J. Cryptology (2002) 15: 47–59
DOI: 10.1007/s00145-001-0016-5

Journal of
CRYPTOLOGY
© 2002 International Association for Cryptologic Research

Cryptanalysis of the ANSI X9.52 CBCM Mode*

Eli Biham
Computer Science Department,
Technion – Israel Institute of Technology,
Haifa 32000, Israel
biham@cs.technion.ac.il
http://www.cs.technion.ac.il/~biham/

Lars R. Knudsen
Department of Informatics, University of Bergen,
Hi-techcenter, N-5020 Bergen, Norway
lars@ramkilde.com
http://www.ramkilde.com

Communicated by Ivan Damgård

Received May 1998 and revised June 2001
Online publication 28 November 2001

Abstract. In this paper we cryptanalyze the CBCM mode of operation, which was almost included in the ANSI X9.52 Triple-DES Modes of Operation standard. The CBCM mode is a Triple-DES CBC variant which was designed against powerful attacks which control intermediate feedback for the benefit of the attacker. For this purpose, it uses intermediate feedbacks that the attacker cannot control, choosing them as a keyed OFB stream, independent of the plaintexts and the ciphertexts. In this paper we find a way to use even this kind of feedback for the benefit of the attacker, and we present an attack which requires a single chosen ciphertext of 2^{65} blocks which needs to be stored and 2^{59} complexity of analysis (CBCM encryptions) to find the key with a high probability. As a consequence of our attack, ANSI decided to remove the CBCM mode from the proposed standard.

Key words. Cryptanalysis, ANSI X9.52, Modes of operation, CBCM mode, Triple-DES, Multiple encryption.

1. Introduction

The Data Encryption Standard (DES) [21] has been the subject of intense debate and cryptanalysis. Already at the introduction of the algorithm in the seventies the DES was criticized for its short key length of 56 bits. As illustrated by Wiener [24], [25],

* A preliminary version of this paper was presented at EUROCRYPT '98 in Finland, June 1998 [8]. The first author was supported by the fund for the promotion of research at the Technion.

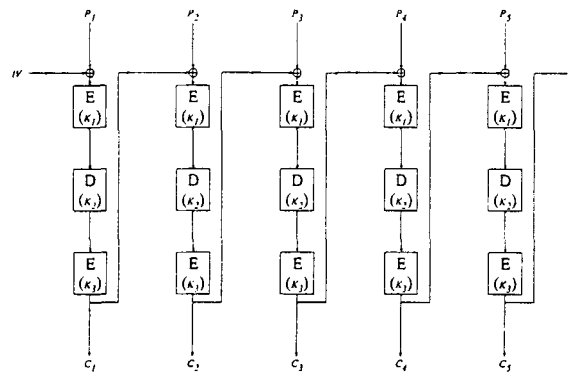


Fig. 1. The outer-CBC mode.

by the recent exhaustive searches over the Internet [12], [13], and by the EFF *deep crack* machine [14] it has become feasible for a powerful attacker to search exhaustively for a DES key. For a higher level of security it is therefore often recommended to use Triple-DES, which encrypts a plaintext three times with three different keys or to use two-key Triple-DES, which encrypts a plaintext with a key K_1, then decrypts with a key K_2, and finally encrypts again with key K_1. This increases the key lengths to 168 and 112, respectively. Direct application of Triple-DES increases the security with respect to key-recovery attacks, but due to the short 64-bit blocksize, still allows for dictionary and matching ciphertext attacks[1] with success probabilities similar to those for single-DES. Therefore, the most popular way to use triple encryption is by performing triple modes of operation. These modes apply three or more DES encryptions for each block and mix the data further in intermediate stages using data obtained from the previous blocks.

One popular mode of operation is the Triple-DES Cipher Block Chaining (TCBC) mode, where the feedback block is the ciphertext block (computed by three DES encryptions). This mode is also called the *outer-CBC* mode [15]. This mode is described in Fig. 1. However, this mode is vulnerable to the matching ciphertext attack. Therefore, it has been proposed to use Triple-DES in a cipher block chaining mode with internal feedback, called the *inner-CBC* mode [15], where the feedback is applied after each single DES encryption. This mode is described in Fig. 2. This mode is not vulnerable to the matching ciphertext attack, and was expected to be as secure as three-key Triple-DES against key recovery attacks. As the best published attack against three-key Triple-DES required 2^{112} complexity [20] it was expected that attacks against this mode require more than 2^{112} operations. (Note that recently Lucks [17] devised a new attack which slightly reduces the complexity of attacking three-key Triple-DES to 2^{108}).

