PROTECTING CIPHER BLOCK CHAINING AGAINST ADAPTIVE CHOSEN PLAINTEXT ATTACK

Chuan-Wen Loe
DSO National Laboratories
20 Science Park Drive, Singapore 118230

Khoongming Khoo
DSO National Laboratories
20 Science Park Drive, Singapore 118230

Keywords: Cipher Block Chaining, Adaptive Chosen Plaintext Attack, Input-Output Masked CBC.

Abstract: In the literature, several encryption modes of operation based on cipher block chaining (CBC) has been proven to be secure under non-adaptive chosen plaintext attack (CPA-1) in the left-or-right (LOR) or find-then-guess (FTG) security models. However, it was shown by Joux et. al. at Crypto 2002 that if we allow the adversary to perform an adaptive chosen plaintext attack (CPA-2), then CBC, ABC and GEM are susceptible to FTG attacks. In this paper, we propose a new CBC-type encryption called input-output masked CBC (IO-CBC) which can protect against FTG and LOR attacks based on forcing an input collision, protects against Joux's FTG attack under proper implementation, and increases the difficulty of linear and differential cryptanalysis. The efficiency of IO-CBC is comparable to CBC because it does only one additonal encryption when compared with CBC. We also reasoned that the security proof of an IO-CBC variant follows from that of OCB.

1 INSECURITY OF CBC-TYPE MODES UNDER CPA-2 ATTACK

The CBC mode is one of the most commonly used encryption mode in practice. Let $E_k(\cdot)$ denote a secure block encryption function with secret key k. CBC can be described as:

Algorithm 1 *CBC Mode:*

Input: Randomly Generated Initial Vector (IV), Plaintext blocks $M[1], M[2], \ldots, M[l]$.

$$\begin{aligned} \textit{Initialize: } O[0] &= IV, \\ \textit{Input: } I[i] &= M[i] \oplus O[i-1], \\ \textit{Output: } O[i] &= E_k(I[i]), \\ \textit{Ciphertext: } C[i] &= O[i], i = 1, \ldots, l. \end{aligned}$$

Output: IV, Ciphertext blocks $C[1], C[2], \ldots, C[l]$.

The CBC mode was proven to be secured against left-or-right distinguishing attack (LOR-secure) (Bellare, 1997, Proposition 15, Lemma 16, Theorem 17) under non-adaptive chosen plaintext attack (CPA-1). However in (Joux, 2002), Joux et. al. proved that under blockwise adaptive chosen plaintext attack (CPA-2), CBC can be distinguished by a find-then-guess (FTG) attack. The FTG attack uses the fact that in CBC under CPA-2, the adversary can force a collision in the input of the block cipher at any two iterations i and j ($j > i$) by setting $M[j] = C[j-1] \oplus C[i-1] \oplus M[i]$. In that case, the ciphertext of iteration i and j will be the same. This fact can also be used to mount a LOR-attack on CBC under CPA-2 assumptions.

Similar LOR and FTG attacks, based on collision of the block cipher input at two blocks, can also be performed against CBC-type encryption modes like the Accumulated Block Chaining (ABC) (Knudsen, 2000) and the Propagating CBC mode (PCBC) (Matyas, 1982). In ABC, the adversary forces a collision as in CBC but instead of comparing $C[i] = C[j]$, he compares $C[i] \oplus M[i-1] = C[j] \oplus M[j-1]$. And in PCBC, to force a collision, the adversary needs to choose a plaintext $M[j]$ such that $M[j] \oplus C[j-1] \oplus M[j-1] = M[i] \oplus C[i-1] \oplus M[i-1]$.

In the LOR and FTG attacks, the adversary is able to force a collision because the "masking" at each block is known, therefore adaptively choose the appropriate plaintext. Moreover, a collision can be verified by observing the ciphertext. This pose an advantage to blackbox cryptanalysis as a stepping stone to determine the mode of operation the encryptor uses.

