Cryptanalysis of Triple-Modes of Operation

Eli Biham[1]

Abstract

Multiple modes of operation, and in particular, triple-modes of operation were suggested as a simple method to improve the strength of blockciphers, and in particular of DES. The developments in the cryptanalysis of DES in the recent years have popularized the triple-modes of DES, and such modes are discussed now towards acceptance as ANSI standards.

In a previous paper we analyzed multiple modes of operation and showed that many multiple modes are much less secure than expected. In this paper we extend these results, with new cryptanalytic techniques, and show that all the triple modes of operation are theoretically not much more secure than a single encryption. We then suggest several candidates for more secure modes.

1 Introduction

Since the introduction of DES[17], and its modes of operation[18], many methods to improve its strength were suggested[1,7,14, and others]. The most popular of these methods is multiple modes of operation, and in particular, triple-modes of operation. These modes were believed[13,12] to be as strong as triple-DES against all kinds of attacks[5,16,4,19,2].

In [3] we have shown that the strength of many multiple modes is comparable to a single encryption, and we suggested to use single modes with an underlying cipher combined by multiple encryption, instead of using multiple modes.

In this paper we go one step further. We develop additional cryptanalytic techniques to attack multiple modes, and show that all the double modes and all the triple modes of DES, except the triple-ECB mode, are not much more secure than a single encryption against finding their keys. These techniques use known plaintext, chosen plaintext or chosen ciphertext attacks. Only a handful of these techniques require

[1]Computer Science Department, Technion - Israel Institute of Technology, Haifa 32000, Israel. Email: biham@cs.technion.ac.il, WWW: http://www.cs.technion.ac.il/~biham/.

more demanding attacks (such as adaptive attacks or known initial value attacks) to find the complete key. If we also consider the dictionary attack against the triple-ECB mode (in which case the key remains unknown, but the attacker can encrypt and decrypt under the unknown key using 2^{64} known plaintexts), all the triple modes are *theoretically* no more secure than a single encryption. We should emphasize that we say *theoretically*, since in some practical means the situation is different: We denote the complexity t as the maximal complexity of the required data, the required memory and the computation steps. Wiener's search machine[19] can find a single-DES key in 3.5 hours in average, but requires only a small memory and one plaintext/ciphertext pair. We can use such a machine for searching the keys of the DES components, and thus have a relatively fast searching times. However, the complexity of our attacks are measured also by the number of the required known/chosen blocks and memory size. Therefore, for example, even 2^{50} complexity might be practically more difficult than the complexity of exhaustive search of DES. Still the complexities we quote are interesting for the theoretical analysis of the modes of operations, and they can serve as useful lower bounds for the strengths of the modes.

All the attacks on all the modes we describe in this paper exhaustively search for the keys using information obtained by the various techniques, and do not assume any special assumptions on the internals of the underlying blockciphers. In particular, all these attacks do not use differential cryptanalysis[5], linear cryptanalysis[16], nor other kinds of attacks based the internals of the ciphers[2,4]. Therefore, all the attacks are applicable to any blockcipher, and their complexity depends only on the block size and the key size.

Our results led us to develop new modes of operation, which are more secure than the regular triple-modes, easily pipelinable, and do not require more applications of the underlined cipher than the triple-modes. These modes do not necessarily satisfy the conjecture from [3]. These modes are of two types: triple modes which combine three single modes in a non-standard way, and quadruple modes.

In Section 2 we describe our novel techniques. In Section 3 we elaborate some more information on some of the more complicated attacks on specific modes. In Section 4 we suggest new candidates for more secure (pipelinable) modes of operation. Section 5 summarizes the paper. Finally, we enclose an appendix in which we give brief hints on the techniques of the attacks on each of the single, double and triple modes of DES, and their complexities.

2 Novel Techniques

In this section we describe very powerful novel techniques for analyzing multiple modes of operation. They require an order of 2^{65}–2^{67} chosen plaintexts. Note that although 2^{65}–2^{67} chosen plaintexts are more than the number of possible one-block plaintexts, encryption of multiple modes results with more than 2^{64} distinct plain-

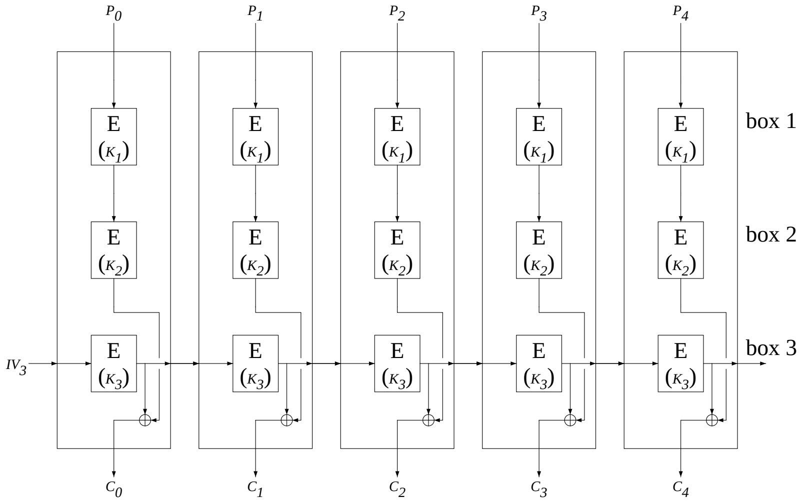

Figure 1. The triple mode: ECB|ECB|OFB.

text/ciphertext pairs, since the internal feedbacks add memory to the encryption process, and the ciphertext is a function of the plaintext and of the feedbacks.

2.1 Technique G

The techniques in [3] which do not use any information on the internals of the underlying blockcipher, e.g., exhaustive search and the birthday techniques, can only attack modes whose plaintexts or ciphertexts are the actual inputs or outputs of some encryption box. This novel technique overcomes this difficulty, and enables to attack modes whose plaintext and ciphertext are mixed with some unknown feedback, or do not affect the encryption box at all. This technique is later used as a building block by the techniques described afterwards.

We describe this technique on two examples: the double mode CBC|OFB and on the triple mode ECB|ECB|OFB.

The attack on the ECB|ECB|OFB (see Figure 1) is as follows: At the first stage of the attack, we cannot hope to attack the ECB modes, since the OFB mode hides the information required for the meet in the middle attack[11, page 83] on the double-ECB mode. Thus, we attack the OFB mode. We choose some arbitrary (one-block) value A, and ask for the ciphertext of 2^{64} consecutive plaintext blocks A, followed by a few blocks with distinct values. The period of the OFB is at most 2^{64}, and thus the period can be identified from the ciphertext.

Assume that $A' = E_{K_2}(E_{K_1}(A))$ and assume that the OFB stream is $v_0, v_1, \ldots, v_{2^{64}-1}$. Then, the received ciphertext is $v_0 \oplus A', v_1 \oplus A', \ldots, v_{2^{64}-1} \oplus A'$ followed by a few other blocks. From the ciphertexts we can compute the differences of consecutive

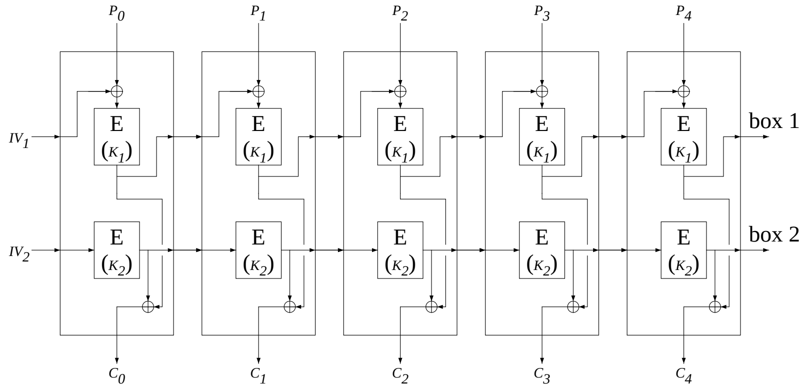

Figure 2. The double mode: CBC|OFB.

OFB blocks $v_0 \oplus v_1$, $v_1 \oplus v_2$, $\ldots$, $v_{2^{64}-2} \oplus v_{2^{64}-1}$.

We now guess an arbitrary value u_0 to appear as one of the v_i's in the cycle of the OFB stream. We exhaustively encrypt u_0 under all the possible keys K, and get $u_1 = E_K(u_0)$, and $u_2 = E_K(u_1)$, and compute the differences $u_0 \oplus u_1$ and $u_1 \oplus u_2$. If these two values do not appear as two consecutive values in the differences $v_0 \oplus v_1$, $v_1 \oplus v_2$, $\ldots$, $v_{2^{64}-2} \oplus v_{2^{64}-1}$, then the key K is not the key or u_0 is not in the OFB cycle. If all the keys do not pass this test for a particular u_0, then certainly u_0 is not in the cycle, and we should try another one. In average we need to try about $2^{64}/\text{period(OFB)}$ u_0's, which is expected to be small, since the period of the OFB mode is expected to be about 2^{64} (see [9]).

Once we found matching u_0 and K, such that $u_0 \oplus u_1 = v_i \oplus v_{i+1}$ and $u_1 \oplus u_2 = v_{i+1} \oplus v_{i+2}$, we assume that we found the key $K_3 = K$, and conclude that $u_0 = v_i$, $u_1 = v_{i+1}$, $u_2 = v_{i+2}$, and identify A' and all the OFB cycle, including the initial value. After peeling up the OFB mode, we remain with a few distinct double-ECB encryptions, which can be attacked by the meet-in-the-middle attack[11].

The attack on the CBC|OFB mode (see Figure 2) is slightly more complex. We choose two block values A and B, and choose the ciphertext as 2^{64} blocks A followed by 2^{64} blocks B. Since the period p of the OFB mode is at most 2^{64}, it can be identified from the plaintext. Thus, we result with $A \oplus v_0$, $A \oplus v_1$, $\ldots$, $A \oplus v_{2^{64}-1}$, $B \oplus v_{2^{64}}$, $B \oplus v_{2^{64}+1}$, $\ldots$, $B \oplus v_{2^{65}-1}$ as the output of the CBC mode, and $P_0 \oplus IV_1$, $P_1 \oplus A \oplus v_0$, $P_2 \oplus A \oplus v_1$, $\ldots$, $P_{2^{64}-1} \oplus A \oplus v_{2^{64}-2}$, $P_{2^{64}} \oplus A \oplus v_{2^{64}-1}$, $P_{2^{64}+1} \oplus B \oplus v_{2^{64}}$, $P_{2^{64}+2} \oplus B \oplus v_{2^{64}+1}$, $\ldots$, $P_{2^{65}-1} \oplus B \oplus v_{2^{65}-2}$ as the input of the encryption box of the CBC mode. Since the period $p \leq 2^{64}$, we can eliminate the unknown v_i's by computing differences between the cycle in the first half of the data and the cycle in the second half of the data, and get $A \oplus B$ as the difference of all the output pairs of corresponding blocks $0 < i < 2^{64} < j < 2^{65}$ (where p divides $j - i$ and $v_j = v_i$), and $(P_j \oplus B \oplus v_{j-1}) \oplus (P_i \oplus A \oplus v_{i-1}) = P_j \oplus P_i \oplus A \oplus B$ as the difference of the inputs of the encryption box.