However, in a series of papers [3], [4], [6], [7], the first author analyzed a large number of multiple modes of operation, and in particular showed how to mount a key-recovery

[1] In these attacks the attacker collects intercepted ciphertext blocks and stores them in a table. Once he has two matching ciphertext blocks he can deduce information of the corresponding plaintext blocks [16], [19], [9]. The probability of success is around 0.5 after the interception of $t = 2^{n/2}$ blocks where n is the block size. For increasing values of t the probability of success quickly approaches 1.

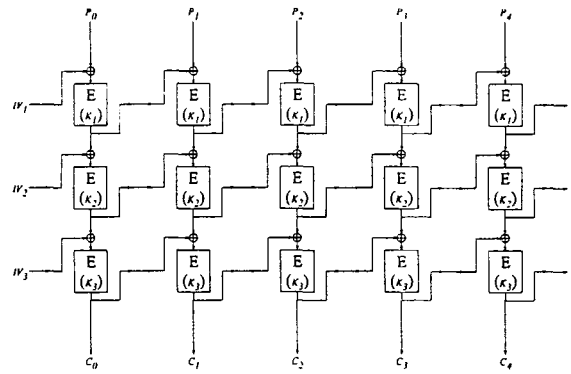


Fig. 2. The inner-CBC mode.

attack [6] against the inner-CBC mode. The complexity of this attack is considerably smaller than one would expect for triple encryption schemes and it is only slightly higher than the complexity of attacking single modes.

In [10] and [9] Coppersmith et al. propose the *CBC with OFB Masking* (CBCM) mode of operation for Triple-DES. The CBCM mode was specially designed to withstand the attacks described in [6] and [7], the dictionary attack, and the matching ciphertext attack. The disadvantage of the proposal is that it uses four DES encryptions using three different DES keys to encrypt each 64-bit plaintext block. In [10] and [9] it is mentioned that the attacks in [6] that use internal feedbacks for the benefit of the attacker leave little hope for devising modes with internal feedbacks. In particular, it is mentioned that the inner-CBC mode is weak due to such feedbacks, while on the other hand modes with only outer feedbacks are unsatisfactory. This motivated the design of a more complex mode which has both outer feedbacks and internal feedbacks, but the internal feedbacks may not be controlled by the attacker, as they are the output of an OFB mode. It is claimed in [10] that the CBCM mode is immune against the kind of attacks described in [6].

Wagner analyzed an early CBCM proposal with an adaptive chosen-IV ciphertext attack. This attack led to the current design, in which only 2^{20} values are allowed for one of the two initial values [10], [9]. The best two attacks that the designers have sought require 2^{34} chosen ciphertexts and 2^{90} complexity of analysis (respectively 2^{44} chosen ciphertexts and 2^{80} complexity of analysis).

In this paper we cryptanalyze the CBCM mode. In our attack we use the internal OFB stream together with the common key K_1 of the first and last DES components for the benefit of the attacker. The attack requires one chosen ciphertext of 2^{65} blocks and 2^{59} complexity of analysis. It stores all the 2^{65} ciphertext blocks in memory. The attack is applicable even if the initial values are not known to (nor chosen by) the attacker. We believe that this attack uncovers a major weakness in the design of this mode. The design took similar (chosen ciphertext) attacks into consideration and was proposed as a replacement for modes which did not resist similar attacks.

The CBCM mode was part of the American National Standards Institute (ANSI) Triple-DES Modes of Operation proposed standard [2]. This standard was almost accepted in September 1997, and was delayed only in order to correct some typos found in

the proposal. The corrected version had to be finally accepted a few weeks later, and this attack was found just before this final vote. As a consequence, ANSI decided to accept the standard [1] without the CBCM mode [18].

In the remainder of this paper we assume that the reader is familiar with the basic ideas of the attacks on multiple modes of operation [6], [7]. As in [6] and [7], we describe the complexity of an attack by the number of encrypted blocks and the time of analysis required to find the key with a high probability.

The paper is organized as follows: In Section 2 we describe the CBCM mode, and in Section 3 we describe our attack. In Section 4 we propose several possible improvements for the CBCM mode. Finally, in the Appendix we show that if an attacker can get encryptions under several different but related keys, then the keys can be found using considerably less data.

2. The CBCM Mode

The CBCM mode is similar to the outer-CBC mode when two-key Triple-DES is used as the underlying cipher, with the following modification:

- between the first and second components, and between the second and third components, mixing with an OFB mask is applied. The same mask is used in both applications. The mask is the output of an OFB mode using a third key.

