The Data Encryption Standard (DES)

The Data Encryption Standard (DES) [31] has been around for more than 25 years. DES was an outcome of a call for primitives in 1974 which didn't result in many serious candidates except for a predecessor of DES, Lucifer [15, 35] designed by IBM around 1971. It took another year for a joint IBM–NSA effort to turn *Lucifer* into DES. The structure of Lucifer was significantly altered and since the design rationale was never made public and the secret key size was reduced from 128-bit to 56-bits this initially resulted in controversy, and some distrust among the public.

However, in spite of all the controversy it is hard to underestimate the role of DES. DES was one of the first commercially developed (as opposed to government developed) ciphers whose structure was fully published. This effectively created a community of researchers who could analyse it and propose their own designs. This lead to a wave of public interest in cryptography, from which much of the cryptography as we know it today was born.

1 Description of DES

The Data Encryption Standard, as specified in FIPS Publication 46-3 [31], is a block cipher operating on 64-bit data blocks. The encryption transformation depends on a 56-bit secret key and consists of sixteen Feistel iterations surrounded by two permutation layers: an initial bit permutation IP at the input, and its inverse IP^{-1} at the output. The structure of the cipher is depicted in Fig. 1. The decryption process is the same as the encryption, except for the order of the round keys used in the Feistel iterations. As a result, most of the circuitry can be reused in hardware implementations of DES.

The 16-round Feistel network, which constitutes the cryptographic core of DES, splits the 64-bit data blocks into two 32-bit words (denoted by L_0 and R_0). In each iteration (or round), the second word R_i is fed to a function f and the result is added to the first word L_i. Then both words are swapped and the algorithm proceeds to the next iteration.

The function f is key-dependent and consists of four stages (see Fig. 2). Their description is given below. Note that all bits in DES are numbered from left to right, i.e., the leftmost bit of a block (the most significant bit) is bit 1.

1. **Expansion (E).** The 32-bit input word is first expanded to 48 bits. This is done by duplicating and reordering half of the bits. The selection of bits is specified by Table 1. The first row in the table refers to the first 6 bits of the expanded word, the second row to bits 7–12, and so on. Thus bit 41 of the expanded word, for example, gets its value from bit 28 of the input word.
2. **Key mixing.** The expanded word is XORed with a round key constructed by selecting 48 bits from the 56-bit secret key. As explained below, a different selection is used in each round.
3. **Substitution.** The 48-bit result is split into eight 6-bit words which are substituted in eight parallel 6×4-bit S-boxes. All eight S-boxes, called