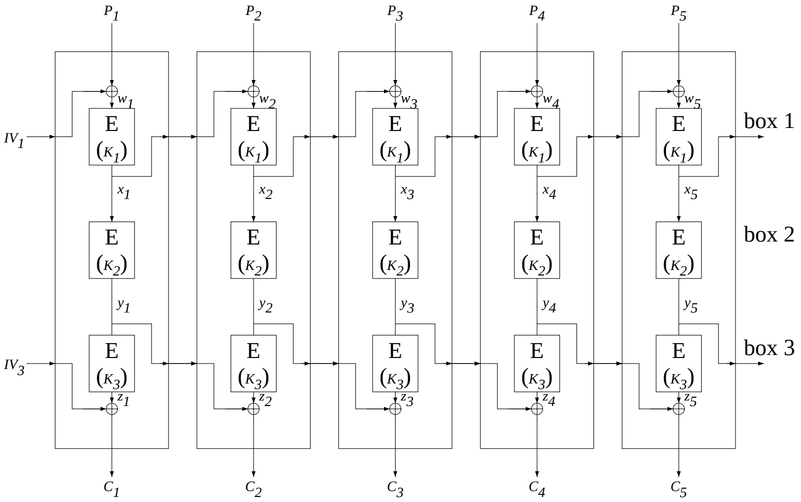

Figure 3. The triple mode: CBC|ECB|CBC^{-1}.

We received period(OFB) differences, whose all output differences are the same $A \oplus B$. We guess u_0, compute its counterpart $u'_0 = u_0 \oplus A \oplus B$, and exhaustively search for the key K which satisfies

$$D_K(u_0) \oplus D_K(u'_0) = P_j \oplus P_i \oplus A \oplus B$$

for some i and j satisfying the above conditions, and verify that it also satisfies the differences of the next block

$$E_K(u_0 \oplus P_{i+1}) \oplus E_K(u'_0 \oplus P_{j+1}) = A \oplus B$$

(or to improve performance, also $E_K(u'_0 \oplus P_{i+1}) \oplus E_K(u_0 \oplus P_{j+1}) = A \oplus B$). Again, only a small number of guesses for u_0 is required in order to find the key of the CBC mode and the initial value.

2.2 Technique H

This technique is the most sophisticated technique so far, and it allows to attack most of the modes that could not be attacked with simpler techniques. It is particularly suitable to attack modes with both feedbackward and feedforward feedbacks, which hide both the input of the first encryption box and the output of the last encryption box from the attacker.

We describe the application of this technique to the CBC|ECB|CBC^{-1} mode, shown in Figure 3 (which was conjectured in [12] to be the strongest triple mode). In this mode, we denote the intermediate values between the plaintext, the encryption boxes, and the ciphertext by w_i, x_i, y_i and z_i as is shown in the figure.

We choose some arbitrary (single-block) value S, and request the ciphertext of 2^{66} tuples of the form (R_i, S, S, S), where the R_i's are chosen at random. We denote these $4 \cdot 2^{66}$ plaintext blocks by P_1, P_2, ..., $P_{2^{68}}$, and their values are given by

$$P_{4i-3} = R_i \qquad P_{4i-2} = S \qquad P_{4i-1} = S \qquad P_{4i} = S \qquad (1 \leq i \leq 2^{66}).$$

After encrypting the plaintext, the attacker gets the 2^{68} ciphertext blocks C_1, C_2, ..., $C_{2^{68}}$.

We denote the equivalence set of all the (indices of the) tuples whose last three ciphertext blocks are the same as of a given tuple by

$$\theta(i) = \{j | (C_{4j-2}, C_{4j-1}, C_{4j}) = (C_{4i-2}, C_{4i-1}, C_{4i})\}.$$

We will further denote $i \equiv j$ if $j \in \theta(i)$ (and necessarily $i \in \theta(j)$, and $\theta(i) = \theta(j)$). It is expected that all the tuples in $\theta(i)$ (including $i \in \theta(i)$) not only their last three blocks are the same in the plaintext and the ciphertext, also the intermediate values (w, x, y, z) have the same property (e.g., if $i \equiv j$ then $(x_{4j-2}, x_{4j-1}, x_{4j}) = (x_{4i-2}, x_{4i-1}, x_{4i})$, etc.). The probability that there are some i and j for which it does not hold is negligible.

We concentrate on some $i \equiv j$ $(i, j > 1)$. For such i and j, $x_{4i-3} = x_{4j-3}$, and thus $w_{4i-3} = w_{4j-3}$. However, for any k, $x_{4k-4} \oplus P_{4k-3} = w_{4k-3}$, and therefore we get differences between the unknown intermediate values x:

$$x_{4i-4} \oplus x_{4j-4} \quad = \quad P_{4i-3} \oplus P_{4i-3} = R_i \oplus R_j.$$

Similarly, we get differences between the unknown intermediate values y:

$$y_{4i-4} \oplus y_{4j-4} \quad = \quad C_{4i-3} \oplus C_{4i-3}.$$

Note that from this single equivalence set we got several differences, since for any $j, k \in \theta(i)$ we found the difference $x_{4j-4} \oplus x_{4k-4}$, and the tuples of the $4j-4$'th and the $4k-4$'th blocks belong to different equivalence sets. Each equivalence set in our 2^{66} tuples contains in average four tuples, and thus from each equivalence set we get in average differences between 16 elements of the x's. By computing the equivalence sets of all the tuples following those for which we know these differences (e.g., $\theta(l)$ where the difference of x_{4l-4} with some other values is already known), and computing the differences for the larger sets, we increase the set of x's for which the differences are known by 144 ($(16-4) \cdot 3 \cdot 4$). After several iterations, the size of this set grows to include all (or almost all) the tuples in the data.

Therefore, after completing the above analysis, we can compute $x_{4i} \oplus x_{4j}$ and $y_{4i} \oplus y_{4j}$ for any i and j. Alternatively, we can write

$$\begin{aligned} x_{4i} &= X \oplus u_{4i} \\ y_{4i} &= Y \oplus v_{4i} \end{aligned}$$

where $u_{4i} = x_{4i} \oplus x_4$ and $v_{4i} = y_{4i} \oplus y_4$ are known, and where only $X = x_4$ and $Y = y_4$ are unknown.

Till now we did not efficiently use all the three plaintext blocks S in the tuples, since if we would have only two such blocks in each tuple, the probabilities for mistakes in the predictions of the equivalences would still be very low. The reason for the extra block S is to find differences of two consecutive blocks, while till now we know only the differences of blocks whose indices are multiples of four. Since we need only two consecutive ciphertext blocks whose plaintexts are S in order to conclude equivalence of two tuples, we can now do it not only on the full tuples, but also shifted by one block, e.g., to compare the second and third blocks inside the tuples to the third and fourth in another tuple, by searching for each i, some j for which

$$(C_{4i-2}, C_{4i-1}) = (C_{4j-1}, C_{4j}),$$

and for each such equality concluding that $x_{4i-1} = x_{4j}$ and $y_{4i-1} = y_{4j}$, and thus also

$$u_{4i-1} = u_{4j} \qquad \text{and} \qquad v_{4i-1} = v_{4j}.$$

We got about 2^{66} pairs of blocks $(4i-1, 4i)$ for which we know $x_{4i-1} \oplus x_{4i} = u_{4i-1} \oplus u_{4i}$ and $y_{4i-1} \oplus y_{4i} = v_{4i-1} \oplus v_{4i}$. The encryption equations

$$\begin{aligned} E_{K_1}(P_{4i} \oplus x_{4i-1}) &= x_{4i} \\ E_{K_3}(y_{4i}) &= y_{4i-1} \oplus C_{4i} \end{aligned}$$

can be written using u and v as:

$$\begin{aligned} E_{K_1}(S \oplus X \oplus u_{4i-1}) &= X \oplus u_{4i} \\ E_{K_3}(Y \oplus v_{4i}) &= Y \oplus v_{4i-1} \oplus C_{4i}. \end{aligned}$$

Since the blocksize is only 64 bits, we get only about 2^{64} distinct instances of the equations, each instance occurs in average four times, and (almost) all the possible values of inputs to the encryption function E are expected.

We now apply another variant of Technique G: we choose some arbitrary (one-block) value a. It is expected that in one of the above instances, $E_{K_1}(a)$ is computed, but we do not know in which. We compute

$$b = E_{K_1'}(a)$$

under all the possible keys K_1' in the keyspace of K_1, and search for the K_1' for which the equality

$$a \oplus E_{K_1'}(a) = (S \oplus X \oplus u_{4i-1}) \oplus (X \oplus u_{4i}) = S \oplus u_{4i-1} \oplus u_{4i}$$

holds. This equation must hold when $S \oplus X \oplus u_{4i-1} = a$. If it holds, we assume for a moment that $S \oplus X \oplus u_{4i-1} = a$ and conclude that $X = S \oplus u_{4i-1} \oplus a$. With the key K_1' and X, we can now encrypt the following blocks and verify that there are no contradictions in the values of x_{4i+3} and x_{4i+4}. If their values do not fit, we try another key K_1' till we find some value which fits. If the values do not fit for any key, then it must be that $E_{K_1}(a)$ is not in the above equations; in this case we choose another value for a, and try again. After a few trials, we find the key K_1 and the value $X = x_4$ with a high probability. From these values, we can complete the knowledge of all the x_i's, and in particular the initial value IV_1. Similarly, we can find K_3, $Y = y_4$, and IV_3. K_2 can now be found by exhaustive search (or by differential or linear cryptanalysis).

2.3 Technique I: Double Stream Cipher Technique

This technique can analyse modes in which the ciphertext is an XOR of two streams. In particular, it can analyse the double OFB mode (whose output is the XOR of the plaintext and of two OFB streams), and many triple modes whose last (or first) mode is an OFB.