In the following E and D respectively denote encryption and decryption with the underlying block cipher. Consider the plaintext $P = P_1, \ldots, P_\ell$ consisting of ℓ blocks, and the user-selected key $K = K_1, K_2, K_3$. In the CBCM mode the ciphertext $C = C_1, \ldots, C_\ell$ is computed as follows:

$$C_i = E_{K_1}(D_{K_2}(E_{K_1}(P) \oplus M_i) \oplus M_i), \tag{1}$$

where

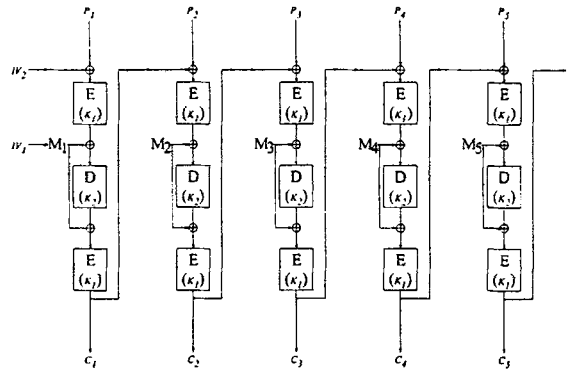
$$M_i = E_{K_3}(M_{i-1}), \tag{2}$$

and $M_0 = IV_1$, $C_0 = IV_2$ are the initial values. Figure 3 describes this mode.

3. The Attack Against the CBCM Mode

The attack we describe is a chosen ciphertext attack, i.e., in this attack it is assumed that the attacker does not know any keys, key-relations, nor initial values; the attacker only chooses one ciphertext stream, and receives the decrypted plaintext under the unknown key and unknown initial values.

The attack goes as follows. Choose two ciphertext block values C_1 and C_2. Request the plaintexts of *one* ciphertext stream of 2^{64} C_1's followed by 2^{64} C_2's, under the unknown key $K = (K_1, K_2, K_3)$ (and any initial values). The period p of the OFB component can be easily identified in at most 2^{64} simple steps from the resulting plaintexts. The expected value of p is about 2^{63} [11]. In the following description we ignore the first plaintext block, as we do not know the feedback IV_2 mixed with it. For simplicity of description we also ignore the first block with the ciphertext C_2.



$M_1, M_2, \ldots$, etc, are the output blocks of an OFB mode with IV_1 as the initial value, encrypted under the key K_3:

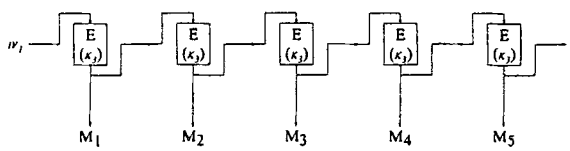


Fig. 3. The CBCM mode.

Denote the plaintexts of the first p blocks (after removal of the original first block) by $P_{1,1}, \ldots, P_{1,p}$, and denote the inputs to the first DES components (in the triple encryption steps) by $Q_{1,i} = P_{1,i} \oplus C_1$ for $i = 1, \ldots, p$. By construction, $Q_{1,i}$ is encrypted to C_1 with the OFB feedback M_i. For each i choose the corresponding block encrypted to C_2 using the same M_i, and denote its plaintext by $P_{2,i}$ (this is easy as the period p of the OFB component is known already). Denote $Q_{2,i} = P_{2,i} \oplus C_2$. This notation does not necessarily take $P_{2,1}, \ldots, P_{2,p}$ in the order they are received, but in an order easier to analyze, as for any i the same OFB block M_i is used in the computation of $P_{1,i}$ and $P_{2,i}$.

If we would now guess K_1, we can encrypt the first DES component, and decrypt the last component, and we are left with only the middle component using the key K_2 surrounded by two masks of M_i. As the masks are unknown, we cannot continue this way and find K_2 without increasing the complexity considerably. However, for this guess of K_1, we can compute $E_{K_1}(Q_{k,i}) \oplus D_{K_1}(C_{k,i})$. If the guessed K_1 is correct, this computed value equals the XORs of the input and the output of the middle DES component, regardless of the unknown value of M_i.

We cannot use this observation directly for an attack, but we can use it as a hint that we should find some technique to hide the existence of the M_i's. The proposed technique uses the double occurrence of each M_i in the computation, and uses fixed points to cancel the effect of the M_i's.

The attack exploits that with a high probability the function $E_{K_2}(\cdot) \oplus V$ has one fixed point for randomly chosen V's (for example, a fixed point of $E_{K_2}(\cdot)$ when $V = 0$). If by some chance this fixed point appears in our data, the whole triple encryption with the OFB mask reduces to something similar to double encryption with only one key K_1 (to

simplify the following discussion, we describe decryption of this mode):

$$\begin{aligned} Q_{k,i} &= D_{K_1}(M_i \oplus E_{K_2}(M_i \oplus D_{K_1}(C_{k,i}))) \\ &= D_{K_1}(M_i \oplus V \oplus (M_i \oplus D_{K_1}(C_{k,i}))) \\ &= D_{K_1}(V \oplus D_{K_1}(C_{k,i})) \end{aligned}$$

and when $V = 0$

$$= D_{K_1}(D_{K_1}(C_{k,i})).$$

We use this property in our attack.