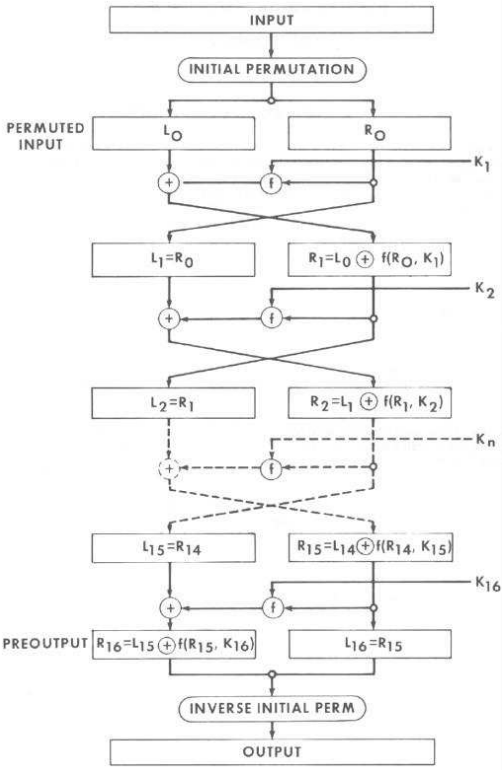

Fig. 1. The encryption function

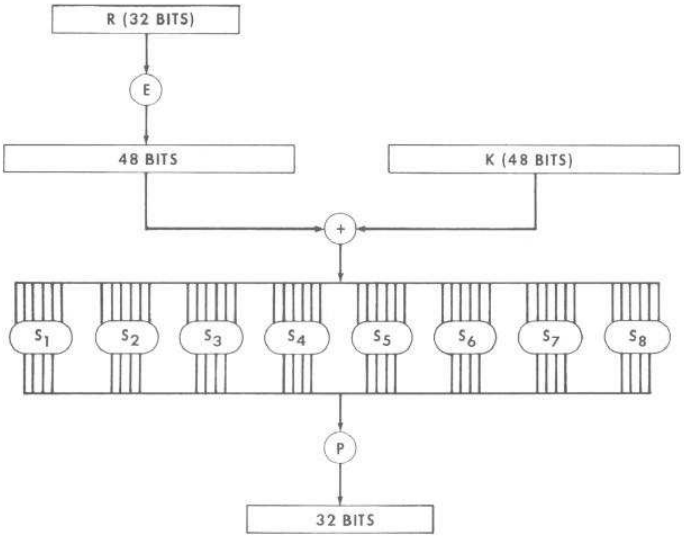

Fig. 2. The function f.

Table 1. Expansion E and permutation P.

E					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

$S_1, S_2, \ldots, S_8$, are different but have the same special structure, as appears from their specifications in Table 2. Each row of the S-box tables consists of a permutation of the 4-bit values $0, \ldots, 15$. The 6-bit input word is substituted as follows: first a row is selected according to the value of the binary word formed by concatenating the first and the sixth input bit. The algorithm then picks the column given by the value of the four middle bits and outputs the corresponding 4-bit word.

4. **Permutation (P).** The resulting 32 bits are reordered according to a fixed permutation specified in Table 1 before being sent to the output. As before, the first row of the table refers to the first four bits of the output.

The selection of key bits in each round is determined by a simple key scheduling algorithm. The algorithm starts from a 64-bit secret key which includes 8 parity bits that are immediately discarded after having been checked. The remaining 56 secret key bits are first permuted according to a permutation PC_1 (see Table 4). The result is split into two 28-bit words C_0 and D_0, which are cyclically rotated over 1 position to the left after rounds 1, 2, 9, 16, and over 2 positions after all other rounds (the rotated words are denoted by C_i and D_i). The round keys are constructed by repeatedly extracting 48 bits from C_i and D_i at 48 fixed positions determined by a table PC_2 (see Table 4). A convenient feature of this key scheduling algorithm is that the 28-bit words C_0 and D_0 are rotated over exactly 28 positions after 16 rounds. This allows hardware implementations to efficiently compute the round keys on-the-fly, both for the encryption and the decryption.

2 Cryptanalysis of DES

DES has been subject to very intensive cryptanalysis. Initial attempts [16] did not identify any serious weaknesses except for the short key-size. It was noted that DES has a *complementation property*, i.e. given an encryption of the plaintext P into the ciphertext C under the secret key K: $E_K(P) = C$, one knows that the complement of the plaintext will be encrypted to the complement of the ciphertext under the complement of the key: $E_{\bar{K}}(\bar{P}) = \bar{C}$ (by complement we mean flipping of all the bits). Another feature was the existence of four weak

Table 2. DES S-boxes.

S_1 :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 :	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1 :	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2 :	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3 :	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2 :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 :	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1 :	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2 :	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3 :	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3 :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 :	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1 :	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2 :	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3 :	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4 :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 :	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1 :	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2 :	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3 :	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5 :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 :	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1 :	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2 :	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3 :	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6 :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 :	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1 :	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2 :	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3 :	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7 :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 :	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1 :	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2 :	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3 :	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8 :	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0 :	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1 :	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2 :	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3 :	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Table 3. Initial and final permutations.

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Table 4. DES key schedule bit selections.