To protect encryption modes against these attacks is to mask the input and output with some unknown data. One possible candidate is IACBC proposed by Jutla. To encrypt m blocks of data, additional $\log(m)$

Loe C. and Khoo K. (2006).
PROTECTING CIPHER BLOCK CHAINING AGAINST ADAPTIVE CHOSEN PLAINTEXT ATTACK.
In *Proceedings of the International Conference on Security and Cryptography*, pages 135-140
DOI: 10.5220/0002100301350140

encryptions of the values $r+1, \ldots, r+\log(m)$ where r is secret. Then these $\log(m)$ encrypted values are expanded using Gray's code to form m pairwise independent secret blocks $S_1, S_2, \ldots, S_m$. The encryption of IACBC is identical to CBC except that the block cipher output $O[i]$ is XORed with S_i to form the ciphertext $C[i]$. Similarly, OCB (Rogaway, 2001) uses (a different) masking to enhance ECB mode.

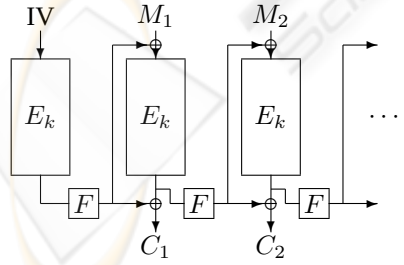
In this paper, we propose a new CBC-type encryption mode called input-output masked CBC mode (IO-CBC). Unlike OCB and IACBC which generates their maskings from an independent (from the plaintext and ciphertext) structured algorithm (Gray's Code), IO-CBC generates its masking from a psedo-random source, the block encryptor.

2 THE INPUT-OUTPUT MASKED CBC MODE

In this Section, we describe the Input-Output Masked CBC mode (IO-CBC). The design goals of IO-CBC are as followed:

1. On-line
2. Memory Free
3. Single Pass
4. Increase effort required in Differential and Linear Cryptanalysis with reused nouce condition
5. Infinite Error Propagation (Better Diffusion)
6. Single Key
7. Minimize Encryption

It is a slight modification of the CBC mode we masked both the block cipher input and output with the previous output block transformed by a specially chosen linear function $F(\cdot)$.



Algorithm 2 *Input-Output Masked CBC Mode:*

$F(\cdot)$ = A non-singular linear function such that $F^i(s) \oplus F^j(s)$ preserves the entropy of s.
Input: Randomly Generated Initial Vector (IV),
Plaintext blocks $M[1], M[2], \ldots, M[l]$.

$$
\begin{aligned}
\textit{Initialize: } O[0] &= E_k(IV), \\
\textit{Input: } I[i] &= M[i] \oplus F(O[i-1]), \\
\textit{Output: } O[i] &= E_k(I[i]), \\
\textit{Ciphertext: } C[i] &= O[i] \oplus F(O[i-1]).
\end{aligned}
$$

Output: IV, Ciphertext blocks $C[1], C[2], \ldots, C[l]$

As we shall see in Section 4, it is important that the encrypted IV $O[0] = E_k(IV)$ to be kept secret in this scheme. To preserve the entropy of $O[0]$, we need a non-singular linear function such that that it is as hard to guess $F^i(O[0]) \oplus F^j(O[0])$ as guessing $O[0]$.

3 EFFICIENCY OF IO-CBC

In IO-CBC, the operations are similar to CBC except for an extra linear transform $F(O[i-1])$ for input-output masking at iteration i. The overall complexity is still equivalent to one block encryption per iteration because the linear function computation can be considered free on most platforms. Thus the efficiency of IO-CBC is only one block encryption more than CBC where the extra work is for encrypting the IV.

Since IO-CBC maskings are dependant from the previous block, hence it is an on-line encryption and does not required additional memory. In comparison, some CBC-type encryption like IACBC cannot do on-line encryption because it requires the sender to know the length of the whole plaintext beforehand (say m blocks) to do $\log(m)$ additional encryption for the Gray's Code masking.

4 SECURITY

In this Section, we consider adaptive chosen plaintext attack on IO-CBC. The assumption is that the adversary can choose the plaintext $M[i]$ after observing the ciphertext $C[1], C[2], \ldots, C[i-1]$. He has no control over the initial vector (IV), which is generated by the encryptor.

4.1 Protection Against Attacks based on Forcing an Input Collision

In FTG and LOR attacks on CBC-type encryption modes under CPA-2, the adversary tries to force an input collision. This is achieved by choosing an appropriate plaintext block $M[i]$, which when XORed with a known input masking, will produce a desired input value $I[i]$. For example, to force a collision in CBC, an adversary will choose an $M[j]$ which satisfies $M[j] \oplus C[j-1] = M[i] \oplus C[i-1]$, $j > i$,

where the input masking-values $C[i-1], C[j-1]$ are known. This is not possible for IO-CBC because the adversary cannot deduce the input masking.