The attack proceeds in the following manner:

1. Choose some arbitrary block value V.
2. Do the following steps for each candidate K' for the key K_1:
3. Compute

$$\begin{aligned} T_1(K') &= D_{K'}(V \oplus D_{K'}(C_1)), \\ T_2(K') &= D_{K'}(V \oplus D_{K'}(C_2)), \end{aligned}$$

   and search the plaintext for blocks i, j matching

$$Q_{1,i} = T_1(K') \quad \text{and} \quad Q_{2,j} = T_2(K').$$

4. If both matches are found compute

$$U = D_{K'}(C_1) \oplus D_{K'}(C_2)$$

   (which also necessarily equals $U = E_{K'}(Q_{1,i}) \oplus E_{K'}(Q_{2,j})$) and verify whether

$$E_{K'}(Q_{2,i}) \oplus E_{K'}(Q_{1,j}) = U$$

   (notice that the indices are interchanged!).
5. If the verification succeeds, the key K_1 is probably K'. If it fails, try the next candidate.
6. If the verification fails for all candidates, choose another V, and repeat the analysis using the same data.

Figure 4 gives some details, where we assume for the purpose of illustration that $i = 1$ and $j = 4$, that the fixed point is $x = E_{K_2}(x) \oplus V = y \oplus V$, and that $z = x \oplus M_i \oplus M_j$.

We explain why the attack works. Assume that $K' = K_1$ and that $E_{K_2}(\cdot) \oplus V$ has one fixed point. Assume further that $Q_{1,i} = T_1(K') = D_{K'}(V \oplus D_{K'}(C_1))$ and $Q_{2,j} = T_2(K') = D_{K'}(V \oplus D_{K'}(C_2))$. Thus, we can assume that the output of the decryption step with K_2 equals the fixed point in both cases. Therefore, $D_{K'}(C_1) \oplus D_{K'}(C_2) = M_i \oplus M_j = U$. Consider the encryptions of $Q_{1,j}$ and $Q_{2,i}$. We know that the OFB blocks used are M_j (respectively M_i) and the outputs of the decryption step with K_2 are $D_{K'}(C_1) \oplus M_j$ (respectively $D_{K'}(C_2) \oplus M_i$). However, since the latter two are equal, it must hold that the inputs to the decryption step with K_2 are equal. Thus, it must hold

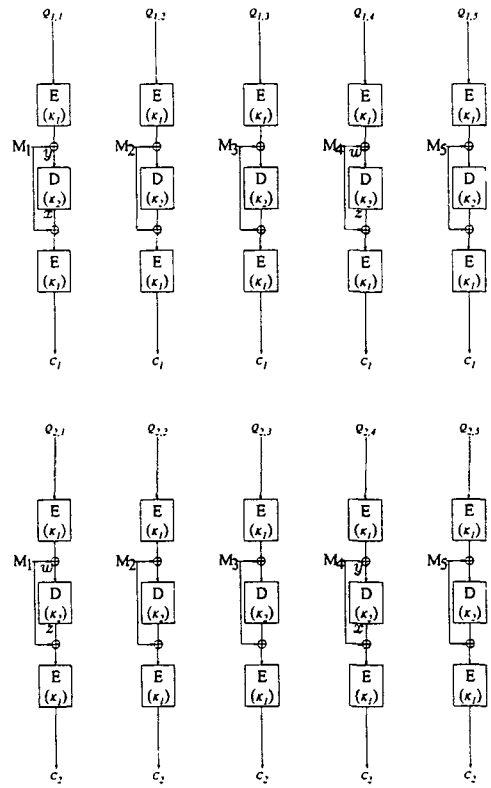


Fig. 4. Some intermediate details of the attack.

that $E_{K'}(Q_{2,i}) \oplus E_{K'}(Q_{1,j}) = M_i \oplus M_j = U$. Clearly, this test fails with a very high probability if $K' \neq K_1$.

Note that the attack still works if there is more than one fixed point for $E_{K_2}(\cdot) \oplus V$. In this case the number of pairs in step 4 above increases slightly for the correct value of K'.

Before we proceed, we wish to elaborate on the probability of success of finding K_1, and methods to improve the probability of success. First, observe that the success of the attack depends on the actual period of the cycle of the OFB component, and that the attacker cannot select the period to his preference. Moreover, the period of the OFB component is uniformly distributed (see [11]). Therefore, we should compute the success rate for each possible period, and then combine the results by the probability distribution of all periods.