Assume we know that some stream $\{s_i\}$ is the result of XORing two cyclic streams $\{a_i\}$ and $\{b_i\}$, by $s_i = a_i \oplus b_i$. In a double OFB mode, we can find the stream s_i with a known plaintext attack, by XORing the plaintext with the ciphertext: $s_i = P_i \oplus C_i$. The purpose of this technique is primarily to find the periods of $\{a_i\}$ and $\{b_i\}$, and then this information will be used to find the keys and the initial values.

We denote the period of $\{a_i\}$ by k and the period of $\{b_i\}$ by l, and assume without loss of generality that $k < l$.[2] Both periods are expected to be about 2^{64}, and thus the expected period of $\{s_i\}$ is about 2^{128}.

We write

$$s_i = a_i \oplus b_i = a_{i \bmod k} \oplus b_{i \bmod l}.$$

Given $k + l + 1$ consecutive blocks of $\{s_i\}$: $s_0, s_1, \ldots, s_{k+l}$, the following equations hold

$$\begin{aligned}
s_0 &= a_0 \oplus b_0 \\
s_k &= a_k \oplus b_k = a_0 \oplus b_k \\
s_l &= a_l \oplus b_l = a_l \oplus b_0 \\
s_{k+l} &= a_{k+l} \oplus b_{k+l} = a_l \oplus b_k
\end{aligned}$$

from which we derive

$$s_0 \oplus s_k \oplus s_l \oplus s_{k+l} = 0$$

We will later use this equation in the form:

$$s_0 \oplus s_{k+l} = s_k \oplus s_l.$$

In order to use this equation, we need to know k and l. We identified that we can modify it slightly, in such a way that we do not need to know them, thus it enables

[2]We ignore the rare case where $k = l$ in which the period of $\{s_i\}$ is itself $k = l$.

us to find them. We rewrite the equations (shifted in the stream by $i - \lfloor (m+l)/2 \rfloor$ blocks, for any i) as

$$\begin{aligned} s_{i-\lfloor (k+l)/2 \rfloor} &= a_{i-\lfloor (k+l)/2 \rfloor} \oplus b_{i-\lfloor (k+l)/2 \rfloor} \\ s_{i-\lfloor |k-l|/2 \rfloor} &= a_{i-\lfloor |k-l|/2 \rfloor} \oplus b_{i-\lfloor |k-l|/2 \rfloor} = a_{i-\lfloor (k+l)/2 \rfloor} \oplus b_{i+\lceil (k+l)/2 \rceil} \\ s_{i+\lceil |k-l|/2 \rceil} &= a_{i+\lceil |k-l|/2 \rceil} \oplus b_{i+\lceil |k-l|/2 \rceil} = a_{i+\lceil (k+l)/2 \rceil} \oplus b_{i-\lfloor (k+l)/2 \rfloor} \\ s_{i+\lceil (k+l)/2 \rceil} &= a_{i+\lceil (k+l)/2 \rceil} \oplus b_{i+\lceil (k+l)/2 \rceil} \end{aligned} \tag{1}$$

and thus

$$s_{i-\lfloor (k+l)/2 \rfloor} \oplus s_{i+\lceil (k+l)/2 \rceil} = s_{i-\lfloor |k-l|/2 \rfloor} \oplus s_{i+\lceil |k-l|/2 \rceil}.$$

We define S_i to be the concatenation of s_i and s_{i+1}, in order to increase the blocksize and to reduce random effects. From now on we restrict the discussion only to even $k+l$; odd $k+l$ can be analyzed similarly with only small changes. Then,

$$S_{i-(k+l)/2} \oplus S_{i+(k+l)/2} = S_{i-|k-l|/2} \oplus S_{i+|k-l|/2}.$$

We define

$$\Delta_j = S_{2^{64}-j} \oplus S_{2^{64}+j}$$

(similarly, for odd $k+l$ we should define $\Delta_j = S_{2^{64}-j} \oplus S_{2^{64}+j+1}$) and for $i = 2^{64}$ get the equation

$$\Delta_{(k+l)/2} = \Delta_{|k-l|/2}.$$

The attack we use is a start in the middle attack. Given 2^{65} stream blocks, we compute Δ_j for each $j = 1, 2, \ldots, 2^{64}$ and search for two distinct j's for which

$$\Delta_{j_1} = \Delta_{j_2}$$

(by keeping the values in a hash tables, and waiting till some value is entered twice into the table). These j_1 and j_2 satisfy

$$S_{2^{64}-j_1} \oplus S_{2^{64}-j_2} \oplus S_{2^{64}+j_2} \oplus S_{2^{64}+j_1} = 0$$

from which we conclude that

$$k = |j_1 - j_2| \qquad \text{and} \qquad l = j_1 + j_2.$$

Given the candidate values for k and l, it is easy to verify that Equation 1 holds for several additional i's. If it does not, the choice of k or l is wrong, and we should find another pair of j_1 and j_2.

Till here, we found only the periods of the two streams, but yet we do not know anything more than the periods. At this point we will use the periods in order to find information on the streams $\{a_i\}$ and $\{b_i\}$. We assume that k and l are coprimes (or at least that $\gcd(k, l)$ is small, which can be analyzed with only a minor modification to the following algorithm). We compute the following algorithm, which results with $d_i = a_i \oplus a_0$ for every i:

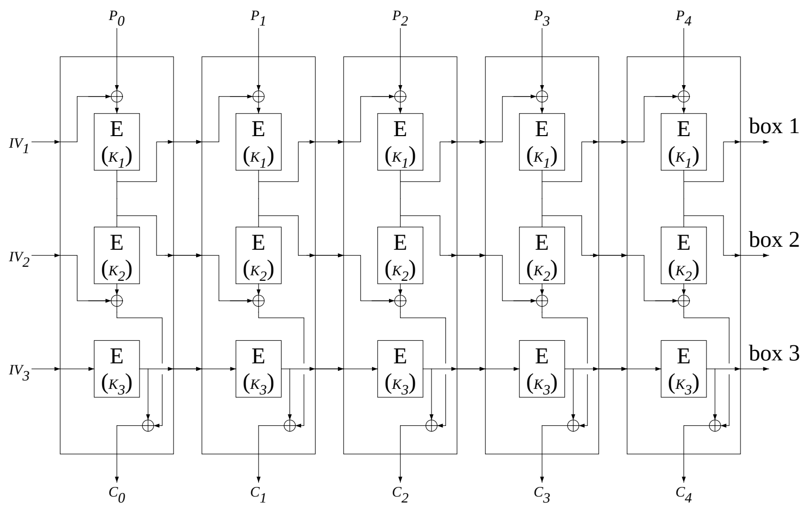

Figure 4. The triple mode: CBC|CBC^{-1}|OFB.

$d_0 = 0$
for $i = 1$ to $k-1$ do
  $d_{il \bmod k} = d_{(i-1)l \bmod k} \oplus (s_{(i-1)l \bmod k} \oplus s_{((i-1)l \bmod k)+l})$

We get all the $d_i = a_i \oplus a_0$, $i = 0, \ldots, k-1$. Now, the stream $\{a_i\}$ is known up to an XOR with one block a_0. The stream $\{b_i\}$ is known under the same condition, since $b_i = s_i \oplus a_i = s_i \oplus d_i \oplus a_0$.

When one of the two streams is the output of an OFB mode (without loss of generality we assume that the OFB stream is $\{b_i\}$), we can use Technique G to solve the key and the initial value. Then, the $\{a_i\}$ stream can be computed by $a_i = s_i \oplus b_i$. If $\{a_i\}$ is also an output of an OFB mode, we can now find the key easily by exhaustively search for the key K for which the equation $E_K(a_1) = a_2$ is satisfied.

This attack is also applicable to many triple modes in which the first (or last) single mode is an OFB, by feeding the other two modes by carefully chosen plaintexts or ciphertexts. The CBC|CBC^{-1}|OFB is an example (see Figure 4). In this mode we fix 2^{65} same plaintext blocks, and get a cycle in the output of first CBC mode. The cycle is not affected by the next CBC^{-1} mode. The result is the XOR of the cycle of the CBC|CBC^{-1} mode and the OFB mode. The above technique can find the periods, and analyse the OFB mode, resulting with the key and the initial value of the OFB mode, and with the exact output of the CBC|CBC^{-1} mode. The CBC|CBC^{-1} mode can then be analyzed with additional 2^{66} chosen plaintexts using Technique H.

Note that this technique is also applicable to stream ciphers, in which the output of two non-linear streams (such as the outputs of NLFSRs) are XORed. This technique can also increase the efficiency of the Berlekamp-Massey algorithm[15] when the outputs of two LFSRs are XORed, by dividing the original problem into two smaller problems in a linear time, and using the Berlekamp-Massey algorithm on each of the smaller problems. In the next technique we show that not only the complexity of the Berlekamp-Massey algorithm can benefit from this attack, attacks on multiple OFB modes can benefit from the Berlekamp-Massey algorithm.

2.4 Technique J: Triple and Multiple Stream Cipher Technique

This technique extends Technique I for the case of an XOR of more than two streams. This "miracle" is obtained using the Berlekamp-Massey algorithm[15] to retrieve the shortest linear feedback shift register that generates the stream. This shortest LFSR is expected to uncover (in most cases) the periods of all the summated streams.

For example, we show the case where three streams are summated (as in the triple-OFB mode): $\{a_i\}$, $\{b_i\}$, and $\{c_i\}$, whose periods are k, l and m, respectively. The summated stream is $s_i = a_i \oplus b_i \oplus c_i$. Then, the following LFSR generates the summated stream:

$$s_{k+l+m} = s_0 \oplus s_k \oplus s_l \oplus s_m \oplus s_{k+l} \oplus s_{k+m} \oplus s_{l+m}.$$

Given the $2(k+l+m)$ first blocks of $\{s_i\}$, the Berlekamp-Massey algorithm reproduces the LFSR from which the periods k, l and m can be deduced (assuming that the LFSR above is the shortest LFSR generating the stream, and it is expected to be, especially since we get 64 simultaneous such streams).

The Berlekamp-Massey algorithm requires $2n$ blocks, and whose complexity is $O(n^2)$, where n is the length of the shortest LFSR, i.e., in the example $n = k + l + m$. However, this complexity is too high for our purposes. Fortunately, there is a variant of the Berlekamp-Massey algorithm, called the recursive Berlekamp-Massey algorithm[6, section 11.7], whose complexity is lower than $O(n \log n \log\log n)$. This complexity is about $2^{66} \cdot 66 \cdot \log 66 = 2^{75}$ when $n \le 2^{66}$, i.e., when up to four OFB streams are summated. These 2^{75} steps are very simple, and are faster to apply than 2^{64} DES encryptions.

