RC 20591 (10/21/96)
Mathematics

IBM Research Report

Triple DES Cipher Block Chaining with Output Feedback Masking

D. Coppersmith

IBM Research Division
T.J. Watson Research Center
Yorktown Heights, New York

D. B. Johnson

Certicom
200 Matheson Blvd W., Suite 103
Mississauga, Ontario, Canada L5R 3L7

S. M. Matyas

IBM Cryptography Center of Competence
522 South Road
Poughkeepsie, NY 12601

IBM Research Division
Almaden · T.J. Watson · Tokyo · Zurich

Triple DES Cipher Block Chaining with Output Feedback Masking

Don Coppersmith
IBM Research
T. J. Watson Research Center
Yorktown Heights, NY 10598

Don B. Johnson
Certicom
200 Matheson Blvd W.,
Suite 103
Mississauga, Ontario
Canada L5R 3L7

Stephen M. Matyas
IBM Cryptography Center
of Competence
522 South Road
Poughkeepsie, NY 12601

Abstract

We propose a new mode of multiple encryption, namely "Triple DES cipher block chaining with output feedback masking." The aim is to provide strong protection against certain attacks ("dictionary attacks" and "matching ciphertext attacks") which exploit the DES blocksize of 64 bits. The new mode obtains this protection through the introduction of secret masking values that are Exclusive-ORed with the intermediate outputs of each triple-DES encryption operation. The secret mask value is derived from a fourth encryption operation per message block, in addition to the three used in previous modes. The new mode is part of a suite of encryption modes proposed in the ANSI X9.F.1 Triple-DES draft standard (X9.52).

1 Introduction

Much effort has gone into attempted cryptanalysis of multiple DES [2, 3, 4, 6, 8, 10, 11, 14, 17, 18, 21]. Because the DES is still a fundamentally sound base to build on, the American National Standards Institute (ANSI) committee X9.F.1 is working to standardize a suite of modes of triple-DES encryption (X9.52). [9]

One proposal has been to use triple DES in cipher block chaining mode with external feedback. This provides a longer effective key length for DES. Unfortunately, it retains the 64-bit message block size. This renders the mode vulnerable to matching text attacks: after 2^{32} blocks have been enciphered, some blocks can be expected to repeat, and this leaks information.

On the other hand, Biham [2, 3] has shown that many internal feedback mechanisms are subject to probing by the cryptanalyst. For example if the feedback comes from ciphertext, the cryptanalyst might mount a chosen-ciphertext attack, manipulating the ciphertext to plant values at internal points, and mount a differential cryptanalysis attack from there.

The present paper proposes a new mode (CBCM), designed to increase the strength of triple DES against the threat of dictionary and matching ciphertext attacks, without needing to change the 64-bit blocksize of the DES algorithm. The method uses secret masking values calculated via a parallel running Output Feedback (OFB) mode, to destroy any relationship between matching ciphertext blocks. OFB is one of several modes of encryption that can be performed with the DES (see Meyer-Matyas [15] and the definitions by standards bodies [1, 12, 13]). The masking values are Exclusive-ORed ($\oplus$) with the intermediate outputs of each single DES operation. The cost for this increased security is one additional OFB encryption (giving a total of four encryptions per block), along with two additional Exclusive-OR operations which are interleaved between the triple-DES encryptions and decryptions (in the order Encrypt-Decrypt-Encrypt or *e*-*d*-*e*).

We believe that the proposed mode is at least as strong as two-key triple DES with regard to key discovery attacks; so far, all known key discovery attacks support this belief, although it is possible that the added Exclusive-OR steps could introduce new attacks. In addition, it avoids some of the weaknesses of previous external or internal feedback triple DES modes. The Exclusive-ORed masking values offer protection from the matching ciphertext attacks against external feedback modes; and the fact that the masking values are generated by a free-running OFB stream, rather than by manipulable ciphertext, protects this mode against Biham's attacks on internal feedback modes.

A related paper [7] gives more information on comparable modes.

The paper is organized as follows. In Section 2 we present, for comparison, the Triple DES in cipher block chaining mode with external feedback, and demonstrate the matching text attacks against it. We also mention the problems with internal feedback. In Section 3 we define our new proposed mode and give some design objectives. Section 4 analyzes the strength of the new mode and outlines the effectiveness of various attacks. Section 5 contains some justification for various design choices made along the way, suggested variants, and properties of the proposed system.

2 Triple-DES Cipher Block Chaining with External Feedback

Our proposed mode builds on the base of another Triple-DES mode, namely Triple DES Cipher Block Chaining or TCBC, which is part of a suite of encryption modes in the ANSI X9.F.1 Triple-DES draft standard (X9.52). We begin by presenting the TCBC mode, for comparison and as a point of departure.

The TCBC mode uses a non-secret 64-bit Initializing Vector (IV), and three secret 64-bit keys (K_1, K_2, and K_3) to encrypt a plaintext $(X_1, X_2, ..., X_n)$ consisting of n 64-bit blocks and produce a ciphertext $(Y_1, Y_2, ..., Y_n)$ consisting of n 64-bit blocks; see Figure 1. At the first iteration of encryption, the IV is Exclusive-ORed with the first block of plaintext, X_1. At each subsequent iteration of encryption, each block of plaintext X_i is Exclusive-ORed with the previous block of ciphertext Y_{i-1}. The result is then encrypted with three-key triple DES (e-d-e) – encrypted under K_1, decrypted under K_2, and encrypted under K_3 – to produce the ciphertext: $Y_i = e(K_3, d(K_2, e(K_1, X_i \oplus Y_{i-1})))$, with Y_0 being interpreted as IV.