In IO-CBC, an adversary can control the input $I[i]$ at iteration i by choosing an appropriate plaintext block $M[i]$ if the input mask $F(O[i-1])$ is known because $I[i] = M[i] \oplus F(O[i-1])$. But to know $F(O[i-1])$, the adversary needs to XOR the known value $C[i-1]$ with the unknown value $F(O[i-2])$ to obtain $O[i-1]$. Similarly, to know $F(O[i-2])$, the adversary needs to XOR $C[i-2]$ with $F(O[i-3])$ to obtain $O[i-2]$. Inductively, we deduce that:

$$F(O[i-1]) = F(C[i-1] \oplus F(C[i-2] \oplus \ldots \oplus F(C[1] \oplus F(O[0]) \ldots)))). \quad (1)$$

$IV, C[1], \ldots, C[i]$ are known values but adversary does not know $O[0] = E_k(IV)$ which is not transmitted. Although the adversary knows the IV, he cannot deduce $E_k(IV)$ from it because E_k is a secure encryption function.

Because $F(\cdot)$ is a linear function, $F(x \oplus y) = F(x) \oplus F(y)$ and we can write equation (1) as:

$$F(O[i-1]) = F(C[i-1]) \oplus F^2(C[i-2]) \oplus \ldots \oplus F^{i-1}(C[1]) \oplus F^i(O[0]). \quad (2)$$

Together with the equation $I[i] = M[i] \oplus F(O[i-1])$, we see that for $i > j$, we can force a collision $I[i] = I[j]$ if we choose $M[i]$ as follows.

$$\begin{aligned} M[i] &= M[j] \oplus F(O[i-1]) \oplus F(O[j-1]) \\ &= M[j] \oplus F(C[i-1]) \oplus F^2(C[i-2]) \oplus \\ &\ldots \oplus F^{i-j}(C[j]) \oplus F^i(O[0]) \oplus F^j(O[0]). \end{aligned}$$

We note that the only unknown summand in the above equation is $F^i(O[0]) \oplus F^j(O[0])$. The condition we impose on $F(\cdot)$ ensures that the entropy of $F^i(O[0]) \oplus F^j(O[0])$ is the same as $O[0]$ which is kept secret. Thus an adversary cannot force a collision.

One possible choice for the function $F(\cdot)$ is to let it be the shift register matrix of a linear feedback shift register (LFSR) whose feedback polynomial is a primitive polynomial $x^n + c_{n-1}x^{n-1} + \ldots + c_2x^2 + c_1x + c_0$.

$$F = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & c_0 \\ 1 & 0 & 0 & \ldots & 0 & c_1 \\ 0 & 1 & 0 & \ldots & 0 & c_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & c_{n-1} \end{pmatrix} \quad (3)$$

The output sequence of such an LFSR is usually called a binary m-sequence.

Proposition 1 *(Shift and add property for binary m-sequences (Golomb, 1982, Chapter 3)) Let $F(\cdot)$ be the shift register matrix of an n-bit LFSR with primitive feedback polynomial (see equation (3)). Let the initial state of this LFSR be s. Then for any $0 \leq i \neq j \leq 2^n - 2$, there exist $0 \leq k \leq 2^n - 2$ such that $F^i(s) \oplus F^j(s) = F^k(s)$.*

By Proposition 1, the entropy of $F^i(s) \oplus F^j(s)$ will be the same as that of s. Therefore the linear map in equation (3) is a suitable choice for the function $F(\cdot)$ in Algorithm 2. However, since LFSR is a shift and generates an unknown most significant bit, hence $LFSR^i(x)$ is similar to $LFSR^{i+1}(x)$. To prevent attacks that exploit this property, let $F^i(x) = LFSR^i(x)$ and $F^{i+1}(x) = LFSR^{i+B}(x)$, where B is the block size.

