i-1} \oplus Counter(i)) \oplus D_k(C_i \oplus Counter(i-1))
\end{aligned} \tag{12}
$$

Since $D_k(C_i \oplus Counter(i)) \neq D_k(C_{i-1} \oplus Counter(i))$, $D_k(C_{i-1} \oplus Counter(i-1)) \neq D_k(C_i \oplus Counter(i-1))$, it is easy to obtain that the probability that (10) equals (12) is about2^{-n}. Here, n is the cipher block size. Since n is large enough (generally 64-bit or 128-bit), we believe that $P_{i+1} \neq P_{i+1}'$, i.e. C-PCBC could resist the attack proposed in [2].

Different from M-PCBC, our algorithm use a counter to differentiate the ciphertext, and the algorithm need not to use a counter of the full block size of the cipher. For example, if we use 128-bit block cipher in the 32-bit processor PC environment, only one 32-bit word XOR operation (only 1 cycle) is needed. 3/4 of the extra computational cost of the M-PCBC is saved. In addition, in contrast to M-PCBC, our work can decrease the data dependence in consecutive instructions, so it is more suitable for modern processors, and would be more efficient in software implementation. Hence, C-PCBC is more efficient than M-PCBC.

V. Conclusion

This paper proposed a novel cipher mode. The new mode is based on the PCBC mode, which is a candidate proposal for Kerberos Version4, for simultaneously providing confidentiality and integrity authentication by adding another XOR operation with the previous plaintext block to the CBC mode. However, PCBC has a flaw that the swap of the ciphertext will not alarm the integrity test. In this paper, we modified the PCBC mode and introduce another XOR operation with a counter into the algorithm to differentiate the output of the block cipher, and we name the new mode C-PCBC. Through the analysis, we proved that not only did our algorithm successfully solve the vulnerability of PCBC, but offered more efficiency in software implementation than another modification mode M-PCBC. Thus, C-PCBC is very attractive in practical use.

References

[1] J.G. Steiner, B. Clifford Neuman, J. L. Schiller, "Kerberos: an authentication service for open network systems," Proceedings of 1988 USENIX Conference, 1998, pp. 191-200.

[2] J. T. Kohl, "The use of encryption in Kerberos for network authentication," Advances in Cryptology-Crypto'89, Springer-Verlag, 1990, pp. 35-43.

[3] J. M. Sierra, J. C. Hernández, N. Jayaram, A. Ribagorda, "Low computational cost integrity for block ciphers," Future Generation Computer Systems, vol. 20 , 2004, pp. 857-863.

[4] Data Encryption Standard (DES), FIPS-46, National Institute of Standard and Technology, 1979.

[5] Advanced Encryption Standard, FIPS-197, National Institute of Standards and Technology, 2001.