Given the periods of the individual streams (k, l and m), we decompose the individual streams, up to an XOR with their first block (which remains unknown), with a similar method to the one used by Technique I. In the case of three streams, as in the above example, the decomposition algorithm is

$e_0 = 0$
for $i = 1$ to $k-1$ do
$$e_{im \bmod k} = e_{(i-1)m \bmod k} \oplus (s_{(i-1)m \bmod k} \oplus s_{((i-1)m \bmod k)+l} \oplus s_{((i-1)m \bmod k)+m} \oplus s_{((i-1)m \bmod k)+l+m})$$
$d_0 = 0$

for $i = 1$ to $k-1$ do
$$d_{2il \bmod k} = d_{2(i-1)l \bmod k} \oplus (e_{2(i-1)l \bmod k} \oplus e_{(2(i-1)l \bmod k)+l})$$

where e_i results with $e_i = a_i \oplus a_{i+l} \oplus a_0 \oplus a_l$, and d_i results with $d_i = a_i \oplus a_0$. Form these values, we can now compute $a_i \oplus a_{i+1}$ for any i (or at least $a_i \oplus a_{i+2}$ for even k's), and similarly we can compute the differences of consecutive blocks in all the individual streams. Technique G is then used to solve the keys and the initial values of the individual OFB streams. The total complexity of this attack is equivalent to less than 2^{64} DES encryptions, and the attack requires about 2^{67} blocks and 2^{66} memory, when up to four streams are summated.

3 The Strength of All the Triple Modes

In this section we elaborate some more information on some of the more complicated attacks. Brief hints on the attacks on all the modes are given in the appendix.

1. ECB|CBC|ECB: The attack requires five chosen ciphertext blocks of the form (A, A, B, B, A), for any block values A and B ($A \neq B$). The intermediate values after the first ECB component (under the real unknown key) must be of the form $(?, A' \oplus A'', B' \oplus A'', B' \oplus B'', A' \oplus B'')$, where A' and B' are the intermediate values entering to the second encryption box (after mixing with the feedbacks), and A'' and B'' are the intermediate values after the second component. These values depend only on the values of the corresponding ciphertext blocks (A or B), since no feedbacks are mixed during their encryption. Therefore, the XOR of the last four (of the five) blocks after the first component is zero. We need only to encrypt the four corresponding plaintext blocks by all the possible keys of the first component, and verify which key results with such zero XOR. The last ECB component can then be analyzed with three blocks of the above five (A, B, B). Don Coppersmith has also developed a similar attack on this mode[8].

2. ECB|CFB^{-1}|CBC: We describe two attacks on this mode. In the first attack we choose three same plaintext blocks (A, A, A). The intermediate data after the second component is thus of the form $(?, A'', A'')$. We now try to decrypt the ciphertext under all possible keys of the last component, and see which key results with two equal second and third blocks. To find the keys of the first and the second components we need an additional plaintext block, so in total we need four blocks (A, A, A, B) to find all three keys.

   The second attack is a meet in the middle attack. We choose tuples (A, A, B), so we can conclude that the XOR of the intermediate value of the second and third blocks after the second component equals the XOR of the same blocks after the first component. Encrypt the two corresponding plaintext blocks under all possible keys, XOR the results of the two blocks, and keep the result in a table. Now, decrypt the last component under all possible keys and check whether the XOR of the second and third resultant blocks appears in the table. If for two tuples, the same keys succeed, we conclude that they are the keys of the first and last components. For this attack, the plaintexts (A, A, B, B, A) suggest the required two tuples.

3. CBC^{-1}|ECB|CBC: The attack on this mode is similar to the first attack on the ECB|CFB^{-1}|CBC mode, where the required chosen plaintext blocks (or symmetrically, chosen ciphertext blocks) are of the form (B, A, A, A).

4. CBC^{-1}|OFB|CBC: We can identify the period of the OFB component by choosing 2^{64} chosen plaintext (or ciphertext) blocks A followed by 2^{64} tuples of the form (R_i, A, A, A). It is expected that for some tuple i, the feedback from the first block R_i to the next will be the same as in some block in the first 2^{64} blocks whose distance is a multiple of the period. In this case, we can use this knowledge to identify the period. Once the period is known, we can use the same data to predict two equal values after the second component, and use them to find the key of the last component. Then, the other keys can be completed with the same data and simpler analysis.

5. CBC|OFB|CBC: We choose 2^{32} ciphertext blocks A, followed by 2^{32} tuples of 2^{32} ciphertext blocks of the form $(A, A, A, A, R_i, R_i, R_i, R_i, \ldots)$, followed by 2^{64} ciphertext tuples of the form (R_i, R_i, R_i, R_i), where R_i are random (the same at each tuple). From the first $2^{64} + 2^{32}$ blocks, we can identify the OFB period. Then we can use all the tuples and search for collisions in the decryption of the last CBC component in distances of multiples of the OFB period (the collisions are similar to the collisions in the birthday attack, but in this case, the birthday paradox is not involved). We can identify such collisions since the corresponding plaintext blocks are expected to be the same. When such a collision found, we continue like in the birthday attack, and find the key of the last component. We complete the rest of the keys with Technique G.

6. ECB|CFB^{-1}|CBC^{-1} and ECB|CBC^{-1}|CFB^{-1}: We choose 2^{34} plaintext tuples of the form $(F, R_i, B_1, F, R_i, B_2)$ or $(B_1, R_i, F, B_2, R_i, F)$ respectively, where F, B_1 and B_2 are fixed in all the tuples, and R_i varies at random (but the same in both occurrences in each tuple). We expect a birthday between $E_{K_1}(B_1) \oplus E_{K_2}(E_{K_1}(R_i))$ and $E_{K_1}(B_2) \oplus E_{K_2}(E_{K_1}(R_j))$, for some i and j. The same blocks will have another match when the indices i and j are exchanged (i.e., in the other halves of the tuples).

   We identify the birthday by computing the difference of the third and sixth ciphertext blocks in each tuple, and searching for equality of the results in two distinct tuples. When found, we can compute $E_{K_1}(R_i) \oplus E_{K_1}(R_j)$ as the difference of two ciphertext blocks, and can exhaustively search for K_1 that satisfies this equation.

7. CBC|CFB|OFB and CFB|CBC|OFB: We choose some arbitrary values A and B, and choose the ciphertext to be 2^{64} consecutive blocks A, followed by 2^{64} tuples of three blocks (B, A, A). After identifying the period of the OFB, we compute the differences in the CBC and the CFB modes, in the same positions in the OFB cycle, so the OFB values are canceled in the differences. Thus, we receive differences $A \oplus B$ in many blocks, and various input differences in the same blocks, and apply Technique G to solve the keys.

8. CBC|CBC^{-1}|CBC and similar modes: This is a birthday attack, in which we look for birthday not only in the last CBC mode, but also in the CBC^{-1} stream. We choose 2^{64} random ciphertext tuples (R_i, R_i, R_i, R_i), and search for the pairs i, j in which $D(R_i) \oplus R_i = D(R_j) \oplus R_j$ (as in the regular birthday technique) together with equal values in the first block of the tuple in the CBC^{-1} mode. Given 2^{64} such tuples, we expect such a birthday. We can identify it easily, since when it occurs, the last three of the four plaintext blocks of one tuple equal the corresponding blocks in the other tuple. Then, we find the key of the last CBC mode just as we do in the regular birthday technique.

   The first two modes are then attacked by Technique H with the same data.

   If the first two modes are CBC|CFB^{-1} or CFB|CBC^{-1}, the data do not suffice. In this case we choose additional tuples (R'_i, A, A, A, A) to the original chosen ciphertext, which after decryption of the last CBC becomes (R''_i, R'''_i, B, B, B), on which we can apply Technique H.

9. CBC|CBC|CBC^{-1}, CFB|CFB|CFB^{-1} and similar modes: The last CBC^{-1} (or CFB^{-1}) single mode is attacked by Technique H with chosen ciphertext tuples (R_i, A, A, A, A).

   If the first two modes are CBC|CFB or CFB|CBC, from a pair in which the difference is of the form (known, 0, 0) or (0, 0, known) we can find the keys of both modes.

   If the first two modes are CBC|CBC or CFB|CFB, the problem is more complicated. One way to find the rest of the keys is to mount an adaptive attack, which after identifying the last key, can attack the remaining double-mode with the data required for such an attack. Another solution will be described in the next subsection.

3.1 Modes which Require More Demanding Attacks

In this subsection we elaborate the attacks on modes which we do not know to attack under known plaintext, chosen plaintext, nor chosen ciphertext attacks which require only one stream to be encrypted. In some sense these modes are the strongest triple modes: the attacks we use against these mode demand extra information: either to know the initial values, or to encrypt two streams with the same keys and the same known initial values. However, we believe that there are other attacks on these modes that do not need the extra requirements.

1. CBC|CBC|CBC^{-1}, CFB|CFB|CFB^{-1} and similar modes, whose first two components are the same: The last CBC^{-1} (or CFB^{-1}) mode is attacked by Technique H as is explained above in item 9.

   As we mentioned there, one way to find the rest of the keys is to mount an adaptive attack, which after identifying the last key, can attack the remaining double-mode with the data required for such an attack.

   A non-adaptive attack can be mounted if we know the initial values IV_1 and IV_2 of these two modes. In this case, we can use the meet-in-the-middle attack to solve both keys.

2. CBC|OFB|CBC^{-1} (and similarly CFB|OFB|CFB^{-1}): If the attacker can receive two (known) streams encrypted under the same keys and the same known initial values, he can find K_1 and K_3 by a meet in the middle attack, in which he keeps $(E_{K_1}(IV_1, P) \oplus E_{K_1}(IV_1, P^*), K_1)$ for every possible K_1 in a table and searches for a K_3 such that $D_{K_3}(IV_3, C) \oplus D_{K_3}(IV_3, C^*)$ appears in the first field in the table (where $E_{K_1}(IV_1, P)$ denotes CBC encryption of the first component, and $D_{K_3}(IV_3, C)$ denotes CBC^{-1} decryption of the last component). Only two ciphertext blocks are required in each stream for the attack to succeed.

3. CBC|CBC|OFB (and similarly CFB|CFB|OFB): If the attacker can receive two chosen streams encrypted under the same keys and the same known initial values, he can find K_1 by choosing the ciphertext streams (A, A) and (B, A) and get the plaintexts P and P^*. He trys all the keys K_1 till the second block of the difference $E_{K_1}(IV_1, P) \oplus E_{K_1}(IV_1, P^*)$ equals $A \oplus B$.