A variant of TCBC uses only two independent keys, setting $K_3 = K_1$. We will refer to this variant as two-key TCBC.

The TCBC mode has the following advantageous attributes:

1. The input and output blocksize is 64 bits, the same as normal DES.

2. It is backwards compatible with respect to single-key DES encryption.

    - Using one key value for all three key inputs results in the same output as a single DES encryption.

3. It has limited error propagation.

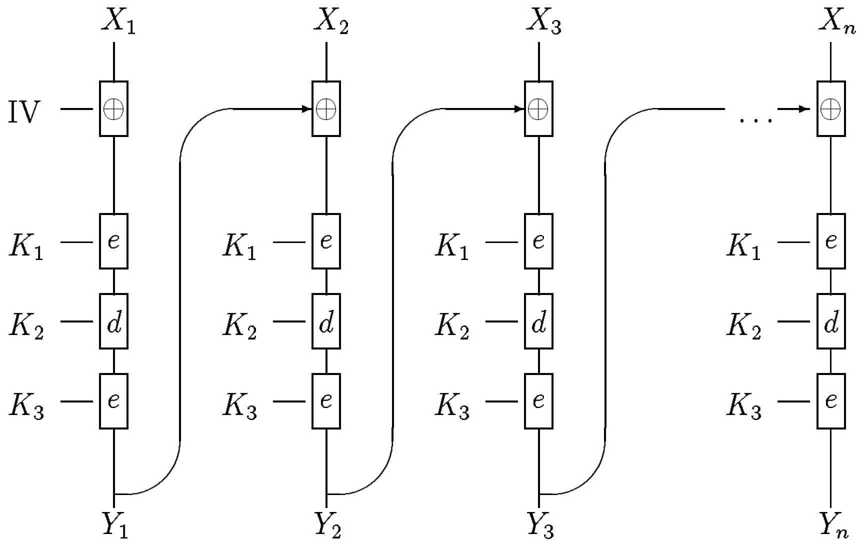

Figure 1: Triple-DES external feedback cipher block chaining (TCBC)

- If one block of ciphertext is corrupted, then only two blocks of recovered plaintext will be corrupted. This is known as the self-healing or self-synchronizing property of CBC encryption.

4. It is resistant to cryptanalytic exhaustive key search attacks.

   - Using two keys ($K_3 = K_1$), if n is the number of known plaintext blocks, then the best known work factor [18] is $\min(2^{120}/n, 2^{112})$.
   - Using three independent keys ($K_3 \neq K_1$), the best known work factor is 2^{112} with some known plaintext; having many known plaintext blocks does not appear to reduce this work factor.

2.1 Matching ciphertext attack

However, the TCBC mode, and indeed any multiple-key CBC mode of operation with external feedback, inherits the 64-bit message-block size from DES. This leads to a concern with text dictionary or matching ciphertext. We sketch the attack here.

Because there are only 2^{64} possible 64-bit message blocks, the birthday phenomenon predicts that after about $2^{32} = \sqrt{2^{64}}$ message blocks have been enciphered, with high probability two of the ciphertexts will agree: $Y_i = Y_j$. [1] When this happens, we see that the outputs of the "pipe" $e(K_3, d(K_2, e(K_1, *)))$ agree, implying that the inputs to this "pipe" also agree:

$$X_i \oplus Y_{i-1} = X_j \oplus Y_{j-1}.$$

A similar result holds no matter how long the collection of encryptions in between – in this case, $e(K_3, d(K_2, e(K_1, *)))$ – as long as there is no inner feedback. The cryptanalyst detects this case because he sees the ciphertexts Y_i and Y_j.

If $Y_i = Y_j$ and the two corresponding plaintexts X_i and X_j are unknown, then the cryptanalyst will have discovered a relation of the form

$$\begin{aligned} Y_{i-1} \oplus X_i = d(K_1, e(K_2, d(K_3, Y_i))) &= d(K_1, e(K_2, d(K_3, Y_j))) = Y_{j-1} \oplus X_j \\ X_i \oplus X_j = Y_{i-1} \oplus Y_{j-1}, & \end{aligned}$$

a possibly useful relation between plaintext messages. This possibility will be called "matching ciphertext".

Similarly, if $Y_i = Y_j$ and X_i is known but X_j is unknown, the cryptanalyst will have discovered the value of X_j. This possibility is essentially the same as a "text dictionary" attack, where the attacker builds a dictionary of known input-output pairs for the $e(K_3, d(K_2, e(K_1, *)))$ sequence of encryptions.

Incidentally, the DESX scheme [16] $Y_i = K_3 \oplus e(K_2, K_1 \oplus X_i)$ falls prey to a similar attack.

Much of the work in multiple DES has concentrated on the expansion of the effective key length, while ignoring the equally important problem of relatively small block size, and accompanying problems with matching ciphertext or dictionary attacks. A primary goal of the present work is to remedy this situation.

2.2 Inner feedback triple DES

Because of the matching ciphertext attacks against external feedback CBC mode, one might consider *internal* feedback: for example, feeding the previous ciphertext in between two encryption stages with an Exclusive-OR operation. But Biham [2, 3] has given convincing evidence that such schemes are not secure against certain key

[1] However, note that chaining is much better than no chaining. Without chaining, the number of possible plaintext blocks is much smaller, so that collisions occur earlier. In effect, doing chaining with a pseudorandom value ensures that a complete dictionary must correspond to the blocksize, which is the best that can be achieved.

discovery attacks. The introduction of manipulable data (such as previous ciphertext) into the middle of the sequence of encryption stages, allows the attacker to probe inside and mount differential attacks.