PC_1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC_2					
14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

keys, for which the cipher is an *involution*: $E_K(E_K(m)) = m$ (for these keys the contents of the key-schedule registers C and D is either all zeros or all ones), and six additional pairs of *semi-weak keys* for which $E_{K1}(E_{K2}(m)) = m$. The complementation and the weak-key properties are the result of interaction of the key-schedule, which splits the key-bits into two separate registers and the Feistel structure of the cipher. A careful study of the cycle structure of DES for weak and semi-weak keys has been given by Moore and Simmons [30]. See the book of Davies and Price [11] for a more detailed account on these and other features of DES identified prior to 1989. The properties of the group generated by DES permutations have been also studied intensively. Coppersmith and Grossman have shown [9] that in principle DES-like components can generate any permutation from the *alternating group* $A_{2^{64}}$ (all *even permutations*, i.e. those that can be represented with an even number of transpositions). However, DES implements only 2^{56} permutations, which is a tiny fraction of all the even permutations. If the set of 2^{56} DES permutations was closed under composition, then multiple encryption as used for example in Triple-DES would be equivalent to single encryption and thus would not provide any additional strength. A similar weakness would be present if the size of the group generated by the DES permutations would be small. Using the special properties of the weak keys it has been shown that DES generates a very large group, with a lower-bound of 2^{2499} permutations [7, 8] which is more than enough to make the *closure attacks* [18] impractical.

In the two decades since its design three important theoretical attacks capable of breaking the cipher faster than exhaustive search have been discovered: differential cryptanalysis (1990) [5], linear cryptanalysis (1993) [22] and the *improved Davies' attack* [3, 12]. An interesting twist is that differential cryptanalysis was known to the designers of DES and DES was constructed in particular to withstand[1] this powerful attack [8]. That is why the cipher's design criteria were kept secret. Many of these secrets became public with the development of differential cryptanalysis and were later confirmed by the designers [33]. Both differential and linear attacks as well as Davies' attack are not much of a threat to real-life applications since they require more than 2^{40} texts for the analysis. For example: a linear attack requires 2^{43} known plaintexts to be encrypted under the same secret key. If the user changes the key every 2^{35} blocks the success probability of the attack would be negligible. Nevertheless, linear attacks were tested [23] in practice, run even slightly faster than theoretically predicted [17], and can potentially use twice less data in a chosen plaintext scenario [20]. In the case of the differential attack 2^{47} chosen plaintexts are required, though the attack would still work if the data is coming from up to 2^{33} different keys. However the huge amount of *chosen* plaintext makes the attack impractical. In the case of Davies' attack the data requirement is 2^{50} *known plaintexts*, which is also clearly impractical.

Though differential and linear attacks are hard to mount on DES they proved to be very powerful tools for cryptanalysis and many ciphers which were not designed to withstand these attacks have been broken, some even with practical attacks. See for example the cipher FEAL [28, 29, 34]. In fact both attacks have been discovered while studying this cipher [4, 25], which was proposed as a more secure alternative to DES.

The exhaustive key search currently remains the most dangerous approach to the cryptanalysis of DES. It was clear from the very start that 56-bit key can be cryptanalysed in practical time using practical amount of resources. In 1977 a design for a key-search machine was proposed by Diffie and Hellman [13] with a cost of US $ 20 million and the ability to find a solution in a single day. Later Hellman proposed a chosen plaintext time-memory tradeoff approach, which would allow to build an even cheaper machine, assuming that a precomputation of 2^{56} encryption steps is done once for a single chosen plaintext. An effective and complete ASIC design for a key-search machine has been proposed by Wiener in 1993 [37]. It was shown that the US $ 1 million machine would run through the full key-space in 7 hours. In 1998 the Electronic Frontier Foundation (EFF) has demonstrated a dedicated hardware machine which cost less than US$ 250 000 and could run through the full key-space in four days [14]. In a parallel development it has been shown that a network of tens of thousands of PCs (a

[1] Note that DES is strong but not optimal against linear cryptanalysis or improved Davies' attack, for example simple reordering of the S-boxes would make the cipher less vulnerable to these attacks without spoiling its strength against the differential attack [24]. This could indicate that the designers of DES did not know about such attacks.

computational power easily available to a computer virus, for example) could do the same work in several weeks. It became clear to everyone that DES had to be upgraded to triple-DES or replaced. At that time the AES competition has been started. As a result of this effort DES has been replaced by a successor, AES, which is based on a 128-bit block 128/192/256-bit key cipher *Rijndael*.