   Another attack requires only one stream with known IV: the attacker chooses a ciphertext stream of one B followed by 2^{64} A's. From the plaintext, he can identify the period of the OFB component, and find the first A in the second period. The feedbacks entering for the block of this A equal the initial values, and thus the above attack can be applied.

4 More Secure Modes

So far we showed that all the triple modes (possibly except the triple-ECB mode), are theoretically not much more secure than a single encryption. In this section we suggest modes, which are candidates to be more secure, and whose complexity of attack is expected to be about 2^{112}–2^{128}. We start with suggestions of triple modes combined in non-standard ways (whose encryption speed is just as of triple-DES, and which can be efficiently pipelined in hardware), and then suggest secure quadruple modes.

4.1 Definitions of Non-Standard Combinations of Modes

In all the discussions above, we only considered multiple modes combined as cascade modes, in which the output of one mode becomes the input to the next mode, and we denoted this cascade by the '|' operator.

We now consider other operators to combine modes. We denote the first of them by the operator '$\rightarrow$', which takes two modes M1 and M2, where M2 is any mode, and M1 is any stream mode (such as the OFB mode) generating a stream independent of the plaintext, and XORs the stream to the plaintext to form the ciphertext[3]. Then, M1$\rightarrow$M2 is the mode which computes the stream generated by M1, and encrypts the result under M2, resulting with a new stream, to be XORed to the plaintext.

For example, the OFB$\rightarrow$CBC mode applies the CBC mode on the stream generated by the OFB mode, and XORs the result to the plaintext. We can combine more than one mode in this way; for example, the OFB$\rightarrow$CBC$\rightarrow$CFB mode applies the CFB mode on the output of OFB$\rightarrow$CBC mode.

Another operator, which we denote by 'M1[M2]', where M1 is any stream mode (or any other mode subject to the footnote), applies the stream mode M1 on the plaintext (e.g., XORs the plaintext and the stream of M1), applies the mode M2 on the result, and applies the *same* M1 on the result (e.g., XORs with the same stream of M1, without computing it again). An extension of this operator applies M1 more than twice, in given positions. We denote this extended operator by 'M1[M2,M3]' (or more generally 'M1[M2,M3,...,Mn]'). It applies M1 on the plaintext (e.g., XORs the stream to the plaintext), then applies M2, then applies M1 again, then applies M3, and then M1 again. This mode actually applies $2n-1$ modes (five in the case of M1[M2,M3]) but takes the time of computing only n (three) modes. Our aim in defining such modes is to get modes which are more secure than standard n-modes, but have the same encryption speed as n-modes. However, we expect that these modes are less secure than $(2n-1)$-modes.

4.2 Secure Triple Modes Combined in Non-Standard Ways

We conjecture that the OFB$\rightarrow$CBC$\rightarrow$CBC and the OFB$\rightarrow$CFB$\rightarrow$CFB modes are secure modes, useful for applications which require a security-extended stream (OFB-like) mode. We conjecture that the modes OFB[CBC,CBC^{-1}], OFB[CFB,CFB^{-1}], OFB[CBC,CBC], and OFB[CFB,CFB] are secure chaining/feedback modes. The complexity of attacking these modes is conjectured to be at least 2^{112}.

[3] An extension of this definition allows M1 to be any mode, where a fixed plaintext block is fed to the mode M1, and the result becomes the output stream. This extension is not very useful when applied to standard DES modes, since the CBC and the CFB modes generate an OFB stream when fed by the zero plaintext blocks.

4.3 Quadruple Modes

We conjecture that the CBC|CBC|CBC^{-1}|CBC^{-1}, the CFB|CFB|CFB^{-1}|CFB^{-1}, and the the OFB[CBC,CBC,CBC^{-1}] modes are more secure than any triple mode. We also believe that the OFB[CBC,CBC,CBC] and the OFB[CFB,CFB,CFB] modes are secure[4], and whose complexities of attack are at least 2^{128}.

5 Summary

In this paper we showed that all the double modes and all the triple modes of DES (possibly except the triple-ECB mode, if dictionary attacks which cannot find the keys are not considered), are much less secure than might be expected, and whose theoretical strength is comparable to a single encryption. Almost all the modes are vulnerable to known plaintext, chosen plaintext, or chosen ciphertext attacks, with only one stream of encryption. Only a few of the modes require to know the initial values or to get two streams with the same keys and the same initial values. We also suggested several candidates for more secure modes.

Acknowledgments

We are grateful to Jim Massey, Ross Anderson, Kenneth Paterson, Ronny Roth and Adi Shamir for many helpful discussions and suggestions.

References

[1] Thomas A. Berson, *Long Key Variants of DES*, Advances in Cryptology, proceedings of CRYPTO'82, pp. 311–313, 1982.

[2] Eli Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Cryptology, Vol. 7, No. 4, pp. 229–246, 1994.

[3] Eli Biham, *Cryptanalysis of Multiple Modes of Operation*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of ASIACRYPT'94, pp. 278–292, 1994.

[4] Eli Biham, Alex Biryukov, *An Improvement of Davies' Attack on DES*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'94, pp. 461–467, 1994.

[4]The mode ECB|ECB|CBC|ECB which was mentioned in preliminary versions of this paper is less secure, and can be attacked using 2^{35} chosen ciphertexts and 2^{58} steps.

[5] Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[6] Richard E. Blahut, *Theory and Practice of Error Control Codes*, Addison Wesley, 1983.

[7] Lawrence Brown, Jennifer Seberry, *Key Scheduling in DES Type Cryptosystems*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of AUSCRYPT'90, pp. 221–228, 1990.

[8] Don Coppersmith, private communications.

[9] Donald W. Davies, Graeme I. P. Parkin, *The average Cycle Size of the Key Stream in Output Feedback Encipherment*, Advances in Cryptology, proceedings of CRYPTO'82, pp. 97–98, 1982.

[10] D. W. Davies, W. L. Price, *Security for Computer Networks*, second edition, John Willey & Sons, 1989.

[11] W. Diffie, M. E. Hellman, *Exhaustive Cryptanalysis of the NBS Data Encryption Standard*, Computer, Vol. 10, No. 6, pp. 74–84, June 1977.

[12] Carl Ellison, *Two Symmetric DES modes*, presented at the RUMP session of CRYPTO'95, 1995.

[13] Burt Kaliski, *Triple-DES: A Brief Report*, RSA laboratories, private communication, October 29, 1993.

[14] Kwangjo Kim, Sangjun Park, Sangjin Lee, *Reconstruction of s^2DES S-boxes and their Immunity to Differential Cryptanalysis*, proceedings of JW-ISC93 – Korea-Japan Joint Workshop on Information Security and Cryptology, Seoul, Korea, October 24–26, 1993.

[15] James L. Massey, *Shift-Register Synthesis and BCH Decoding*, IEEE Transactions on Information Theory, Vol. 15, No. 1, pp. 122–127, January 1969.

[16] Mitsuru Matsui, *Linear Cryptanalysis Method for DES Cipher*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'93, pp. 386–397, 1993.

[17] National Bureau of Standards, *Data Encryption Standard*, U.S. Department of Commerce, FIPS pub. 46, January 1977.

[18] National Bureau of Standards, *DES Modes of Operation*, U.S. Department of Commerce, FIPS pub. 81, December 1980.

[19] Michael 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, August 1993.

A Appendix

In this appendix we list all the single, double and triple modes, together with their attacking techniques and complexity of cryptanalysis.

In the *Attack* field, we describe the type of attack and hints about the used techniques. For some of the modes, we describe more than one technique for the attack. In such a case, the first technique finds the key of one single mode, while later, the other technique is used to find the key of the other mode(s). In this field we use the following notations:

KPA Known plaintext attack.

CPA Chosen plaintext attack.

CCA Chosen ciphertext attack.

Ad Adaptive attacks.

KIV Attacks in which the initial values are known (see Section 3.1).

2SKIV Attacks in which the attacker chooses two streams, that are both encrypted under the same key and the *same known* initial value (see Section 3.1).

Ex Exhaustive search for the key of a single mode (Technique E in [3]).

Bi Birthday technique (Technique F in [3]).

Col Like 'Bi', however the collection of pairs is exhaustive rather than using the birthday paradox.

G Technique G.

H Technique H.

I Technique I.

MM Meet in the middle, using large tables (similar to the attack on double DES[11, page 83]).

(number) Some more details on the attack are elaborated in item (number) in section 3.

(*) Some more details on the attack are elaborated in section 3.1.

The complexity field describes three complexity parameters: the number of plaintexts required/the number of steps of the attack/the required memory size. These complexities are measured for DES as the underlying blockcipher. In these complexities, all the (up to three) keys and all the (up to three) initial values are found. Note that the memory size is given only when the amount of memory required for tables is non-negligible. The required memory for keeping the plaintexts and the ciphertexts is not counted.

Note that in this appendix we only give hints for one (or sometimes two) attacks on each mode. It is not the purpose of this appendix to give the best attacks on these modes, nor the most efficient or the easiest to implement attacks. In particular, some modes which are attacked with a chosen ciphertext attacks in this appendix can also be attacked with a chosen plaintext attack, and sometimes even with a known plaintext attack.