4.2 Elimination of Output Masking

For the adversary, in addition to the difficulty of controlling the input $I[i]$ in IO-CBC, it is also difficult for him to deduce the encryption output $O[i]$. This is because he can only observe the ciphertext $C[i]$ which is $O[i]$ XORed with an unknown masking. However in the LOR and FTG attacks of (Alkassar, 2001; Bellare, 1997; Joux, 2002), it is essential that the adversary be able to deduce the output (e.g. to determine when an output collision $O[i] = O[j]$ occurs in a CBC attack) to distinguish the plaintext message that is being encrypted.

To get around the effect of the secret masking at the output is to find a linear combination of the output masks which sums to a known value. This is the basic idea used in (Joux, 2002; Sung, 2003) where they used the relation $S_1 \oplus S_2 \oplus S_3 \oplus S_4 = 0$ to attack IACBC. However, the attack in (Joux, 2002) can be prevented if we keep the encrypted IV secret. We show that IO-CBC can be protected in a similar way.

For the case when $F(\cdot)$ is a linear function in IO-CBC, there exist a linear combination which sums to a known value. This is because $m(F) = 0$ when $m(x) = \bigoplus_{i \in I} x^i$ is the minimal polynomial of the linear function $F(\cdot)$. Thus

$$\bigoplus_{i \in I} F^i(O[0]) = 0.$$

together with equation (2), this implies:

$$\bigoplus_{i \in I} F(O[i-1]) = \bigoplus_{i \in I} (F(C[i-1]) \oplus F^2(C[i-2]) \oplus \ldots \oplus F^{i-1}(C[1])). \quad (4)$$

Therefore we have a linear combination of output masks that sum up to a known value.

It can be shown that Joux's FTG attack on IACBC can be applied to IO-CBC in a similar way using equation (4); but under the assumption that the encrypted IV, $E_k(IV) = O[0]$, is given under flawed implementation. However if $O[0]$ is given, IO-CBC will be reduced into a CBC since from equation (2),

all masking will be revealed. Hence adversary can attack flawed implemented IO-CBC as though it is a CBC mode, and Joux's attack will be unnecessary.

IO-CBC is designed for a single key encryption, however if it is extended to two keys, having separate keys for each of the masking and data, the key recovery attack for IACBC in (Sung, 2003) cannot be applied on IO-CBC. The attack in (Sung, 2003) expressed the maskings S_i in terms of plaintext and ciphertext, and eliminate the linear masking with $S_1 \oplus S_2 \oplus S_3 \oplus S_4 = 0$. Thus we get an expression involving just the plaintext, ciphertext and key k_1. The key k_0 that is used for the secret masking is excluded, which reduces the key space complexity. Thus we can perform exhaustive key search on k_0 and k_1 separately to deduce them with a complexity of $5 \cdot 2^n$ instead of 2^{2n}, where n is the size of each key.

Based on equation (4), we consider the key recovery attack of (Sung, 2003) on a 2-key variant of IO-CBC. In this variant, we use a different key k_0 for encryption of IV ($O[0] = E_{K_0}(IV)$) and k_1 for the plaintext. We need to express the secret masking $F(O[i-1])$ in terms of $M[i]$, $C[i]$ and K_1:

$$\begin{aligned} C[i] &= O[i] \oplus F(O[i-1]) \\ F(O[i-1]) &= C[i] \oplus E_{k_1}(I[i]) \\ F(O[i-1]) &= C[i] \oplus E_{k_1}(M[i] \oplus F(O[i-1])) \end{aligned}$$

For ease of explaination, suppose there exist an output mask relation: $F(O[a-1]) \oplus F(O[b-1]) \oplus F(O[c-1] = 0$. Then

$$\begin{aligned} &E_{k_1}(M[a] \oplus F(O[a-1])) \oplus \\ &E_{k_1}(M[b] \oplus F(O[b-1])) \oplus \\ &E_{k_1}(M[c] \oplus F(O[c-1])) \\ &= C[a] \oplus C[b] \oplus C[c] \end{aligned}$$

Although the output masking is eliminated, the input masking exists and from equation (2), we see that k_0 remains as a factor in $F(O[a-1]), F(O[b-1]) and F(O[c-1])$. Hence we are unable to exclude k_0 from k_1, and brute force complexity remains at 2^{2n}.

4.3 Resistance to Differential and Linear Cryptanalysis

In this section, we shall see that IO-CBC makes differential and linear cryptanalysis harder.