Our present scheme (CBCM) will also introduce masking data into the middle of the sequence of encryption stages, but this data will be generated by a free-standing output-feedback (OFB) stream. The masking data is not under the control of the attacker, so that Biham's attacks are avoided.

3 Triple-DES CBC with OFB Masking (CBCM) Mode

The newly proposed mode uses Triple-DES CBC and OFB Masking. We abbreviate the mode as CBCM. It begins with the Triple-DES Cipher Block Chaining mode (TCBC). It introduces a secret masking value (see Figure 2), which is calculated via a parallel-running OFB mode and is Exclusive-ORed at each iteration with the intermediate outputs of the CBC mode. The CBCM mode is a unique design with the characteristic that it cannot be simulated using a combination of other modes.

The process uses two publicly known 64-bit initializing vectors, IV1 and IV2. IV1, used in the OFB portion, is selected randomly from a collection of 2^{20} possible values, say those with high order 44 bits set to 0. IV2, used in the CBC portion, is a fixed value, say 0.

We present here the design objectives for the new scheme:

1. CBCM should be at least as strong as two-key TCBC mode with regard to key discovery attacks.

2. CBCM should be stronger than TCBC mode with regard to text dictionary and matching ciphertext attacks.

3. It should mask patterns in the input plaintext.

4. The input and output blocksize is 64 bits; this is an ANSI X9.F.1 requirement.

5. Its design should be understandable, to facilitate analysis.

Note: it was not a design objective to be backwards compatible with an existing mode of operation.

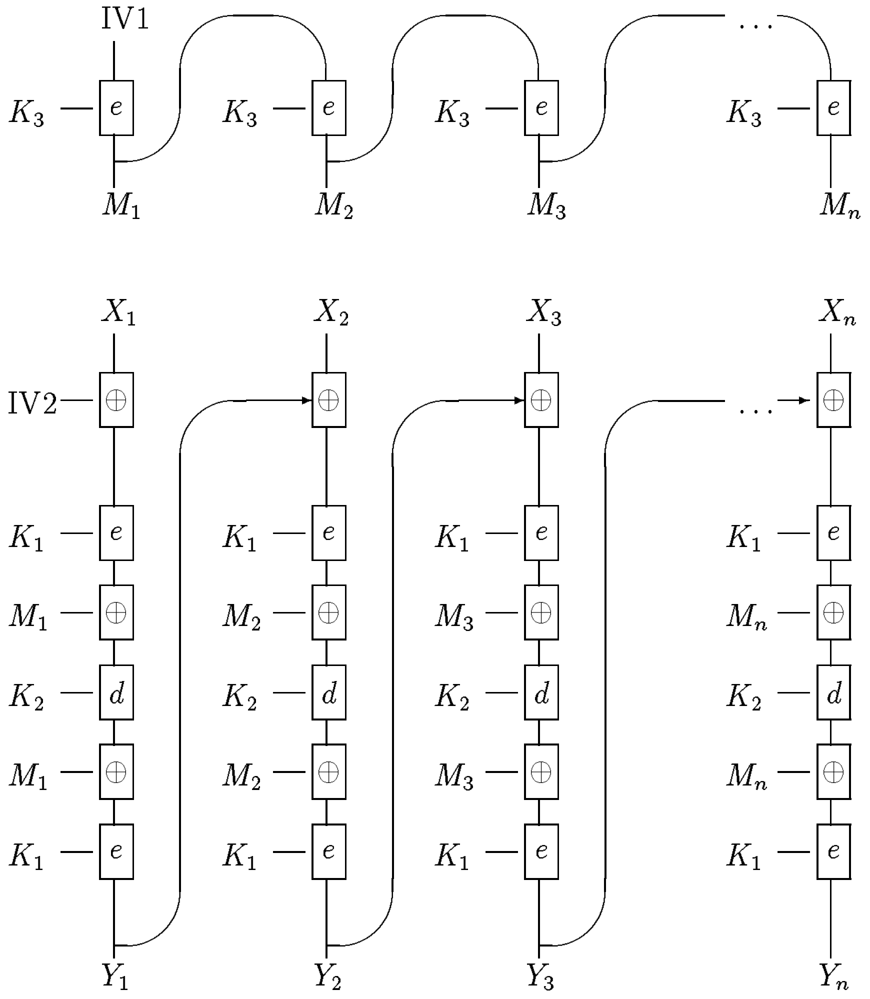

Figure 2: Triple-DES external feedback CBC with OFB masking (CBCM) mode

4 Strength of Triple-DES CBC with OFB Masking (CBCM) Mode

With regard to a matching cipher text or dictionary attack, the adversary is forced to launch a separate attack for each IV1 and each separate iteration of CBC encryption. We assume that an adversary cannot use the keys (K_1, K_2, and K_3) in other cryptographic operations that might allow the keys to be attacked with less work than in the CBCM mode. For example, we assume that an adversary has no means to cause K_3 to be used in an ordinary OFB mode such that its outputs would be exposed, since this would allow the M_1, M_2, etc. values to be exposed for a particular IV1 and K_3.

With regard to a cryptanalytic key discovery attack, CBCM mode appears to be about as strong as two-key TCBC mode. Of the attacks which we investigated, one of the most promising (described below) appears to require 2^{90} DES operations, as well as 2^{34} blocks of chosen ciphertext and corresponding plaintext. (The chosen ciphertext requirement is unusual, and arises because of the outer chaining.) Another requires 2^{80} DES operations and 2^{44} blocks of chosen ciphertext. Without large amounts of known plaintext and ciphertext (2^{34} blocks), the known attacks require 2^{112} or more operations.