Given the stream, denote the period of the cycle of the OFB component by p. Since for each block in the stream there is one value of V such that the encryption of the middle component creates a fixed point of $E_{K_2}(\cdot) \oplus V$, the probability that the randomly selected V makes a fixed point in some block in the cycle is $p/2^{64}$, and the probability that it makes the same fixed point in both cycles is $(p/2^{64})^2$. Therefore, the attack finds the correct value of the key K_1 for a fraction of $p^2/2^{128}$ of the values of V. Since the

period of the OFB component is uniformly distributed, we get that the average success probability is $2^{-64}\sum_{p=1}^{2^{64}} p^2/2^{128} = 2^{-3\cdot 64}\sum_{p=1}^{2^{64}} p^2 \approx 1/3$.

Higher probabilities of success can be reached by repeating the attack with several, say k, V's. The probability of success becomes

$$1 - 2^{-64}\sum_{p=1}^{2^{64}}\left(\frac{1-p^2}{2^{128}}\right)^k \simeq 1 - \int_0^1 (1-x^2)^k\,dx = 1 - \frac{2^{2k}(k!)^2}{(2k+1)!}.$$

For example, for $k = 4$, repeating four times with four different V's increases the complexity of analysis to $4 \cdot 2^{56} = 2^{58}$, and increases the success rate to about 59.4% without affecting the amount of data required. Analyzing eight V's increases the success rate to about 70% with complexity up to 2^{59} (or about 2^{58} on average).

By increasing the number of chosen ciphertexts slightly the complexity of analysis can be further reduced. Choosing three different ciphertext blocks C_1, C_2, C_3 increases the probability of finding *two* identical fixed points in the three cycles to 1/2, and choosing l different ciphertext blocks $C_1, C_2, C_3, \ldots, C_l$, $l > 1$, increases the success rate, i.e., the probability of finding *two* identical fixed points in the l cycles as follows:

$$1 - \int_0^1 \left(\underbrace{lx(1-x)^{l-1}}_{\text{fixed point appears once}} + \underbrace{(1-x)^l}_{\text{fixed point does not appear}} \right)^k dx$$

$$= 1 - \left(\sum_{i=0}^{k}\binom{k}{i} l^i (1-l)^{k-i}\frac{1}{kl-i+1}\right).$$

When $k = 1$ this is

$$1 - \left[1 - \frac{l-1}{l+1}y^{l+1}\right] = \frac{l-1}{l+1} = 1 - \frac{2}{l+1},$$

which is 0.5 when $l = 3$ and 0.6 when $l = 4$.

Table 1 summarizes the success rate for various values of k and l. Note that additional improvements with the same amount of data and complexity are possible, such as using the observation that larger l's are mostly used for short periods p, thus allowing us to reduce the number of blocks of later C_j's and have more C_j's instead.

Once the key K_1 is found, K_2 can be found. Choose a random value a and assume it is decrypted in the second DES component (using key K_2) for one of the encryptions the attacker has already. Since the period p is expected to be 2^{63} this will be the case with probability 1/2. Compute $b = a \oplus D_{K_2}(a)$ for all values of K_2 and check if $b = E_{K_1}(Q_{1,i}) \oplus D_{K_1}(C_1)$ for some i (for any K_2 this check involves only one lookup in a hash table, once all values of $E_{K_1}(Q_{1,i}) \oplus D_{K_1}(C_1)$ are precomputed into the hash table). Note that K_1 is known at this stage. For every candidate of the key K_2 and a corresponding index i, compute $M_i = E_{K_1}(Q_{1,i}) \oplus a$. Since the period of the OFB stream is known we find the encryption in the second collection of p plaintexts where M_i was used, and test whether $Q_{2,i} = D_{K_1}(M_i \oplus E_{K_2}(M_i \oplus D_{K_1}(C_{2,i})))$. This test leaves only very few possible values of the key K_2. If the attack fails, choose another value for a and repeat the attack. This phase of the attack requires two encrypt/decrypt

Table 1. Success rate (%) for various increased values of k and l.