Linear cryptanalysis is a known plaintext attack while differential cryptanalysis is a chosen chosen plaintext attack. For example in DES, the adversary needs about 2^{43} known plaintext-ciphertext pairs to launch a successful linear attack (Matsui, 1994) and 2^{47} chosen plaintext-ciphertext pairs to launch a differential attack (Biham, 1992).

In our context, we name a plaintext-ciphertext pair an input-output pair. IO-CBC makes it hard for the adversary to collect known input-output pairs. The reason is because we have XOR-masked the plaintext $M[i]$ with $F(O[i-1])$ to form the input $I[i]$. And as explained before, knowing $F(O[i-1])$ is equivalent to knowing $O[0] = E_k(IV)$ which is kept secret. Thus known input attack cannot be applied. Moreover, the input masking is secret and thus cannot control the encryption input by choosing an appropriate plaintext. Thus chosen input attack cannot be performed. Even if adversary managed to control or deduce the input, he will have difficulty in deducing the output from the ciphertext because of the secret masking $F(O[i-1])$.

One possibility to proceed with chosen and known input-output attacks is that the adversary can guess the value of $O[0] = E_k(IV)$ and apply linear or differential cryptanalysis for each of these guess. In that case, we need to multiply the attack complexity of differential or linear cryptanalysis by 2^B where B is the block size of the block cipher E_k. This greatly increase the complexity of the attack.

The other way, is to sent the first message session and from the returned ciphertext, adaptively adjust the plaintext for the next session (assume same IV). This is so since from equation (2), the unknown data $F^i(O[0])$ is blockwise similar in both sessions. However after the 1^{st} block of the 2^{nd} session, a different ciphertext (from 1^{st} session) will masked the 2^{nd} block of the 2^{nd} session. Hence the only practical attack it to perform an adaptive blockwise chosen plaintext attack with reused IV.

Note that in two IO-CBC sessions which uses the same IV, the input and output differences are the same as the plaintext and ciphertext differences at the first block. However, they will be different from block two onwards. Therefore another way to perform differential attack is to form plaintext differences at the first block in many IO-CBC sessions with the same IV. However, this is not feasible in practice because the IV is randomly generated and not controlled by the adversary while we will need around 2^{47} IO-CBC sessions with the same IV to perform differential attack on DES.

Re-use of IV may pose a threat to OCB and IACBC, in particular, differential attack can be applied on OCB since the input-output masking in both sessions are equal and will cancel each other at every block. Thus all the blocks of two OCB sessions which uses the same IV can be used in a differential attack. For example, if a IO-CBC session transmits 2^{40} blocks of encrypted data, then we only need $2^7 = 128$ sessions that uses the same IV to perform differential attack on DES, which is much less than

the 2^{47} sessions needed in IO-CBC.

Thus IO-CBC increases the work required for differential cryptanalysis as compared to OCB under reused nouce condition.

4.4 Isomorphism to OCB

There are many other linear functions that the authors have yet to research on to ensure a secure mode of operation. However, IO-CBC and OCB share similar features of masking both the input as well as the output of the block encryptor using linear transformation of unknown data.

Hence IO-CBC will be isomorphic to OCB if the non-singular linear (in GF(2)) function $F(\cdot)$ in IO-CBC is the masking function in OCB, $F^i(O[0]) = \gamma_i \cdot O[0] \oplus O'[0]$. Where γ_i is the Gray's code representation of integer i and $O'[0]$ is the encrypted result of $O[0]$.

From equation (2), we have:

$$F(O[i-1]) = G(C[i-1] \cdots C[1]) \oplus F^i(O[0]) \quad (5)$$

Where $G(C[i-1] \cdots C[1])$ is a linear function of known values, and $F^i(O[0])$ is the masking technique used in OCB. Since $G(C[i-1] \cdots C[1])$ is known, it does not provide resistance to any distinguishing algorithm. Hence $F^i(O[0])$ is the only masking in IO-CBC, which is similar to OCB. So it is obvious that IO-CBC with linear function $F^i(O[0]) = \gamma_i \cdot O[0] \oplus O'[0]$ is a general case of OCB.

Given the security proof of OCB (Rogaway, 2001), we can be sure that IO-CBC with linear function (5) will be at least as secure as OCB. Furthermore using the provable security and authenticity in (Rogaway, 2001), we can be sure that IO-CBC can be extended to a variant with both privacy and authenticity in a single pass mode of operation.