Attack 1. *Requirements: 2^{34} chosen ciphertext; 2^{90} DES encryptions.*

In one of the most promising attacks, the opponent creates two long ciphertexts. The first consists of 2^{33} repetitions of some constant 64-bit block Y, so $Y_i = Y$. The second contains 2^{33} repetitions of some other constant 64-bit block Y', so $Y'_i = Y'$. He then requests the decryption of both, using the same unknown keys K_1, K_2, K_3, and the same initializing vector IV1 for both ciphertexts.

Because the masking values M_i depend only on K_3, IV1, and the index i, and not on the plaintext or ciphertext, one sees that same values of M_i will be used for both encryptions.

The opponent is looking for a pair of indices (i, j) where $M_i \oplus d(K_1, Y_i) = M_j \oplus d(K_1, Y'_j)$. Call this a "coincidence". These two quantities represent the outputs of the decryption stage $d(K_2, *)$ at different instances; because they agree, the corresponding inputs agree, namely $M_i \oplus e(K_1, X_i \oplus Y_{i-1}) = M_j \oplus e(K_1, X'_j \oplus Y'_{j-1})$. Combining these two equations, he would have that $d(K_1, Y_i) \oplus e(K_1, X_i \oplus Y_{i-1}) = d(K_1, Y'_j) \oplus e(K_1, X'_j \oplus Y'_{j-1})$.

Because of the choice of ciphertext ($Y_i = Y_j$ and $Y'_i = Y'_j$), one coincidence leads to another, namely $M_j \oplus d(K_1, Y_j) = M_i \oplus d(K_1, Y'_i)$.

The attacker selects a trial value k_1 for the key K_1. He then evaluates the quantities $d(k_1, Y_i) \oplus e(k_1, X_i \oplus Y_{i-1})$ and $d(k_1, Y_j') \oplus e(k_1, X_j' \oplus Y_{j-1}')$ for indices i, j up to 2^{33}, and compares the lists, looking for matches: $d(k_1, Y_i) \oplus e(k_1, X_i \oplus Y_{i-1}) = d(k_1, Y_j') \oplus e(k_1, X_j' \oplus Y_{j-1}')$. In case of a match, he also checks whether the second match (predicted by the second coincidence) holds: namely, whether $d(k_1, Y_j) \oplus e(k_1, X_j \oplus Y_{j-1}) = d(k_1, Y_i') \oplus e(k_1, X_i' \oplus Y_{i-1}')$. When both hold, one calls it a "double match". One expects that for the correct key $k_1 = K_1$, there will be one or two such double matches, while probably none will appear for incorrect keys. So this procedure serves to find K_1. Its cost is $2^{56} \times 2 \times 2^{33} = 2^{90}$ decipherments.

Having found K_1, it is easier to find K_3 and K_2 in the reduced system. (We are grateful to Bart Preneel for suggesting the following improvement to our original attack.) Given $2^{16} = 65,536$ ciphertext strings, each containing $2^{17} = 131,072$ identical blocks $Y_{a,b} = y_a, a \leq 2^{16}, b \leq 2^{17}$, and identical, known, IV1. He looks for double matches as before: $d(K_1, Y_{a,b}) \oplus e(K_1, X_{a,b} \oplus Y_{a,b-1}) = d(K_1, Y_{c,d}) \oplus e(K_1, X_{c,d} \oplus Y_{c,d-1})$ and $d(K_1, Y_{a,d}) \oplus e(K_1, X_{a,d} \oplus Y_{a,d-1}) = d(K_1, Y_{c,b}) \oplus e(K_1, X_{c,b} \oplus Y_{c,b-1})$. Find $M_b \oplus M_d = d(K_1, Y_{a,b}) \oplus d(K_1, Y_{c,d})$ with $b, d \leq 2^{17}$. (He expects to find one or two such double matches, because of the birthday phenomenon.) Now for each of 2^{56} trial values k_3 for K_3, and using the known initializing vector IV1, he runs through 2^{17} steps of output feedback mode to discover the values $M_1, M_2, ..., M_b, ..., M_d$, corresponding to this guessed value of k_3. If the guess is correct, the calculated values M_b and M_d will satisfy the equation, while there will probably be no false alarms from incorrect guesses. Having thus found K_3, at a cost of $2^{56} \times 2^{17} = 2^{73}$ encipherments, he can find K_2 by key exhaustion, at a cost of 2^{56} encipherments.

The requirements of 2^{34} blocks of chosen ciphertext and $2^{90} + 2^{17} + 2^{56} \approx 2^{90}$ encryptions render this attack infeasible for many years to come.

Remark. A variant of this attack still works when two independent keys K_1, K_4 are used in place of the two instances of K_1. Assume there are two "double matches", with indices (i, j) and (k, ℓ). For each match, $d(K_4, Y) \oplus d(K_4, Y') = M_i \oplus M_j = e(K_1, X_i \oplus Y_{i-1}) \oplus e(K_1, X_j' \oplus Y_{j-1}') = e(K_1, X_j \oplus Y_{j-1}) \oplus e(K_1, X_i' \oplus Y_{i-1}')$, so that $e(K_1, X_i \oplus Y_{i-1}) \oplus e(K_1, X_i' \oplus Y_{i-1}') = e(K_1, X_j \oplus Y_{j-1}) \oplus e(K_1, X_j' \oplus Y_{j-1}')$. For each trial k_1 we would enumerate $f(i) = e(k_1, X_i \oplus Y_{i-1}) \oplus e(k_1, X_i' \oplus Y_{i-1}')$ for all i and look for matches: $f(i) = f(j)$ and $f(k) = f(\ell)$. For the correct value $k_1 = K_1$ we will also have $e(K_1, X_i \oplus Y_{i-1}) \oplus e(K_1, X_j' \oplus Y_{j-1}') = d(K_4, Y) \oplus d(K_4, Y') = e(K_1, X_k \oplus Y_{k-1}) \oplus e(K_1, X_\ell' \oplus Y_{\ell-1}')$. This serves to identify K_1, and exhaustion identifies K_4.