	l								
k	2	3	4	5	6	7	8	16	32
1	33.3	50.0	60.0	66.7	71.4	75.0	77.8	88.2	93.9
2	46.7	62.9	71.4	76.8	80.4	83.1	85.1	92.4	96.1
3	54.3	69.3	76.8	81.3	84.4	86.6	88.2	94.0	97.0
4	59.4	73.3	80.0	84.0	86.7	88.6	90.0	95.0	97.5
5	63.1	76.1	82.2	85.8	88.2	89.9	91.2	95.6	97.8
6	65.9	78.2	83.8	87.2	89.4	90.9	92.1	96.0	98.0
7	68.2	79.8	85.1	88.2	90.2	91.6	92.7	96.4	98.2
8	70.0	81.1	86.1	89.0	90.9	92.2	93.2	96.7	98.3
16	78.3	86.7	90.4	92.3					
32	84.2								

operations for each block to remove the outer encryptions using K_1, and leave only the middle encryption surrounded by mixing with the M_i's. The complexity of this removal is $2 \cdot 2^{63}$ Single-DES encryptions, i.e., equivalent to the CBCM encryption of 2^{62} blocks. The complexity of the rest of the phase is about $2 \cdot 2^{56}$, as on average we need to perform two exhaustive searches for K_2. Thus, the total complexity of this phase is 2^{62}. However, a more careful analysis shows that it can be performed more efficiently by removing the outer encryptions of a smaller number of blocks, at the cost of applying the analysis more times. The total complexity of this phase is then reduced to about 2^{59}.

Once the keys K_1 and K_2 are found, K_3 can be found using the same data as before. Given the single M_i (or one of a few M_i's) recovered in the previous phase, we can try all 2^{56} possible candidates for K_3. Each such candidate proposes a value for $M_{i+1} = E_{K_3}(M_i)$. Then we verify whether the encryption of the next block succeeds with this M_{i+1}. If the verification fails, the values of the keys recovered so far are incorrect. Otherwise, the recovered keys are very likely to be the correct values. This phase of finding K_3 takes about $2 \cdot 2^{56}$ encryption steps.

The complexity of the attack is as follows:

1. Recovering K_1:
   (a) 2^{57} Double-DES encryptions for each chosen value of V, i.e., an equivalent of encrypting 2^{56} blocks with the CBCM mode.
   (b) About 2^{65} chosen ciphertext blocks (assuming $l = 2$).
   (c) Using $k = 4$, the probability of success is larger than $1/2$, and the total complexity is up to 2^{58} (on average it is only slightly more that 2^{57}).
2. Recovering K_2 requires about 2^{61} Single-DES encryption steps, which is equivalent to encryption of 2^{59} blocks with the CBCM mode.
3. Recovering K_3 requires about 2^{56} Single-DES encryption steps.
4. There are also 2^{65} extra operations, each performs an XOR to compute the $Q_{k,i}$'s and then enters the result into a hash table. The total time complexity of these operations is smaller than the time required to encrypt 2^{59} blocks with the CBCM mode.
5. The total time complexity of analysis is expected to be about 2^{59} using 2^{65} chosen ciphertext blocks.

4. Possible Improvements of the CBCM Mode

The presented attack exploits both the internal OFB feedbacks and the use of the same key K_1 in the first and last DES components. As noted in [10] and [9] it is not advisable to use two different OFB feedbacks instead of one. In [7] some modes are cryptanalyzed just when the OFB components are different, and these attacks are not possible when the OFB streams are the same.

The attack can be thwarted by introducing a third OFB component, which will mix into the ciphertext. This is also similar to the modes proposed in [5] and [7]. Changing the key of the last DES component to a fourth key K_4 is also expected to thwart our attack.

Furthermore, computing the initial values IV_1 and IV_2 as the result of a cryptographic keyed one-way function of the transmitted "IV" rather than sending them in the clear, inhibits their knowledge by the attacker, and thus protect against the related key attacks described in the Appendix and against Wagner's attacks on the CBCM mode [10]. For example, such protection is achieved using the three keys K_1, K_2, K_3 used in the CBCM mode as the key of a keyed hash function with random transmitted "IV," and selecting the output as the actual initial value (IV_1, IV_2).

Recently Wagner devised known-IV and chosen-IV attacks [23] against the modes of operation described in [5]. These attacks work only if the attacker knows the initial values, or can affect them in a specified way, and thus they are also thwarted by choosing the initial values as the result of a keyed-one-way-function of the transmitted "IV." As this protecting technique is very simple and very effective against known-IV and chosen-IV attacks, which usually have a relatively small complexity, we recommend using it in all the modes of operation. On the other hand, this technique does not protect against all kinds of attacks. In particular, the attack in this paper and the attacks described in [6] and [7] are not affected, as they require only one stream of ciphertext, and do not assume any special property of the initial values.

5. Conclusion

In this paper we presented a chosen-ciphertext attack on the CBCM mode of operation, which was close to be included in an ANSI. The attack presented is highly theoretical but shows the difficulty in building practical modes of operation based on 64-bit block ciphers with a high level of security.