3 Extensions of DES

So where is DES today? DES is not obsolete. Due to substantial cryptanalytic effort and the absence of any practical cryptanalytic attack, the structure of DES has gained public trust. There have been several proposals to remedy the short key size problem plaguing the cipher:

- **Triple-DES (Diffie-Hellman [13]).** The idea is to multiple encrypt the block using DES three times with two or three different keys. This method gains strength both against cryptanalytic attacks as well as against exhaustive search. It is weak against related key attacks, however, and the speed is three times slower than single DES. A two-key variant in the form of *Encrypt-Decrypt-Encrypt (E-D-E)*, i.e., $E_{K_1}(D_{K_2}(E_{K_1}(m)))$ has been proposed by IBM (Tuchman, 1978) and is still in wide use by the banking community. The convenience of this option is that it is backward compatible with a single DES encryption, if one sets $K_1 = K_2$.
- **Independent subkeys (Berson [1]).** The idea is to use independently generated 48-bit subkeys in each round. The total key-size is 768 bits, which stops the exhaustive search attack. However, the cryptanalytic attacks like differential or linear do work almost as good as for DES. The speed of this proposal is as for single DES, but it has a slower key-schedule.
- **Slow key-schedule (Quisquater et al. [32] or Knudsen [10]).** Exhaustive search is stopped by loosing key-agility of a cipher.
- **DES-X. (Rivest, 1984).** The idea is to XOR additional 64-bits of secret key material at the input and at the output of the cipher. See the article on DES-X for more details. This is very effective against exhaustive search, but does not stop old cryptanalytic attacks on DES, and allows new related key attacks. This approach allows to reuse old hardware. The speed is almost the same as that of a single DES.
- **Key-dependent S-boxes. (Biham-Biryukov [2]).** The idea is similar to DES-X, but the secret key material is XORed before and after the S-boxes. S-boxes are reordered to gain additional strength. The result is secure against exhaustive search and improves the strength against cryptanalytic attacks (with the exception of related key attacks). This approach applies to software or to hardware which permits to load new S-boxes. The speed is the same as that of a single DES.

As of today two-key and three-key triple DES is still in wide use and is included in NIST and ISO standards. However, two-key triple DES variants are not recommended for use due to dedicated meet-in-the-middle attack by

Oorschot and Wiener [36] with complexity $2^{120-\log n}$ steps given $O(n)$ *known plaintexts* and memory. For example, if $n = 2^{40}$, complexity of attack is 2^{80} steps. This attack is based on an earlier attack by Merkle and Hellman [27] which required 2^{56} *chosen plaintexts*, steps, and memory. These attacks are hard to mount in practice, but they are an important certificational weakness.

The recommended usage mode for triple-DES is *Encrypt-Encrypt-Encrypt (E-E-E)* (or *Encrypt-Decrypt-Encrypt (E-D-E)*) with three independently generated keys (i.e. 168 key bits in total), for which the best attacks are the classical meet-in-the-middle attack with only three known plaintexts, 2^{56} words of memory and 2^{111} analysis steps; and the attack by Lucks [21] which requires 2^{108} time steps and 2^{45} known plaintexts. These attacks are clearly impractical.

The DES-X alternative is also in popular use due to simplicity and almost no speed loss. Thorough analysis of a generic construction is given in [19] and the best currently known attack is a slide attack [6] with complexity of n known plaintexts and $2^{121-\log n}$ analysis steps (for example: 2^{33} known plaintexts and memory and 2^{87} analysis steps).

–Alex Biryukov, Christophe De Cannière.