Attack 2. *Requirements: 2^{34} known plaintext and ciphertext; 2^{90} DES encryptions; 2^{112} elementary operations.*

A related attack uses 2^{34} known plaintext and corresponding ciphertext (instead of chosen ciphertext), and 2^{90} encryptions, but seems to also require 2^{112} elementary operations such as Exclusive-OR ($\oplus$). Considering a DES encipherment to be as complex as $256 = 2^8$ elementary operations, this is the equivalent of 2^{104} encipherments. The attacker is assumed to know IV1. For each trial value k_1 for K_1, he evaluates the quantity $e(k_1, Y_{i-1} \oplus X_i) \oplus d(k_1, Y_i)$ for each index $i \leq 2^{34}$, and look for a few matches: $e(k_1, Y_{i-1} \oplus X_i) \oplus d(k_1, Y_i) = e(k_1, Y_{j-1} \oplus X_j) \oplus d(k_1, Y_j)$. He hopes that these matches arise because $d(k_1, Y_i) \oplus M_i = d(k_1, Y_j) \oplus M_j$. With each trial key k_1 he lists some triples $(i, j, M_i \oplus M_j)$ obtained by this process. Now for each trial value k_3 for K_3, he evaluates the various M_i from the known IV1, and sees whether any triples on the list are satisfied. He needs to do about 2^{90} encipherments altogether, but checking the lists appears to require 2^{112} elementary operations.

Attack 3. *Requirements: T blocks of known plaintext and ciphertext;* 2^{112} *elementary operations;* $\max(2^{88}, 2^{112}/T)$ *DES encryptions;* $(2^{120}/T)$ *24-byte blocks of storage (not a firm requirement).*

For each trial value k_2 for K_2 and each of $2^{64}/T$ possible intermediate message blocks Z (say those blocks Z whose low order $\log_2 T$ bits are 0), the attacker precomputes the triple $(k_2, Z, d(k_2, Z))$, and places these in an (immense) table.

He hopes that among the T indices i, there will be an index i for the input to the $d(k_2, *)$ decryption stage, namely $M_i \oplus e(K_1, X_i \oplus Y_{i-1})$, will match one of the Z; on probabilistic grounds this should happen once.

For each of T indices i, and for each of 2^{56} possible values k_3 for K_3, he evaluates the value M_i corresponding to k_3; we write this as $M_i(k_3)$ to emphasize its dependence on k_3. For each i and each possible k_1, he evaluates $e(k_1, X_i \oplus Y_{i-1})$. He selects those triples (i, k_3, k_1) for which the low order $(\log_2 T)$ bits of $M_i(k_3)$ agree with those of $e(k_1, X_i \oplus Y_{i-1})$, so that their Exclusive-OR will be a Z. There are 2^{112} such triples. For each such triple, he evaluates $M_i(k_3) \oplus d(k_1, Y_i)$, which should be equal to $d(k_2, Z)$ where $Z = M_i(k_3) \oplus e(k_1, X_i \oplus Y_{i-1})$ if all the k_i are right. From the immense table, he retrieves those values of k_2 (if any) satisfying $(Z, d(k_2, Z)) = (M_i(k_3) \oplus e(k_1, X_i \oplus Y_{i-1}), M_i(k_3) \oplus d(k_1, Y_i))$. The expected number of such k_2 is $2^{56}/2^{64} = 2^{-8}$. So the expected number of "false alarm" triples (k_1, k_2, k_3) is $2^{112-8} = 2^{104}$. He can try each by exhaustion to find the correct triple, or can repeat the experiment to see whether any triple occurs several times.

This attack, as described, has incredibly large storage requirements. We have not paid much attention to storage cost, and techniques such as those of van Oorschot and Wiener [19] might well reduce it.

Attack 4. *Requirements: 170 blocks of known plaintext and ciphertext;* 2^{120} *DES* *encryptions;* 2^{64} *bits of storage.*

This attack makes use of the fact that, for a fixed key k_2, as Z ranges over all possible 64-bit message blocks, the function $f(Z) = Z \oplus d(k_2, Z)$ ranges over only a fraction $63\% = 1 - (1/e)$ of message space.

The attack has an outer loop over possible values k_2 for the unknown key K_2. For each such value, the attacker runs through all 64-bit message blocks Z and tabulate which 64-bit blocks W can be achieved as $W = f(Z) = Z \oplus d(k_2, Z)$. He sets up a 2^{64}-bit vector indicating which values of W are possible.

Still within the outer loop (maintaining the same guessed value of k_2), he runs through all possible values k_1 for the unknown key K_1. Within this inner loop, he selects a message index i and evaluates the quantity $W = W(k_1, k_2, i) = e(k_1, X_i \oplus Y_{i-1}) \oplus d(k_1, Y_i)$. He checks the bit table to see whether W has been achieved as $f(Z)$ for some block Z. (He does not need to see Z itself.) If the guesses k_1 and k_2 are correct, then he will have indeed $W = f[e(k_1, X_i \oplus Y_{i-1}) \oplus M_i]$, as can be verified:

$$\begin{array}{rcccc} W & = & e(k_1, X_i \oplus Y_{i-1}) & \oplus & d(k_1, Y_i) \\ & = & [e(k_1, X_i \oplus Y_{i-1}) \oplus M_i] & \oplus & [M_i \oplus d(k_1, Y_i)] \end{array}$$

If the guesses are incorrect, then with probability 37% W will fail to be listed in the table, and he can eliminate this guess. A corpus of 170 pairs (X_i, Y_i) will probably eliminate all false guesses, since $2^{112} \times (1 - 1/e)^{170} \approx 1$. The average number of trials per false guess is only $1 + 0.63 + 0.63^2 + 0.63^3 + \ldots = 2.718$.