Acknowledgments

We are grateful to Nathan Keller for helping solving some of the mathematical equations and to the anonymous referees for valuable comments.

Appendix. A Related-Key Attack

In this appendix we show that the CBCM mode is vulnerable to a *known-IV related-key attack*, where an attacker is able to get encryptions under several different unknown

but related keys. This attack signficantly reduces the required number of plaintexts at the expense of a stronger requirement on the keys. Our attack finds the value of the key K_3 using 2^{33} blocks and has 2^{55} complexity of analysis. One design principle of the CBCM mode is that it should not be possible for an attacker to control the M_i's, in other words, it should be impossible to find the value of K_3. In the following, let $E_{K_1,K_2,K_3}(IV_1, IV_2, P_1, \ldots, P_n)$ denote the encryption of plaintext blocks $P_1, ..., P_n$ in the CBCM mode and let $C_1, ..., C_n$ denote the corresponding ciphertext blocks. Decryption $D_{K_1,K_2,K_3}(\cdot)$ is defined correspondingly.

The first variant of our related key attack is a *known plaintext attack* and can be described as follows: the attacker gets the encryptions of about 2^{33} messages, each consisting of a single, fixed plaintext block P for 2^{33} different values of the key K_3. That is, the attacker obtains

$$C(i) = E_{K_1,K_2,K_3 \oplus a(i)}(IV_1(i), IV_2, P)$$

for $i = 1, \ldots, 2^{33}$, where the $a(i)$'s are known to the attacker and where $a(i) \neq a(j)$ for $i \neq j$. The values of $IV_1(i)$ can be arbitrary and may vary, as long as they are known to the attacker. With a high probability the attacker finds $C(i) = C(j)$ for some $i \neq j$, and expects that this comes from coinciding masking values used in the encryptions, i.e., that $M_1(i) = M_1(j)$, where $M_1(i) = E_{K_3 \oplus a(i)}(IV_1(i))$ and $M_1(j) = E_{K_3 \oplus a(j)}(IV_1(j))$. This can be confirmed by checking if equal ciphertexts are obtained for the two keys with the same initial values for a plaintext $P' \neq P$. The attacker then searches exhaustively for K_3, that is, he solves the equation

$$E_{K_3 \oplus a(i)}(IV_1(i)) = E_{K_3 \oplus a(j)}(IV_1(j))$$

for K_3, which can be done in at most 2^{57} Single-DES encryptions, which are equivalent to encryption of 2^{55} blocks in the CBCM mode. The attack is independent of the values of $IV_1(i)$, as long as they are known. Therefore, restricting the possible values of IV_1 does not help to avoid the attack. Note that in [10] and [9] it is suggested that the initial values are sent in the clear and thus known to an attacker.

The second variant of the attack is a *chosen ciphertext attack*, which works even if the values of IV_2 vary and are unknown. The attacker now gets $P_1(i)$ and $P_2(i)$ from $D_{K_1,K_2,K_3 \oplus a(i)}(IV_1(i), IV_2(i), C_1, C_2)$ for $i = 1, \ldots, 2^{33}$, that is, the decryptions of a ciphertext consisting of two fixed blocks C_1 and C_2. Now he looks for a match $P_2(i) = P_2(j)$ and proceeds similarly to before.

The situation is simpler after K_3 is found. It is clear that the resulting (double encryption) scheme is not stronger than a two-key triple encryption scheme with respect to key-recovery attacks. In fact it is possible to find the values of K_1 and K_2 using less data than the best known attack on two-key triple encryption [22]. Use K_3 to compute M_i, $1 \leq i \leq 2s$, such that for almost every 64-bit value U there exist M_i and M_j, where $i \leq s$ and $s < j \leq 2s$, such that $U = M_i \oplus M_j$. Store each such distinct value of U in a table together with the indices i, j.

With $s \simeq \sqrt{2} \cdot 2^{35}$ it is possible to construct 2^{70} such values of U, which result in almost all possible 64-bit values. The probability that one particular value is not reached is approximately $(1/e)^{64} \approx 2^{-92}$.

Subsequently, choose an additional chosen ciphertext consisting of s identical blocks C_1 followed by s identical blocks C_2. For all values k of K_1 compute $U = D_k(C_1) \oplus$

$D_k(C_2)$. Since K_3 is assumed to be known at this stage it is possible for the attacker to find a pair M_i and M_j such that $U = M_i \oplus M_j$, where M_i was used to decrypt C_1 and M_j was used to decrypt C_2. Let P_i and P_j denote the two corresponding plaintext blocks. If $E_k(P_i \oplus C_1) \oplus E_k(P_j \oplus C_2) = U$, k is a possible value of K_1. The attack is repeated a few times to get a unique value of K_1. Once K_1 is found, it is straightforward to find K_2 by an exhaustive key search. Although this part of the attack requires the memory to store up to 2^{64} values, the chosen text requirements are small, and, in particular, crucially smaller than similar attacks on the TCBC mode of operation.