References

[1] T. A. Berson, "Long key variants of DES," in *Advances in Cryptology – CRYPTO'82* (D. Chaum, R. L. Rivest, and A. T. Sherman, eds.), pp. 311–313, Plenum Press, 1983.

[2] E. Biham and A. Biryukov, "How to strengthen DES using existing hardware," in *Advances in Cryptology – ASIACRYPT'94* (J. Pieprzyk and R. Safavi-Naini, eds.), vol. 917 of *Lecture Notes in Computer Science*, pp. 398–412, Springer-Verlag, 1995.

[3] E. Biham and A. Biryukov, "An Improvement of Davies' Attack on DES," *Journal of Cryptology*, vol. 10, no. 3, pp. 195–206, 1997.

[4] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," in Menezes and Vanstone [26], pp. 2–21.

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

[6] A. Biryukov and D. Wagner, "Advanced slide attacks," in *Advances in Cryptology – EUROCRYPT 2000* (B. Preneel, ed.), vol. 1807 of *Lecture Notes in Computer Science*, pp. 589–606, Springer-Verlag, 2000.

[7] K. W. Campbell and M. J. Wiener, "DES is not a group," in *Advances in Cryptology – CRYPTO'92* (E. F. Brickell, ed.), vol. 740 of *Lecture Notes in Computer Science*, pp. 512–520, Springer-Verlag, 1993.

[8] Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks," *IBM Journal of Research and Development.*, vol. 38, no. 3, pp. 243–250, 1994.

[9] D. Coppersmith and E. Grossman, "Generators for certain alternating groups with applications to cryptography," *SIAM Journal Applied Math*, vol. 29, no. 4, pp. 624–627, 1975.

[10] I. Damgard and L. R. Knudsen, "Two-key triple encryption," *Journal of Cryptology*, vol. 11, no. 3, pp. 209–218, 1998.