5 ERROR PROPAGATION

When a legitimate user receives a ciphertext, he can recover the plaintext by performing IO-CBC in reverse.

$$\begin{aligned} O[0] &= E_k(IV), \\ O[i] &= C[i] \oplus F(O[i-1]), \\ I[i] &= D_k(O[i]),\ D_k \text{ is the decryption function.} \\ M[i] &= I[i] \oplus F(O[i-1]), i = 1, \ldots, l. \end{aligned}$$

If an error bit occurs at $C[i]$ during transmission, it will cause a bit error in $O[i]$. This will cause $I[i]$ and $M[i]$ to be garbled. Similarly, $O[i+1] = C[i+1] \oplus F(O[i])$ will have some bit errors because of bit errors in $F(O[i])$. This will cause $I[i+1]$ and $M[i+1]$ to be garbled. Inductively, we see that $M[i], M[i+1], \ldots, M[l]$ will all be garbled. Therefore IO-CBC has infinite error propagation.

Thus IO-CBC is suitable for an environment where we do not allow for any error in the received ciphertext, e.g. we do not want the message to be tampered with during transmission. The infinite error propagation property of IO-CBC is a very good way to detect whether a transmitted ciphertext has been tampered with, by introducing a checksum as we shown a variant in Section 4.4.

Error propagation also provides more resistance to birthday attack. In (Knudsen, 2000), the *matching ciphertext attack* is described as follows:

Fact 1 *Consider an n-bit block cipher used in the ECB, CBC or CFB mode. It is assumed that the plaintext blocks are chosen at random from a uniform distribution. If s blocks are encrypted under the same key, information is leaked about some plaintext blocks with a probability of* $p_s = 1 - (1 - 2^{-n})^{s(s-1)/2}$. *When* $s = 2^{(n+1)/2}$, *this probability is about 0.63.*

This attack basically tries to detect if there is a collision in the plaintext blocks by observing the ciphertext. E.g. for ECB, $C_i = C_j \Rightarrow P_i = P_j$. In (Knudsen, 2000), it is stated that error propagation modes are generally less prone to such an attack, this will include ABC and IO-CBC.

Lastly, another advantage of error propagation is to provide better diffusion for encryption operations (Knudsen, 2000). Suppose the plaintext space has low entropy, say each plaintext block can take only s values where s is small. If we are using ECB mode, then it is easy to deduce that the plaintext has low entropy by observing the ciphertext. However, if $C_i = g_k(M_i, \cdots, M_{i-v}, C_{i-1}, \cdots, C_{i-w})$, then any ciphertext block can take up to s^{v+w} values. If $s < 2^{B/(v+w)}$ then some B-bit values will never occur, where B is the block size. Hence in (Knudsen, 2000), the way to provide more diffusion is to make v and w big or to construct a mode such that each ciphertext depends on all previous plaintext and ciphertext blocks. When such a property is satisfied, it will introduce infinite error propagation mode as in ABC and IO-CBC.

6 VARIANT WHERE THE INPUT-OUTPUT MASK IS A NONLINEAR FUNCTION

In IO-CBC, we have used linear functions for the transformation $F(\cdot)$. The rational for using a linear transformation is to reduce the computational overhead of the masking. To prevent future attacks that

exploit the linear property of $F(\cdot)$, it might be useful to replace it by an efficient computable nonlinear function.

Usually, the security of a block cipher against standard attacks like differential and linear cryptanalysis is higher than required. For example, by the wide-trail design principle of AES (Daemon, 2002), the maximum differential characteristic probability of AES is $(4/256)^{25} = 2^{-150}$ for 4 rounds, $(4/256)^{50} = 2^{-300}$ for 8 rounds and $(4/256)^{75} = 2^{-450}$ for 12 rounds. However, we just need the characteristic differential probability to be smaller than 2^{-128} for protection against differential attack. A similar estimate holds for the strength of AES against linear cryptanalysis.

In that case, we can afford to reduce the block cipher by one round and let $F(\cdot)$ be an unkeyed round of the block cipher. E.g., for AES-256, we can reduce it to 13 rounds and let $F(\cdot)$ be an AES round without XORing with a subkey. The performance is still equivalent to one block encryption per iteration because we are taking one round out of the block cipher to form the function $F(\cdot)$. Therefore the overall efficiency of IO-CBC is one encryption more than CBC where the extra work is for encrypting the IV.