Single Modes

No.	Mode	Attack	Complexity	Data	Inverse
1	ECB	KPA/Ex	$1/2^{56}/-$		1
2	CBC	KPA/Ex	$2/2^{56}/-$		3
3	CBC^{-1}	KPA/Ex	$2/2^{56}/-$		2
4	OFB	KPA/Ex	$2/2^{56}/-$		4
5	CFB	KPA/Ex	$2/2^{56}/-$		6
6	CFB^{-1}	KPA/Ex	$2/2^{56}/-$		5

Double Modes

No.	Mode	Attack	Complexity	Data	Inverse
7	ECB|ECB	KPA/MM	$2/2^{57}/2^{56}$		7
8	ECB|CBC	KPA/MM (or CCA)	$3/2^{57}/2^{56}$ (or $3/2^{58}/-$)	(BAA)	19
9	ECB|CBC^{-1}	CPA/Ex	$3/2^{58}/-$	BAA	13
10	ECB|OFB	CPA/Ex	$2^{64}/2^{58}/-$		25
11	ECB|CFB	KPA/MM (or CCA)	$3/2^{57}/2^{56}$ (or $3/2^{58}/-$)	(AAB)	37
12	ECB|CFB^{-1}	CPA/Ex	$3/2^{58}/-$	AAB	31
13	CBC|ECB*	CCA/Ex (or CPA/Ex)	$3/2^{58}/-$ (or $2^{64}/2^{58}/-$)	BAA	9
14	CBC|CBC	CCA/Bi	$2^{34}/2^{59}/2^{33}$		21
15	CBC|CBC^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/2^{66}$		15
16	CBC|OFB	CPA/I or CCA/G	$2^{65}/2^{65}/2^{65}$ or $2^{66}/2^{66}/-$		27
17	CBC|CFB	CCA/Ex	$4/2^{58}/-$	BAAA	39
18	CBC|CFB^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/-$		33
19	CBC^{-1}|ECB	KPA/MM (or CPA)	$3/2^{57}/2^{56}$ (or $3/2^{58}/-$)	(BAA)	8
20	CBC^{-1}|CBC	KPA/MM (or CPA)	$3/2^{57}/2^{56}$ (or $3/2^{58}/-$)	(AAA)	20
21	CBC^{-1}|CBC^{-1}	CPA/Bi	$2^{34}/2^{59}/2^{33}$		14
22	CBC^{-1}|OFB	CPA/Ex	$2^{64}/2^{58}/-$		26
23	CBC^{-1}|CFB	KPA/MM (or CPA)	$3/2^{57}/2^{56}$ or $3/2^{58}/-$	(AAA)	38
24	CBC^{-1}|CFB^{-1}	CPA/Ex	$4/2^{58}/-$	AAAB	32
25	OFB|ECB	CPA/Ex or CCA/Ex	$2^{64}/2^{58}/-$		10
26	OFB|CBC	CCA/Ex	$2^{64}/2^{58}/-$		22
27	OFB|CBC^{-1}	CPA/G or CCA/I	$2^{65}/2^{66}/-$ or $2^{65}/2^{65}/2^{65}$		16
28	OFB|OFB	KPA/I	$2^{65}/2^{65}/2^{64}$		28
29	OFB|CFB	CCA/Ex	$2^{64}/2^{58}/-$	BAA...A	40
30	OFB|CFB^{-1}	CPA/G or CCA/I	$2^{66}/2^{66}/-$ or $2^{65}/2^{65}/2^{65}$		34
31	CFB|ECB	CCA/Ex	$3/2^{58}/-$	AAB	12
32	CFB|CBC	CCA/Ex	$4/2^{58}/-$	AAAB	24
33	CFB|CBC^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/-$		18
34	CFB|OFB	CPA/I or CCA/G	$2^{65}/2^{65}/2^{65}$ or $2^{66}/2^{66}/-$		30
35	CFB|CFB	CCA/Bi	$2^{34}/2^{59}/2^{33}$		42
36	CFB|CFB^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/2^{66}$		36
37	CFB^{-1}|ECB	KPA/MM (or CPA)	$3/2^{57}/2^{56}$ (or $3/2^{58}/-$)	(AAB)	11

38	CFB^{-1}|CBC	KPA/MM (or CPA)	$3/2^{57}/2^{56}$ (or $3/2^{58}/-$)	(AAA)	23
39	CFB^{-1}|CBC^{-1}	CPA/Ex	$4/2^{58}/-$	BAAA	17
40	CFB^{-1}|OFB	CPA/Ex	$2^{64}/2^{58}/-$	BAA...A	29
41	CFB^{-1}|CFB	KPA/MM (or CPA)	$3/2^{57}/2^{56}$ (or $3/2^{58}/-$)	(AAA)	41
42	CFB^{-1}|CFB^{-1}	CPA/Bi	$2^{34}/2^{59}/2^{33}$		35

* Davies and Price suggest this mode in [10].

Triple Modes

No.	Mode	Attack	Complexity	Data	Inverse
43	ECB|ECB|ECB	The best known attack is KPA/MM whose complexity is $3/2^{113}/2^{56}$. Dictionary attacks (which cannot recover the keys) require only 2^{64} known plaintexts.			43
44	ECB|ECB|CBC	CCA/Bi/MM	$2^{33}/2^{58}/2^{56}$		115
45	ECB|ECB|CBC^{-1}	CPA/G/MM	$2^{64}/2^{58}/2^{56}$		79
46	ECB|ECB|OFB	CPA/G/MM	$2^{64}/2^{58}/2^{56}$		151
47	ECB|ECB|CFB	CCA/Bi/MM	$2^{33}/2^{58}/2^{56}$		223
48	ECB|ECB|CFB^{-1}	CPA/G/MM	$2^{64}/2^{58}/2^{56}$		187
49	ECB|CBC|ECB	CCA/Ex/Ex/Ex(1)	$5/2^{59}/-$	AABBA	55
50	ECB|CBC|CBC	CCA/Bi/Bi/Ex	$2^{33}/2^{59}/2^{33}$		127
51	ECB|CBC|CBC^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/-$		91
52	ECB|CBC|OFB	CPA/I/Ex or CCA/Ex/G	$2^{65}/2^{65}/2^{65}$ or $2^{64}/2^{58}/-$		163
53	ECB|CBC|CFB	CCA/Bi	$2^{34}/2^{59}/2^{33}$		235
54	ECB|CBC|CFB^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/-$		199
55	ECB|CBC^{-1}|ECB	CPA/Ex/Ex/Ex(1)	$5/2^{59}/-$	AABBA	49
56	ECB|CBC^{-1}|CBC	CPA/Ex/Ex	$5/2^{59}/-$	AABBA	121
57	ECB|CBC^{-1}|CBC^{-1}	CPA/Ex/Bi	$2^{34}/2^{59}/2^{33}$		85
58	ECB|CBC^{-1}|OFB	CPA/Ex/Ex	$2^{64}/2^{58}/-$		157
59	ECB|CBC^{-1}|CFB	CPA/Ex/Ex	$5/2^{59}/-$	AABBA	229
60	ECB|CBC^{-1}|CFB^{-1}	CPA/Bi(6)	$2^{36}/2^{59}/2^{33}$		193
61	ECB|OFB|ECB	CPA/MM or CCA/MM	$2^{64}/2^{58}/2^{56}$		61
62	ECB|OFB|CBC	CCA/MM	$2^{64}/2^{58}/2^{56}$		133
63	ECB|OFB|CBC^{-1}	CPA/Ex/G	$2^{65}/2^{65}/-$		97
64	ECB|OFB|OFB	CPA/I	$2^{65}/2^{65}/2^{65}$		169
65	ECB|OFB|CFB	CCA/MM	$2^{64}/2^{58}/2^{56}$		241
66	ECB|OFB|CFB^{-1}	CPA/Ex/G	$2^{65}/2^{65}/-$		205
67	ECB|CFB|ECB	CCA/Ex/Ex/Ex	$5/2^{59}/-$	AABBA	73
68	ECB|CFB|CBC	CCA/Bi	$2^{34}/2^{59}/2^{33}$		145
69	ECB|CFB|CBC^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/-$		109
70	ECB|CFB|OFB	CPA/I or CCA/Ex/H	$2^{65}/2^{65}/2^{65}$ or $2^{65}/2^{65}/-$		181
71	ECB|CFB|CFB	CCA/Bi	$2^{34}/2^{59}/2^{33}$		253
72	ECB|CFB|CFB^{-1}	CPA/Ex/H or CCA/Ex/H	$2^{66}/2^{66}/2^{66}$		217
73	ECB|CFB^{-1}|ECB	CPA/Ex/Ex/Ex	$5/2^{59}/-$	AABBA	67
74	ECB|CFB^{-1}|CBC	CPA/Ex/Ex/Ex(2)	$4/2^{58}/-$	AAAB	139
75	ECB|CFB^{-1}|CBC^{-1}	CPA/Bi(6)	$2^{36}/2^{59}/2^{33}$		103
76	ECB|CFB^{-1}|OFB	CPA/Ex	$2^{64}/2^{58}/-$		175
77	ECB|CFB^{-1}|CFB	CPA/Ex/Ex/Ex	$4/2^{58}/-$		247
78	ECB|CFB^{-1}|CFB^{-1}	CPA/Ex/Bi	$2^{34}/2^{59}/2^{33}$		211
79	CBC|ECB|ECB	CCA/G/MM	$2^{64}/2^{58}/2^{56}$		45
80	CBC|ECB|CBC	CCA/Bi	$2^{34}/2^{59}/2^{33}$		117
81	CBC|ECB|CBC^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/-$		81
82	CBC|ECB|OFB	CPA/I	$2^{65}/2^{65}/2^{65}$		153
83	CBC|ECB|CFB	CCA/Bi/Ex	$2^{34}/2^{59}/2^{33}$		225
84	CBC|ECB|CFB^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/-$		189
85	CBC|CBC|ECB	CCA/Ex/Bi	$2^{34}/2^{59}/2^{33}$		57
86	CBC|CBC|CBC	CCA/Bi	$2^{34}/2^{60}/2^{33}$		129
87	CBC|CBC|CBC^{-1}	CCA/H(9) [Ad or KIV or K_3 only]	$2^{68}/2^{66}/-$		93
88	CBC|CBC|OFB	KIV-CCA(*)	$2^{64}/2^{59}/-$		165
89	CBC|CBC|CFB	CCA/Bi	$2^{34}/2^{60}/2^{33}$		237
90	CBC|CBC|CFB^{-1}	CCA/H(9) [Ad or KIV or K_3 only]	$2^{68}/2^{66}/-$		201
91	CBC|CBC^{-1}|ECB	CPA/H or CCA/H	$2^{68}/2^{66}/-$		51
92	CBC|CBC^{-1}|CBC	CCA/Bi(8)	$2^{66}/2^{58}/2^{33}$		123
93	CBC|CBC^{-1}|CBC^{-1}	CPA/H(9) [Ad or KIV or K_1 only]	$2^{68}/2^{66}/-$		87
94	CBC|CBC^{-1}|OFB	CPA/I/H	$2^{66}/2^{66}/2^{66}$		159
95	CBC|CBC^{-1}|CFB	CCA/Bi(8)	$2^{66}/2^{58}/2^{33}$		231