Once all possible values k_1 have been ruled out for a given k_2, he cycles to the next k_2, until finding the correct value.

Notice that this attack presumed no knowledge of IV1 or K_3, and in fact treated the M_i as independent unknown quantities.

Attack 5. *Matching ciphertext. Requirements:* $2^{42}/\sqrt{L}$ *messages, each of length* $L \times 8$ *bytes, totalling* $2^{45}\sqrt{L}$ *bytes. Yields information on one pair of message blocks.*

This attack demonstrates the strength against matching text attacks. Suppose $N = 2^{42}/\sqrt{L}$ messages X^h have been transmitted, each consisting of $L \times 8$ bytes, each having an IV1 chosen independently from the 2^{20} possible values. For message X^h, let X_i^h and Y_i^h denote the i-th block of plaintext and corresponding ciphertext, and let $\mathrm{IV1}^h$ denote the selected value of IV1. The mask M_i^h depends on the block index i and on the initializing vector $\mathrm{IV1}^h$, but does not otherwise depend on the message identifier h. Whenever there are two messages X^h and $X^{h'}$ whose IV1 values agree ($\mathrm{IV1}^h = \mathrm{IV1}^{h'}$), for each position i within the message the "pipe" $e(K_3, M_i^h \oplus d(K_2, M_i^h \oplus e(K_1, *)))$ has the same input/output relations for the two messages X^h and $X^{h'}$.

With N and L related as above ($N^2L \approx 2^{64} \times 2^{20}$), there is a good probability of finding indices h, h', i with $\mathrm{IV1}^h = \mathrm{IV1}^{h'}$ and with $Y_i^h = Y_i^{h'}$; notice that the i indices must agree. When this happens the analyst can calculate $X_i^h \oplus X_i^{h'} = Y_{i-1}^h \oplus Y_{i-1}^{h'}$ as in the matching ciphertext attack.

The requirements here are far more severe, though: instead of 2^{42} message blocks of 8 bytes each, we need to see $2^{42}/\sqrt{L}$ message strings of $8L$ bytes each, totalling $2^{45}\sqrt{L}$ bytes. For example, if a typical message has length $2048 = 8 \times 256$ bytes, then $L = 256$, and we need to see $2^{42}/\sqrt{2^8} = 2^{38}$ such messages, for a total of $2^{38} \times 2048 = 2^{49}$ bytes, to get one useful instance of matching ciphertext.

Roughly speaking, the attacker needs to run $2^{20}L = 2^{28}$ different matching ciphertext attacks, one for each combination of position i within the message and initializing vector $\mathrm{IV1}^h$, because the varying M_i^h renders the input-output relations of the pipe $e(K_3, M_i^h \oplus d(K_2, M_i^h \oplus e(K_1, *)))$ different for each different pair $(i, \mathrm{IV1}^h)$.

Attack 6. *Chosen IV1 attack. Requirements:* 2^{44} *chosen ciphertexts,* 2^{80} *encipherments.*

We are grateful to David Wagner [20] for pointing out this attack to us. Its existence forces us to restrict the range of allowable values for IV1. We only sketch the attack here.

The attacker decrypts 2^{20} messages, each consisting of 2^{24} identical 8-byte blocks Y, for a total of 2^{47} bytes. He expects a coincidence between two M values – $M_i^h = M_{i'}^{h'}$ – which he will discover by looking for matching strings of plaintext. For those two IV1 values (h and h'), and for each of 2^{56} possible values k_3 for K_3, he produces the streams M_j^h and $M_j^{h'}$ and checks whether the agreement $M_i^h = M_{i'}^{h'}$ holds. This distinguishes the correct key K_3. Knowing the correct key K_3 and being able to manipulate $IV1$ to some degree, he can then probe inside the pipe and determine K_1 and K_2.

Wagner's attack originally dealt with an earlier proposal, where IV1 was free to range. There the attacker could use shorter messages, needing only 2^{33} blocks of chosen ciphertext, and requiring only 2^{56} encryptions. With the restrictions on IV1, the attacker needs longer messages, and then the trial key exhaustion on k_3 becomes much more expensive, because with each k_3 he needs to run the OFB portion for about 2^{24} steps in order to check the match.

Attack 7. *Linear and Differential Cryptanalysis.*

The number of known plaintext/ciphertext pairs required for linear or differential cryptanalysis is exponential in the number of rounds. Since CBCM has 48 rounds (besides the M_i being added in after rounds 16 and 32), and 16-round single DES requires 2^{43} known texts for linear cryptanalysis and 2^{47} for differential cryptanalysis, a rough estimate of $(2^{43})^3 = 2^{129}$ leads one to believe that linear or differential cryptanalysis will not be effective against this mode.

Summary: The known attacks require at least 2^{90} DES operations and 2^{34} blocks of chosen ciphertext, or in the case of Wagner's attack, 2^{80} encryptions and 2^{44} blocks of chosen ciphertext. The work factor is even larger than 2^{90} unless there is available a large corpus of known plaintext and corresponding ciphertext. We have not proved a lower bound, and further investigation is warranted.