As in the analysis of Section 4, this variant will satisfy equation (1). Because $F(\cdot)$ is a nonlinear function, equation (1) cannot be simplified and $F(O[i-1])$ is an increasingly complex function of the secret $O[0] = E_k(IV)$ as i increases. Thus it is more difficult for the adversary to deduce $F(O[i-1])$ and control the input $I[i]$ to force a collision by choosing $M[i]$.

With respect to Joux's FTG attack in (Joux, 2002) or Sung's key recovery attack in (Sung, 2003), it may be difficult to find a linear combination of the output mask $F(O[i-1])$ which sums to a known value because the recursion of $F(\cdot)$ in equation (1) will result in a complex highly nonlinear function.

7 CONCLUSION AND FUTURE WORK

In this paper, we have introduced a new CBC-type mode of operation called IO-CBC. We have shown that it is as efficient as CBC mode and provides protection against various adaptive chosen plaintext attacks introduced in (Joux, 2002; Sung, 2003). It also makes differential and linear cryptanalysis harder than other modes of operation like ECB, CBC and OCB. Tbe IO-CBC mode has infinite error propagation which makes it suitable for applications that needs to detect occurence of any errors during transmission. From section 4.4, we get a provable security similar to OCB given that the linear function is similar to OCB. A useful problem for future research is to establish the provable security of IO-CBC and a wider class of linear function $F(\cdot)$.

Having confidence that a variant of IO-CBC is isomorphic to OCB in section 4.4, another problem for future research is to extend IO-CBC to a provable variant, like IACBC and OCB, that does both confidentiality and authentication. Together with the infinite propagation property, tampering with transmitted ciphertext will be easily detected.

REFERENCES

Alkassar, A., Geraldy, A., Pfitzmann, B. and Sadeghi, A. R. (2001). Optimized Self-Synchronizing Mode of Operation. LNCS 2335, *Fast Software Encryption 2001*. Springer-Verlag.

Bellare, M., Desai, A., Jokipii, E. and Rogaway, P. (1997). A Concrete Security Treatment of Symmetric Encryption. Proceedings of *Foundations of Computer Science'97*. IEEE Press, 1997.

Biham, E. and Shamir, A. Differential Cryptanalysis of the Full 16-Round DES. LNCS 740, *Crypto '92*, Springer-Verlag, 1993.

Daemon, J. and Rijmen, V. *The Design of Rijndael: AES - The Advanced Encryption Standard*, Springer, 2002.

Golomb, S.W. *Shift Register Sequences*, Revised Edition, Agean Park Press, 1982.

Joux, A., Martinet, G. and Valette, F. Blockwise Adaptive Attackers: Revisiting the (In)Security in some Provably Secure Encryption Modes: CBC, GEM, IACBC. LNCS 2442, *Crypto 2002*, pp. 17-30, Springer-Verlag, 2002.

Jutla, C. Encryption Modes with Almost Free Message Integrity. LNCS 2045, *Eurocrypt 2001*, pp. 529-544, Springer-Verlag, 2001.

Knudsen, L. Block Chaining Modes of Operation. Technical Report, Department of Informatics, University of Bergen, 2000.

Matsui, M. The First Experimental Cryptanalysis of the Data Encryption Standard. LNCS 839, *Crypto '94*, pp. 1-11, Springer-Verlag, 1994.

Matyas, M. and Matyas, S. *Cryptography: A New Dimension in Computer Data Security*, John Wiley and Sons, New York, 1982.

Preneel, B., Nuttin, M., Rijmen, V. and Buelens, J. Cryptanalysis of DES in the CFB mode. LNCS 773, *Crypto '93*, pp. 212-223, Springer-Verlag, 1994.

Rogaway, P., Bellare, M., Black, J. and Krovetz, T. OCB: A block-cipher mode of operation for efficient authenticated encryption. *http://www.cs.ucdavis.edu/ rogaway*, 2001.

Jaechul Sung, Deukjo Hong and Sangjin Lee Key Recovery Attacks on RMAC, TMAC, and IACBC *LNCS 2727*, pp. 265-273, 2003.