96	CBC|CBC^{-1}|CFB^{-1}	CPA/H(9)	$2^{68}/2^{58}/-$		195
97	CBC|OFB|ECB	CCA/Ex/G	$2^{65}/2^{65}/-$		63
98	CBC|OFB|CBC	CCA/Col/G(5)	$2^{66}/2^{66}/-$		135
99	CBC|OFB|CBC^{-1}	2SKIV-KPA/MM(*)	$4/2^{58}/2^{56}$		99
100	CBC|OFB|OFB	CPA/J	$2^{67}/2^{75}/2^{66}$		171
101	CBC|OFB|CFB	CCA/Col/G	$2^{66}/2^{66}/-$		243
102	CBC|OFB|CFB^{-1}	2SKIV-KPA/MM(*)	$4/2^{58}/2^{56}$		207
103	CBC|CFB|ECB	CCA/Bi(6)	$2^{36}/2^{59}/2^{33}$		75
104	CBC|CFB|CBC	CCA/Bi	$2^{34}/2^{60}/2^{33}$		147
105	CBC|CFB|CBC^{-1}	CCA/H(9)	$2^{68}/2^{58}/-$		111
106	CBC|CFB|OFB	CCA/G(7)	$2^{65}/2^{60}/-$		183
107	CBC|CFB|CFB	CCA/Bi	$2^{34}/2^{60}/2^{33}$		255
108	CBC|CFB|CFB^{-1}	CCA/H(9)	$2^{68}/2^{58}/-$		219
109	CBC|CFB^{-1}|ECB	CPA/H or CCA/H	$2^{68}/2^{66}/-$		69
110	CBC|CFB^{-1}|CBC	CCA/Bi(8)	$2^{68}/2^{58}/2^{33}$		141
111	CBC|CFB^{-1}|CBC^{-1}	CPA/H(9)	$2^{68}/2^{58}/-$		105
112	CBC|CFB^{-1}|OFB	CPA/I/H	$2^{68}/2^{66}/2^{66}$		177
113	CBC|CFB^{-1}|CFB	CCA/Bi(8)	$2^{68}/2^{58}/2^{33}$		249
114	CBC|CFB^{-1}|CFB^{-1}	CPA/H(9) [Ad or KIV or K_1 only]	$2^{68}/2^{66}/-$		213
115	CBC^{-1}|ECB|ECB	CPA/Bi/MM	$2^{33}/2^{58}/2^{56}$		44
116	CBC^{-1}|ECB|CBC	CPA/Ex/Ex(3) or CCA/Ex/Ex(3)	$4/2^{58}/-$	BAAA	116
117	CBC^{-1}|ECB|CBC^{-1}	CPA/Bi	$2^{34}/2^{59}/2^{33}$		80
118	CBC^{-1}|ECB|OFB	CPA/G/Ex	$2^{64}/2^{58}/-$		152
119	CBC^{-1}|ECB|CFB	CPA/Ex or CCA/Ex	$4/2^{58}/-$	BAAA or AAAB	224
120	CBC^{-1}|ECB|CFB^{-1}	CPA/Bi/Ex	$2^{34}/2^{59}/2^{33}$		188
121	CBC^{-1}|CBC|ECB	CCA/Ex/Ex	$5/2^{59}/-$	AABBA	56
122	CBC^{-1}|CBC|CBC	CCA/Ex/Bi	$2^{34}/2^{59}/2^{33}$	AAAA+...	128
123	CBC^{-1}|CBC|CBC^{-1}	CPA/Bi(8)	$2^{66}/2^{58}/2^{33}$		92
124	CBC^{-1}|CBC|OFB	CPA/I/Ex	$2^{65}/2^{65}/2^{65}$		164
125	CBC^{-1}|CBC|CFB	CCA/Ex/Ex	$5/2^{58}/-$	BAAAA	236
126	CBC^{-1}|CBC|CFB^{-1}	CPA/Bi(8)	$2^{68}/2^{58}/2^{33}$		200
127	CBC^{-1}|CBC^{-1}|ECB	CPA/Bi/Bi/Ex	$2^{33}/2^{59}/2^{33}$		50
128	CBC^{-1}|CBC^{-1}|CBC	CPA/Ex/Bi	$2^{34}/2^{59}/2^{33}$	AAAA+...	122
129	CBC^{-1}|CBC^{-1}|CBC^{-1}	CPA/Bi	$2^{34}/2^{60}/2^{33}$		86
130	CBC^{-1}|CBC^{-1}|OFB	CPA/Col	$2^{66}/2^{59}/-$		158
131	CBC^{-1}|CBC^{-1}|CFB	CPA/Ex/Bi	$2^{34}/2^{59}/2^{33}$	AAAA+...	230
132	CBC^{-1}|CBC^{-1}|CFB^{-1}	CPA/Bi	$2^{34}/2^{60}/2^{33}$		194
133	CBC^{-1}|OFB|ECB	CPA/MM	$2^{64}/2^{58}/2^{56}$		62
134	CBC^{-1}|OFB|CBC	CPA/(4) or CCA/(4)	$2^{66}/2^{58}/-$		134
135	CBC^{-1}|OFB|CBC^{-1}	CPA/Col/G(5)	$2^{66}/2^{66}/-$		98
136	CBC^{-1}|OFB|OFB	CPA/I	$2^{65}/2^{65}/2^{65}$		170
137	CBC^{-1}|OFB|CFB	CPA/(4) or CCA/(4)	$2^{66}/2^{58}/-$		242
138	CBC^{-1}|OFB|CFB^{-1}	CPA/Col/G	$2^{66}/2^{66}/-$		206
139	CBC^{-1}|CFB|ECB	CCA/Ex/Ex/Ex(2)	$4/2^{58}/-$	AAAB	74
140	CBC^{-1}|CFB|CBC	CCA/Ex/Ex	$5/2^{58}/-$	AAAAB	146
141	CBC^{-1}|CFB|CBC^{-1}	CPA/Bi(8)	$2^{68}/2^{58}/2^{33}$		110
142	CBC^{-1}|CFB|OFB	CPA/I/Ex	$2^{65}/2^{65}/2^{65}$		182
143	CBC^{-1}|CFB|CFB	CCA/Ex/Bi	$2^{34}/2^{59}/2^{33}$	AAAA+...	254
144	CBC^{-1}|CFB|CFB^{-1}	CPA/Bi(8)	$2^{66}/2^{58}/2^{33}$		218
145	CBC^{-1}|CFB^{-1}|ECB	CPA/Bi	$2^{34}/2^{59}/2^{33}$		68
146	CBC^{-1}|CFB^{-1}|CBC	CPA/Ex/Ex	$5/2^{58}/-$	AAAAB	140
147	CBC^{-1}|CFB^{-1}|CBC^{-1}	CPA/Bi	$2^{34}/2^{60}/2^{33}$		104
148	CBC^{-1}|CFB^{-1}|OFB	CPA/Col	$2^{66}/2^{59}/-$		176
149	CBC^{-1}|CFB^{-1}|CFB	CPA/Ex/Ex	$5/2^{58}/-$	AAAAB	248
150	CBC^{-1}|CFB^{-1}|CFB^{-1}	CPA/Bi	$2^{34}/2^{60}/2^{33}$		212
151	OFB|ECB|ECB	CCA/G/MM	$2^{64}/2^{58}/2^{56}$		46
152	OFB|ECB|CBC	CCA/G/Ex	$2^{64}/2^{58}/-$		118
153	OFB|ECB|CBC^{-1}	CCA/I	$2^{65}/2^{65}/2^{65}$		82
154	OFB|ECB|OFB	CPA/I or CCA/I	$2^{65}/2^{65}/2^{64}$		154
155	OFB|ECB|CFB	CCA/G/Ex	$2^{64}/2^{58}/-$		226
156	OFB|ECB|CFB^{-1}	CCA/I	$2^{65}/2^{65}/2^{65}$		190
157	OFB|CBC|ECB	CCA/Ex/Ex	$2^{64}/2^{58}/-$		58
158	OFB|CBC|CBC	CCA/Col	$2^{66}/2^{59}/-$		130
159	OFB|CBC|CBC^{-1}	CCA/I/H	$2^{66}/2^{66}/2^{66}$		94
160	OFB|CBC|OFB	CCA/I/G	$2^{66}/2^{66}/2^{66}$		166
161	OFB|CBC|CFB	CCA/Col	$2^{66}/2^{59}/-$		238
162	OFB|CBC|CFB^{-1}	CCA/I/H	$2^{66}/2^{66}/2^{66}$		202