[11] D. W. Davies and W. L. Price, *Security for Computer Networks*. John Wiley & Sons, 1989. New York, 2nd. edition.
[12] D. W. Davies and S. Murphy, "Pairs and triplets of DES S-Boxes," *Journal of Cryptology*, vol. 8, pp. 1–25, 1995.
[13] W. Diffie and M. Hellman, "Exhaustive cryptanalysis of the NBS data encryption standard," *Computer*, vol. 10, no. 6, pp. 74–84, 1997.
[14] Electronic Frontier Foundation (EFF), "DES cracker." `http://www.eff.org/DEScracker/`, 1998.
[15] H. Feistel, "Cryptography and computer privacy," *Scientific American*, vol. 228, pp. 15–23, May 1973.
[16] M. E. Hellman, R. Merkle, R. Schroppel, L. Washington, W. Diffe, S. Pohlig, and P. Schweitzer, "Results of an initial attempt to cryptanalyze the NBS Data Encryption Standard," Technical report, Stanford University, U.S.A, 1976.
[17] P. Junod, "On the complexity of Matsui's attack," in *Selected Areas in Cryptography, SAC 2001* (S. Vaudenay and A. M. Youssef, eds.), vol. 2259 of *Lecture Notes in Computer Science*, pp. 199–211, Springer-Verlag, 2001.
[18] B. S. Kaliski, R. L. Rivest, and A. T. Sherman, "Is the data encryption standard a group?," *Journal of Cryptology*, vol. 1, pp. 3–36, 1988.
[19] J. Kilian and P. Rogaway, "How to protect DES against exhaustive key search," in *Advances in Cryptology – CRYPTO'96* (N. Koblitz, ed.), vol. 1109 of *Lecture Notes in Computer Science*, pp. 252–267, Springer-Verlag, 1996.
[20] L. R. Knudsen and J. E. Mathiassen, "A chosen-plaintext linear attack on DES," in *Fast Software Encryption, FSE 2000* (B. Schneier, ed.), vol. 1978 of *Lecture Notes in Computer Science*, pp. 262–272, Springer-Verlag, 2001.
[21] S. Lucks, "Attacking triple encryption," in *Fast Software Encryption, FSE'98* (S. Vaudenay, ed.), vol. 1372 of *Lecture Notes in Computer Science*, pp. 239–257, Springer-Verlag, 1998.
[22] M. Matsui, "Linear cryptanalysis method for DES cipher," in *Advances in Cryptology – EUROCRYPT'93* (T. Helleseth, ed.), vol. 765 of *Lecture Notes in Computer Science*, pp. 386–397, Springer-Verlag, 1993.
[23] M. Matsui, "The first experimental cryptanalysis of the data encryption standard," in *Advances in Cryptology – CRYPTO'94* (Y. Desmedt, ed.), vol. 839 of *Lecture Notes in Computer Science*, pp. 1–11, Springer-Verlag, 1994.
[24] M. Matsui, "On correlation between the order of S-boxes and the strength of DES," in *Proceedings of Eurocrypt'94* (A. De Santis, ed.), no. 950 in Lecture Notes in Computer Science, pp. 366–375, Springer-Verlag, 1995.
[25] M. Matsui and A. Yamagishi, "A new method for known plaintext attack of FEAL cipher," in *Proceedings of Eurocrypt'92* (R. A. Rueppel, ed.), no. 658 in Lecture Notes in Computer Science, pp. 81–91, Springer-Verlag, 1992.
[26] A. Menezes and S. A. Vanstone, eds., *Advances in Cryptology – CRYPTO'90*, vol. 537 of *Lecture Notes in Computer Science*, Springer-Verlag, 1991.
[27] R. C. Merkle and M. E. Hellman, "On the security of multiple encryption," *Communications of the ACM*, vol. 24, 1981.
[28] S. Miyaguchi, "The FEAL-8 cryptosystem and a call for attack," in *Advances in Cryptology – CRYPTO'89* (G. Brassard, ed.), vol. 435 of *Lecture Notes in Computer Science*, pp. 624–627, Springer-Verlag, 1990.
[29] S. Miyaguchi, "The FEAL cipher family," in Menezes and Vanstone [26], pp. 627–638.
[30] J. H. Moore and G. J. Simmons, "Cycle structures of the DES with weak and semi-weak keys," in *Advances in Cryptology – CRYPTO'86* (A. M. Odlyzko, ed.), vol. 263 of *Lecture Notes in Computer Science*, pp. 9–32, Springer-Verlag, 1987.

[31] National Institute of Standards and Technology, "FIPS-46-3: Data Encryption Standard (DES)," May 1999. Available at `http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf`.

[32] J.-J. Quisquater, Y. Desmedt, and M. Davio, "The importance of "good" key scheduling schemes (how to make a secure DES scheme with $\leq$ 48 bit keys)," in *Advances in Cryptology – CRYPTO'85* (H. C. Williams, ed.), vol. 218 of *Lecture Notes in Computer Science*, pp. 537–542, Springer-Verlag, 1986.

[33] sci.crypt, "Subject: DES and differential cryptanalysis." unpublished, `http://www.esat.kuleuven.ac.be/~abiryuko/coppersmith_letter.txt`, 1992.

[34] A. Shimizu and S. Miyaguchi, "Fast data encipherment algorithm FEAL," in *Advances in Cryptology – EUROCRYPT'87* (D. Chaum and W. L. Price, eds.), vol. 304 of *Lecture Notes in Computer Science*, pp. 267–278, Springer-Verlag, 1988.

[35] J. L. Smith, "The design of Lucifer: A cryptographic device for data communications," Technical report, IBM T.J. Watson Research Center, Yorktown Heights, N.Y., 10598, U.S.A., 1971.

[36] P. C. van Oorschot and M. J. Wiener, "A known plaintext attack on two-key triple encryption," in *Advances in Cryptology – EUROCRYPT'90* (I. Damgård, ed.), vol. 473 of *Lecture Notes in Computer Science*, pp. 318–325, Springer-Verlag, 1990.

[37] M. Wiener, "Efficient des key search," *Practical Cryptography for Data Internetworks*, pp. 31–79, 1996. presented at the rump session of Crypto'93.