References

[1] ANSI X9.52-1998. *Triple Data Encryption Algorithm Modes of Operation.*

[2] ANSI draft X9.52. *Triple Data Encryption Algorithm Modes of Operation*, Revision 6.0, May 1996.

[3] E. Biham. On modes of operation (Abstract). In R. Anderson, editor, *Fast Software Encryption - Proc. Cambridge Security Workshop, Cambridge, U.K.*, LNCS 809, pages 116–120. Springer-Verlag, Berlin, 1994.

[4] E. Biham. How to Forge DES-Encrypted Messages in 2^{28} Steps. Technical Report CS884, Technion, August 1996.

[5] E. Biham. Cryptanalysis of Triple Modes of Operation. Technical Report CS885, Technion, August 1996. A preliminary version of [7].

[6] E. Biham. Cryptanalysis of multiple modes of operation. *Journal of Cryptology*, 11(1):45–58, 1998.

[7] E. Biham. Cryptanalysis of triple modes of operation. *Journal of Cryptology*, 12(3):161–184, 1999.

[8] E. Biham and L. R. Knudsen. Cryptanalysis of the ANSI X9.52 CBCM mode. In K. Nyberg, editor, *Advances in Cryptology - EUROCRYPT '98*, LNCS 1403, pages 100–111. Springer-Verlag, Berlin, 1998.

[9] D. Coppersmith, D. B. Johnson, and S. M. Matyas. A proposed mode for Triple-DES encryption. *IBM Journal of Research and Development*, 40(2):253–261, March 1996.

[10] D. Coppersmith, D. B. Johnson, and S. M. Matyas. Triple DES Cipher Block Chaining with Output Feedback Masking. Technical Report RC 20591, IBM, October 1996. Presented at the rump session of CRYPTO '96.

[11] D. W. Davies and G. I. P. Parkin. The average cycle size of the key stream in output feedback encipherment (Abstract). In D. Chaum, R. L. Rivest, and A. T. Sherman, editors, *Advances in Cryptology: Proceedings of Crypto '82*, pages 97–98. Plenum, New York, 1982.

[12] The DESCHALL home page. `http://www.frii.com/~rcv/deschall.htm`.

[13] The distributed.net home page. `http://www.distributed.net/`.

[14] Electronic Frontier Foundation. *Cracking DES—Secrets of Encryption Research, Wiretap Politics & Chip Design*. O'Reilly, Cambridge, MA, 1998. ISBN 1-56592-520-3.

[15] B. S. Kaliski and M. J. B. Robshaw. Multiple encryption: weighing security and performance. *Dr. Dobb's Journal*, pages 123–127, January 1996.

[16] L. R. Knudsen. Block Ciphers – Analysis, Design and Applications. Ph.D. thesis, Aarhus University, 1994.

[17] S. Lucks. Attacking triple encryption. In S. Vaudenay, editor, *Fast Software Encryption, Fifth International Workshop, FSE '98, Paris, France, March* 1998, LNCS 1372, pages 239–253. Springer-Verlag, Berlin, 1998.

[18] J. Markoff. U.S. group delays encryption standard. *New York Times*, March 31, 1998.

[19] U. M. Maurer. New approaches to the design of self-synchronizing stream ciphers. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91*, LNCS 547, pages 458–471. Springer-Verlag, Berlin, 1991.

[20] R. Merkle and M. Hellman. On the security of multiple encryption. *Communications of the ACM*, 24(7):465–467, 1981.

[21] National Bureau of Standards. *Data Encryption Standard.* Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington DC, January 1977.

[22] P. C. van Oorschot and M. J. Wiener. A known-plaintext attack on two-key triple encryption. In Ivan B. Damgård, editor, *Advances in Cryptology - EUROCRYPT* '90, LNCS 473, pages 318–325. Springer-Verlag, Berlin1990.

[23] D. Wagner. Cryptanalysis of some multiple modes of operation. In S. Vaudenay, editor, *Fast Software Encryption, Fifth International Workshop, FSE* '98, *Paris, France, March* 1998, LNCS 1372, pages 254–269. Springer-Verlag, Berlin, 1998.

[24] M. J. Wiener. Efficient DES Key Search. Technical Report TR-244, School of Computer Science, Carleton University, Ottawa, May 1994. Presented at the Rump Session of CRYPTO '93.

[25] M. J. Wiener. Efficient DES key search - an update. *CryptoBytes*, 3(2):6–8, 1998.