163	OFB$\vert$CBC^{-1}$\vert$ECB	CPA/Ex/G or CCA/I/Ex	$2^{65}/2^{58}/-$ or $2^{65}/2^{65}/2^{65}$		52
164	OFB$\vert$CBC^{-1}$\vert$CBC	CCA/I/Ex	$2^{65}/2^{65}/2^{65}$		124
165	OFB$\vert$CBC^{-1}$\vert$CBC^{-1}	KIV-CPA(*)	$2^{64}/2^{59}/-$		88
166	OFB$\vert$CBC^{-1}$\vert$OFB	CPA/I/G	$2^{66}/2^{66}/2^{66}$		160
167	OFB$\vert$CBC^{-1}$\vert$CFB	CCA/I/Ex	$2^{65}/2^{65}/2^{65}$		232
168	OFB$\vert$CBC^{-1}$\vert$CFB^{-1}	CPA/G(7)	$2^{65}/2^{60}/-$		196
169	OFB$\vert$OFB$\vert$ECB	CCA/I	$2^{65}/2^{65}/2^{65}$		64
170	OFB$\vert$OFB$\vert$CBC	CCA/I	$2^{65}/2^{65}/2^{65}$		136
171	OFB$\vert$OFB$\vert$CBC^{-1}	CCA/J	$2^{67}/2^{75}/2^{66}$		100
172	OFB$\vert$OFB$\vert$OFB	KPA/J	$2^{67}/2^{75}/2^{66}$		172
173	OFB$\vert$OFB$\vert$CFB	CCA/I	$2^{65}/2^{65}/2^{65}$		244
174	OFB$\vert$OFB$\vert$CFB^{-1}	CCA/J	$2^{67}/2^{75}/2^{66}$		208
175	OFB$\vert$CFB$\vert$ECB	CCA/Ex	$2^{64}/2^{58}/-$		76
176	OFB$\vert$CFB$\vert$CBC	CCA/Col	$2^{66}/2^{59}/-$		148
177	OFB$\vert$CFB$\vert$CBC^{-1}	CCA/I/H	$2^{66}/2^{66}/2^{66}$		112
178	OFB$\vert$CFB$\vert$OFB	CCA/I/G	$2^{66}/2^{66}/2^{66}$		184
179	OFB$\vert$CFB$\vert$CFB	CCA/Col	$2^{66}/2^{59}/-$		256
180	OFB$\vert$CFB$\vert$CFB^{-1}	CCA/I/H	$2^{66}/2^{66}/2^{66}$		220
181	OFB$\vert$CFB^{-1}$\vert$ECB	CPA/Ex/H or CCA/I	$2^{65}/2^{65}/-$ or $2^{65}/2^{65}/2^{65}$		70
182	OFB$\vert$CFB^{-1}$\vert$CBC	CCA/I/Ex	$2^{65}/2^{65}/2^{65}$		142
183	OFB$\vert$CFB^{-1}$\vert$CBC^{-1}	CPA/G(7)	$2^{65}/2^{60}/-$		106
184	OFB$\vert$CFB^{-1}$\vert$OFB	CPA/I/G	$2^{66}/2^{66}/2^{66}$		178
185	OFB$\vert$CFB^{-1}$\vert$CFB	CCA/I/Ex	$2^{65}/2^{65}/2^{65}$		250
186	OFB$\vert$CFB^{-1}$\vert$CFB^{-1}	KIV-CPA(*)	$2^{64}/2^{59}/-$		214
187	CFB$\vert$ECB$\vert$ECB	CCA/G/MM	$2^{64}/2^{58}/2^{56}$		48
188	CFB$\vert$ECB$\vert$CBC	CCA/Bi/Ex	$2^{34}/2^{58}/2^{33}$		120
189	CFB$\vert$ECB$\vert$CBC^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/-$		84
190	CFB$\vert$ECB$\vert$OFB	CPA/I	$2^{65}/2^{65}/2^{65}$		156
191	CFB$\vert$ECB$\vert$CFB	CCA/Bi/Ex	$2^{34}/2^{59}/2^{33}$		228
192	CFB$\vert$ECB$\vert$CFB^{-1}	CPA/H or CCA/H	$2^{68}/2^{66}/-$		192
193	CFB$\vert$CBC$\vert$ECB	CCA/Bi(6)	$2^{36}/2^{59}/2^{33}$		60
194	CFB$\vert$CBC$\vert$CBC	CCA/Bi	$2^{34}/2^{60}/2^{33}$		132
195	CFB$\vert$CBC$\vert$CBC^{-1}	CCA/H(9)	$2^{68}/2^{58}/-$		96
196	CFB$\vert$CBC$\vert$OFB	CCA/G(7)	$2^{65}/2^{60}/-$		168
197	CFB$\vert$CBC$\vert$CFB	CCA/Bi	$2^{34}/2^{60}/2^{33}$		240
198	CFB$\vert$CBC$\vert$CFB^{-1}	CCA/H(9)	$2^{68}/2^{58}/-$		204
199	CFB$\vert$CBC^{-1}$\vert$ECB	CPA/H or CCA/H	$2^{68}/2^{66}/-$		54
200	CFB$\vert$CBC^{-1}$\vert$CBC	CCA/Bi(8)	$2^{68}/2^{58}/2^{33}$		126
201	CFB$\vert$CBC^{-1}$\vert$CBC^{-1}	CPA/H(9) [Ad or KIV or K_1 only]	$2^{68}/2^{66}/-$		90
202	CFB$\vert$CBC^{-1}$\vert$OFB	CPA/I/H	$2^{66}/2^{66}/2^{66}$		162
203	CFB$\vert$CBC^{-1}$\vert$CFB	CCA/Bi(8)	$2^{68}/2^{58}/2^{33}$		234
204	CFB$\vert$CBC^{-1}$\vert$CFB^{-1}	CPA/H(9)	$2^{68}/2^{58}/-$		198
205	CFB$\vert$OFB$\vert$ECB	CCA/Ex/G	$2^{65}/2^{65}/-$		66
206	CFB$\vert$OFB$\vert$CBC	CCA/Col/G	$2^{66}/2^{66}/-$		138
207	CFB$\vert$OFB$\vert$CBC^{-1}	2SKIV-KPA/MM(*)	$2/2^{58}/2^{56}$		102
208	CFB$\vert$OFB$\vert$OFB	CPA/J	$2^{67}/2^{75}/2^{66}$		174
209	CFB$\vert$OFB$\vert$CFB	CCA/Col/G	$2^{66}/2^{66}/-$		246
210	CFB$\vert$OFB$\vert$CFB^{-1}	2SKIV-KPA/MM(*)	$2/2^{58}/2^{56}$		210
211	CFB$\vert$CFB$\vert$ECB	CCA/Ex/Bi	$2^{34}/2^{59}/2^{33}$		78
212	CFB$\vert$CFB$\vert$CBC	CCA/Bi	$2^{34}/2^{60}/2^{33}$		150
213	CFB$\vert$CFB$\vert$CBC^{-1}	CCA/H(9) [Ad or KIV or K_3 only]	$2^{68}/2^{66}/-$		114
214	CFB$\vert$CFB$\vert$OFB	KIV-CCA(*)	$2^{64}/2^{59}/-$		186
215	CFB$\vert$CFB$\vert$CFB	CCA/Bi	$2^{34}/2^{60}/2^{33}$		258
216	CFB$\vert$CFB$\vert$CFB^{-1}	CCA/H(9) [Ad or KIV or K_3 only]	$2^{68}/2^{66}/-$		222
217	CFB$\vert$CFB^{-1}$\vert$ECB	CPA/Ex/H or CCA/Ex/H	$2^{66}/2^{66}/2^{66}$		72
218	CFB$\vert$CFB^{-1}$\vert$CBC	CCA/Bi(8)	$2^{66}/2^{58}/2^{33}$		144
219	CFB$\vert$CFB^{-1}$\vert$CBC^{-1}	CPA/H(9)	$2^{68}/2^{58}/-$		108
220	CFB$\vert$CFB^{-1}$\vert$OFB	CPA/I/H	$2^{68}/2^{66}/2^{66}$		180
221	CFB$\vert$CFB^{-1}$\vert$CFB	CCA/Bi(8)	$2^{66}/2^{58}/2^{33}$		252
222	CFB$\vert$CFB^{-1}$\vert$CFB^{-1}	CPA/H(9) [Ad or KIV or K_1 only]	$2^{68}/2^{66}/-$		216
223	CFB^{-1}$\vert$ECB$\vert$ECB	CPA/Bi/MM	$2^{33}/2^{58}/2^{56}$		47
224	CFB^{-1}$\vert$ECB$\vert$CBC	CPA/Ex/Ex or CCA/Ex/Ex	$4/2^{58}/-$	BAAA or AAAB	119
225	CFB^{-1}$\vert$ECB$\vert$CBC^{-1}	CPA/Bi/Ex	$2^{34}/2^{59}/2^{33}$		83
226	CFB^{-1}$\vert$ECB$\vert$OFB	CPA/G/Ex	$2^{64}/2^{58}/-$		155
227	CFB^{-1}$\vert$ECB$\vert$CFB	CPA/Ex/Ex or CCA/Ex/Ex	$4/2^{58}/-$	AAAB	227
228	CFB^{-1}$\vert$ECB$\vert$CFB^{-1}	CPA/Bi/Ex	$2^{34}/2^{59}/2^{33}$		191
229	CFB^{-1}$\vert$CBC$\vert$ECB	CCA/Ex/Ex	$5/2^{59}/-$	AABBA	59

230	CFB^{-1}|CBC|CBC	CCA/Ex/Bi	$2^{34}/2^{59}/2^{33}$	AAAA+...	131
231	CFB^{-1}|CBC|CBC^{-1}	CPA/Bi(8)	$2^{66}/2^{58}/2^{33}$		95
232	CFB^{-1}|CBC|OFB	CPA/I/Ex	$2^{65}/2^{65}/2^{65}$		167
233	CFB^{-1}|CBC|CFB	CCA/Ex/Ex	$5/2^{58}/-$	BAAAA	239
234	CFB^{-1}|CBC|CFB^{-1}	CPA/Bi(8)	$2^{68}/2^{58}/2^{33}$		203
235	CFB^{-1}|CBC^{-1}|ECB	CPA/Bi	$2^{34}/2^{59}/2^{33}$		53
236	CFB^{-1}|CBC^{-1}|CBC	CPA/Ex/Ex	$5/2^{58}/-$	BAAAA	125
237	CFB^{-1}|CBC^{-1}|CBC^{-1}	CPA/Bi	$2^{34}/2^{60}/2^{33}$		89
238	CFB^{-1}|CBC^{-1}|OFB	CPA/Col	$2^{66}/2^{59}/-$		161
239	CFB^{-1}|CBC^{-1}|CFB	CPA/Ex/Ex	$5/2^{58}/-$	BAAAA	233
240	CFB^{-1}|CBC^{-1}|CFB^{-1}	CPA/Bi	$2^{34}/2^{60}/2^{33}$		197
241	CFB^{-1}|OFB|ECB	CPA/MM	$2^{64}/2^{58}/2^{56}$		65
242	CFB^{-1}|OFB|CBC	CPA/(4) or CCA/(4)	$2^{66}/2^{58}/-$		137
243	CFB^{-1}|OFB|CBC^{-1}	CPA/Col/G	$2^{66}/2^{66}/-$		101
244	CFB^{-1}|OFB|OFB	CPA/I	$2^{65}/2^{65}/2^{65}$		173
245	CFB^{-1}|OFB|CFB	CPA/(4) or CCA/(4)	$2^{66}/2^{58}/-$		245
246	CFB^{-1}|OFB|CFB^{-1}	CPA/Col/G	$2^{66}/2^{66}/-$		209
247	CFB^{-1}|CFB|ECB	CCA/Ex/Ex/Ex	$4/2^{58}/-$	AAAB	77
248	CFB^{-1}|CFB|CBC	CCA/Ex/Ex	$5/2^{58}/-$	AAAAB	149
249	CFB^{-1}|CFB|CBC^{-1}	CPA/Bi(8)	$2^{68}/2^{58}/2^{33}$		113
250	CFB^{-1}|CFB|OFB	CPA/I/Ex	$2^{65}/2^{65}/2^{65}$		185
251	CFB^{-1}|CFB|CFB	CCA/Ex/Bi	$2^{34}/2^{59}/2^{33}$	AAAA+...	257
252	CFB^{-1}|CFB|CFB^{-1}	CPA/Bi(8)	$2^{66}/2^{58}/2^{33}$		221
253	CFB^{-1}|CFB^{-1}|ECB	CPA/Bi	$2^{34}/2^{59}/2^{33}$		71
254	CFB^{-1}|CFB^{-1}|CBC	CPA/Ex/Bi	$2^{34}/2^{59}/2^{33}$	AAAA+...	143
255	CFB^{-1}|CFB^{-1}|CBC^{-1}	CPA/Bi	$2^{34}/2^{60}/2^{33}$		107
256	CFB^{-1}|CFB^{-1}|OFB	CPA/Col	$2^{66}/2^{59}/-$		179
257	CFB^{-1}|CFB^{-1}|CFB	CPA/Ex/Bi	$2^{34}/2^{59}/2^{33}$	AAAA+...	251
258	CFB^{-1}|CFB^{-1}|CFB^{-1}	CPA/Bi	$2^{34}/2^{60}/2^{33}$		215