These chosen ciphertext attacks require more resources than even the known plaintext attacks against two-key TCBC. For attacks requiring 2^{90} DES operations, attacks on two-key TCBC need 2^{30} blocks of *known* plaintext and ciphertext, while attacks on CBCM require 2^{34} blocks of *chosen* ciphertext. At 2^{80} DES operations, attacks on two-key TCBC need 2^{40} blocks of known plaintext and ciphertext, and attacks on CBCM require 2^{44} blocks of chosen ciphertext.

5 Comments and rationale

5.1 Remarks on attributes of IV1 or IV2

Several of the attacks depend on characteristics of IV1. Attack 1 is facilitated by the reuse of IV1 for two different messages, while not requiring it to be known, while Attack 2 and Attack 3 are made easier by a known value of IV1.

On the other hand, Attack 6 is made easier by a freely varying IV1, and is made harder by restricting its range.

Notice that even if one demanded that a particular IV1 value never be reused, one might be concerned that an attacker with temporary access to the target machine can force the reuse of the same IV1 on the decryption of several different messages.

The original draft proposal for ANSI [9] did not specify whether IV1 should be public or secret, nor whether it should be fixed or changing. But in light of the various attacks, it appears that a compromise must be made between constant IV1 (Attack 5) and freely varying IV1 (Attack 6). The choice of a set of 2^{20} possible values seems to maintain a high level of security against both classes of attacks. This necessitates a publicly known IV1, to avoid the technical difficulties of transporting a secret, changing IV1 value for each new message.

On the other hand, the second initializing vector IV2 is clearly not critical. It may be publicly known. It may be held constant, say IV2=0, or, if it varies in time, it may be transmitted in the clear. Notice that the role of IV2 is similar to that of a ciphertext Y_0, which is not protected. We recommend a public, constant value of IV2, for simplicity.

5.2 Some Design Rationale

This system was designed to yield high security without too much computational complexity. At this point in the discussion we are able to describe the reasoning behind some of the choices which were made, recognizing that other design choices might also have satisfied the criteria.

- *Why is M_i Exclusive-ORed in twice?*

  If the second instance of M_i were eliminated, then we could mount an attack with a few chosen ciphertexts, two instances of the same IV1, and the cost of single key exhaustion. Namely, design two ciphertexts Y_i and Y_i' with $Y_2 = Y_3$ and $Y_2' = Y_3'$. Obtain the two decipherments X_i and X_i'. Verify that the following equation must hold: $e(K_1, X_2 \oplus Y_1) \oplus e(K_1, X_3 \oplus Y_2) = e(K_1, X_2' \oplus Y_1') \oplus e(K_1, X_3' \oplus Y_2')$. Try all possibilities k_1 for K_1 to find the correct one. Very loosely speaking, the present scheme, by sandwiching two instances of "$M_i \oplus *$" around the decryption step $d(K_2, *)$, inherits some strength from the one-way function $f(y) = y \oplus d(K, y)$.

- *Why do we have the two outer encryptions by K_1? Why not just use $Y_i = M_i \oplus d(K_2, M_i \oplus X_i \oplus Y_{i-1})$?*

  As seen in Attack 1, once K_1 is discovered, the remaining keys can be discovered more easily.

- *What about "weak keys"?*

  We don't feel that this is a major concern. If the keys are generated at random, with probability less than $1/2^{50}$ the user will accidentally get one of the 16 "weak" or "semi-weak" keys [8] as either K_1, K_2, or K_3. In the worst case, a weak key as K_3 would lead to alternating values of M_i, namely $M_i = M_{i+2}$, causing the scheme to behave like two-key TCBC with near-constant masking values in between. The cryptanalyst would apparently need to see 2^{32} ciphertext values before noticing this accident, and even then he would be no better off than against two-key TCBC.

  An extremely cautious implementation can filter out weak keys before use, just as in ordinary DES.

5.3 Variants

We present here several variants, which merit further study. We point out that only the basic scheme was presented to ANSI.

One variant, suggested by an anonymous referee, would replace the stream cipher $M_i = e(K_3, M_{i-1})$ by a counter mode: $M_i = e(K_3, IV1 + i)$. This would facilitate resynchronization, as well as random-access decryption from the middle of a file. This flexibility apparently comes at the cost of some security, however; in several attacks, especially Attack 6, the attacker, interested in specific plaintext locations i and j, could attack these locations directly without following the OFB chain from the beginning. This would reduce the computational requirements of Attack 6 from 2^{80} to about 2^{56} encipherments.

A second variant: Instead of using the same key K_1 twice, an independent key K_4 could be used for the second instance of K_1. This would add to the number of secret keys that have to be transmitted and stored. It might increase the security of the scheme; but by the remark on Attack 1, it will not increase security by much. The same considerations occur in the choice between two-key triple DES and three-key triple DES.

Much more promising is the idea to use two different quantities M_i and M_i' in the two Exclusive-OR steps. Drawbacks are that it would again require an additional secret key K_4 to generate the second stream, and more importantly, it would require an additional encryption for each block, for a total of five encryptions per block rather than the present four. But this variant might considerably enhance security; it seems to block all the attacks listed so far. Further work is required to evaluate the security of this variant.

5.4 Properties of the present system

Finally, we are able to list the properties of the present system, to enable comparison with other systems.

1. The input and output blocksize is 64 bits, the same as normal DES.

2. It is not backwards compatible with respect to single-key DES encryption.

3. It has limited error propagation with regard to corruption, but unlimited with regard to synchronization.

- If one block of ciphertext is corrupted, then only two blocks of recovered plaintext will be corrupted.
- If synchronization is lost (a block is lost), then all succeeding recovered plaintext will be in error.

4. It requires four DES encryptions or decryptions per 64-bit block, as compared to three for TCBC.

5. Pipelining: The OFB process (using K_3) can be run in parallel with the CBC process (using K_1 and K_2), but the CBC process cannot be pipelined except by interleaving the data. (That is, the input to $e(K_1, *)$ would be $X_i \oplus Y_{i-3}$ instead of $X_i \oplus Y_{i-1}$, effectively breaking the message into three interleaved streams, in order to facilitate hardware pipelining.)

6. It uses three secret keys.

7. It cannot be simulated directly using existing DES modes of operation; this is a minor disadvantage with regard to reusing existing hardware.

8. It appears to be secure against chosen plaintext and chosen ciphertext attacks: known attacks require 2^{34} blocks of chosen text and 2^{90} encryptions, or 2^{44} blocks of chosen text and 2^{80} encryptions.

9. It has a complementarity property, inherited from ordinary DES.

   - If IV1, K_3 and K_2 are all complemented, the resulting system is not affected; it still encrypts X_i into Y_i.
   - If IV2, K_1 and K_2 are all complemented, the resulting system encrypts X_i into $C(Y_i)$, the complement of Y_i.

   This is of minor importance.

10. Most importantly, the potential concerns with text dictionary attacks and with matching ciphertext attacks have been addressed.

6 Conclusion and Open Problems

We have presented a new method for multiple DES encryption. Like other triple-DES modes, it uses several independent keys, achieving strength against key discovery attacks; its security in this regard is apparently the same as two-key TCBC. Unlike other modes, it also defends against attacks based on the small block-size, namely dictionary attacks and matching ciphertext attacks. The added cost is acceptable – four encryptions per block instead of three. We therefore recommend that the new mode (CBCM) be adopted by ANSI.

We encourage the reader to explore other attacks against this scheme.

With the introduction of four encipherments per block, a wider selection of modes becomes available. It would be of interest to investigate some of these modes and see whether security could be improved even further, or whether provable security could be attained.

6.1 Acknowledgements

Technical comments and discussions from Chris Holloway, Tom Jones, Bart Preneel, Ron M. Smith, Jr., Michael Steiner, and David Wagner are gratefully acknowledged.

References

[1] ANSI X3.106, "Data Encryption Algorithm - Modes of Operation," ANSI, May 16 1983.

[2] Eli Biham, "On Modes of Operation," *Fast Software Encryption*, Cambridge Security Workshop Proceedings, Cambridge, U.K., 116-120, Springer-Verlag, Berlin, 1994.

[3] Eli Biham, "Cryptanalysis of Multiple Modes of Operation," *Advances in Cryptology - ASIACRYPT '94,* pp 278-292, Springer-Verlag, Berlin, 1994. Early draft provided to ANSI X9.F.1,

[4] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard,* Springer-Verlag, New York, 1993.

[5] Matt Blaze, "A Cryptographic File System for Unix," Proceedings of the 1st ACM Conference on Computer and Communication Security, Fairfax, VA, 9-16 (November 1993).

[6] D. Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks." *IBM Journal of Research and Development*, May 1994, vol.38, (no.3):243-250.

[7] D. Coppersmith, D. B. Johnson and S. M. Matyas, "A proposed mode for triple-DES encryption," to appear, *IBM Journal of Research and Development*, 1996.

[8] Donald W. Davies, "Some regular properties of the 'Data Encryption Standard' algorithm," *Advances in Cryptology – Proceedings of Crypto 82,* 89-96, Plenum Press, 1983.

[9] Draft 5.2, X9.52, "TDEA Modes of Operation," 29 March 1996, X9.F.1 Committee, ANSI.

[10] Federal Information Processing Standards Publication (FIPS PUB) 46-2, *Data Encryption Standard (DES)*, National Institute of Standards and Technology, Washington, DC, 1993.

[11] M. Hellman, "A cryptanalytic time-memory trade-off," *IEEE Transactions on Information Theory,* Vol IT-26, no.4, 401-406, 1980.

[12] ISO/IEC 8372, "Information technology - Data cryptographic techniques - Modes of operation for a 64-bit block cipher algorithm," ISO/IEC 8372, 1987.

[13] ISO/IEC 10116, "Information technology - Security techniques - Modes of operation of an n-bit block cipher algorithm," IS 10116, 1991.

[14] M. Matsui, "The first experimental cryptanalysis of the Data Encryption Standard," *Advances in Cryptology – Crypto '94*, 1-11, Springer-Verlag, New York, 1994.

[15] Carl H. Meyer and Stephen M. Matyas, *Cryptography–A New Dimension in Computer Data Security*, John Wiley and Sons, New York, 1982.

[16] RSA Data Security, Inc. Product documentation, "Mailsafe Note #3."

[17] Frank Rubin, "Foiling an exhaustive key-search attack," *CRYPTOLOGIA 11*, No. 2, 102-107 (April 1987).

[18] P. C. van Oorschot and M. J. Wiener, "A Known-Plaintext Attack on Two-Key Triple Encryption," *Advances in Cryptology – Eurocrypt '90,* 318-325, Springer-Verlag, Berlin, 1991.

[19] P.C. van Oorschot and M.J. Wiener, "Parallel collision search with cryptanalytic applications," preprint.

[20] David Wagner, personal communication, 1996.

[21] M. J. Wiener, "Efficient DES Key Search," Technical Report TR-244, School of Computer Science, Carleton University, Ottawa, Canada, May 1994. Presented at the rump session of Crypto'93.