KEY SCHEDULING IN DES TYPE CRYPTOSYSTEMS

Lawrence Brown and Jennifer Seberry

Department of Computer Science
University College, UNSW, Australian Defence Force Academy
Canberra ACT 2600. Australia.

Abstract

This paper reviews some possible design criteria for the key schedule in a DES style cryptosystem. The key schedule involves a Key Rotation component, and the permutation $PC2$. Together these provide for a diffusion of dependency of ciphertext bits on key bits. Some empirical rules which seem to account for the derivation of the key schedule used in the DES are first presented. A number of trials were run with various key schedules, and some further design rules were derived. An alternative form of key schedule was then tested. This used either a null $PC2$, or one in which permutations only occurred within the inputs to a given S-box, and a much larger rotation schedule than used in the DES. This was found to be as effective as the key schedule used in the current DES, and is proposed for use in new cryptosystems.

1. Introduction

The Data Encryption Standard (DES) [NBS77] is currently the only certified encryption standard. It has achieved wide utilization, particularly in the banking and electronic funds transfer areas, and is an Australian standard [ASA85] among others. With the current significant use of DES (especially in banking), there is interest in designing and building a DES-type cryptosystem with an extended key length of either 64 (rather than 56) or 128 bits. This is one of a continuing series of papers [Bro89], [BrS90], [PiS89], [Pie89], analysing aspects of the current DES, and indicating criteria to be used in the design of future schemes.

This paper will concentrate on the design of the key schedule, which involves a key rotation component, and the permutation $PC2$. Together these provide for a diffusion of dependency of ciphertext bits in key bits. As a measure of effectiveness, Meyer's analysis of output bit dependence on key bits will be used [MeM82]. Some empirical rules for the key schedule, derived previously [Bro89], will be presented. A discussion of some alternatives to the current schedule will be presented, followed by the results obtained from testing a number of alternate schedules. A presentation of the implications from these in the design of any extended DES type schemes will conclude the paper.

2. The Key Schedule in DES

The central component of the DES cryptosystem is the function g, which is a composition of expansion function E, eight substitution boxes (S-boxes) S, and a permutation P[1]. Function g has as inputs the plaintext $[L(i-1), R(i-1)]$ from the previous round, and a selection of key bits $K(i)$ (see Fig 1.). This may be written as:

---

[1] A more detailed description of these functions may be found in [NBS77], [ASA85] or [SeP89].

$$g\colon\ R(i) = L(i-1) \oplus P(S(E(R(i-1)) \oplus K(i))),\ L(i) = R(i-1).$$

Fig 1. DES as a Mixing Function

The *key schedule* in a DES algorithm is responsible for forming the sixteen 48-bit sub-keys $K(i)$ used in the rounds of the encryption procedure. This function is important since if the same key is used on successive rounds, it can weaken the resulting algorithm (see [GrT78], [MeM82], [MoS87], [MoS86], and [ASA85]). In detail, the 64-bit key is permuted by $PC1$. This permutation performs two functions: first it strips the eight parity bits out, and then distributes the remaining 56 bits over two 28-bit halves $C(0)$ and $D(0)$. The cryptographic significance of this permutation is questionable [DDF83]. Subsequently for each round, each 28-bit register is rotated left either one or two places according to the following schedule (subsequently denoted KS):

Table 1 - Key Schedule for DES

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Shift	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1
Total	1	2	4	6	8	10	12	14	15	17	19	21	23	25	27	28

After the shift, the resultant 28-bit vectors are permuted by $PC2$ (which in fact consists of two 28-bit permutations, each of which selects 24 bits) to form the sub-key for that round. This permutation may be written in terms of which S-box each bit is directed to, as shown in Table 2 (nb: an * indicates an autoclave S-box input rather than a message input; an X specifies exclusion of that bit).

Table 2 - Current DES Permutation PC2
C: 1 4* 2* 3 1* 2 4 3* X 2* 1 3 4 1* 2 4* 1 X 3 4 2 X 3* 1 X 3 4 2
D: 8 6* 5 8* 6 7 X 8 5 X 7 6 5* 8 X 7* 6 8* 5 6* 7 8 6 5 7* X 5* 7

The sub-keys $K(i)$ may be written as;

$$K(i) = PC2(KS(U,\, i)), \qquad \text{where } U = PC1(K)$$

and $KS(U, i)$ is the key rotation schedule for input block U at round i.

3. Empirical Key Schedule Design Criteria

In Brown [Bro89], some empirical design rules for the key schedule are presented. The rules presented for permutation $PC2$ are (if the bits are sorted into ascending order of their input bits):

- 1 bits permuted to the same S-box input are no closer than 3 bits apart
- 2 bits permuted to an S-box input must have a span from lowest to highest input bit number of at least 22 of the 28 bits in each key half (alternatively, the average spacing must be at least 3 2/3)
- 3 bits permuted to the selector bits a, f on a given S-box must not be adjacent in the sorted list of input bits
- 4 bits not selected by $PC2$ must be at least 3 places apart

The design of the key schedule KS is obviously related to the design of $PC2$ by rules 1 and 4 given above. Brown notes that the key schedule KS ensures that:

- 1 each bit is used as input to each S-box
- 2 no bit is used as input to the same S-box on successive rounds
- 3 the total number of bits rotated is 56 (which implies that $K(0) = K(16)$, enabling the decryption operation to use right shifts in reverse order).

4. Ciphertext Dependence on Key Bits

This analysis is complex, and is dependent on the choices of permutations P and $PC2$ as well as the KS the key schedule[2]. To quantify this dependency, a 64 * 56 array F_r is formed, in which element $F_r[i,\, j]$ specifies a dependency of output bit $X(j)$ on key bit $U(i)$. The vector U is that formed after PC1 is applied, ie $U = PC1(K)$. The number of marked elements in G_r will be examined to provide a profile of the degree of dependence achieved by round r. Details of the derivation of this matrix, and the means by which entries are propagated, may be found in [MeM82]. This analysis technique will be used as a measure of effectiveness for possible key schedules. In particular, two criteria are used:

- rate of growth of output bit dependence on key bits by any S-box inputs
- rate of growth of output bit dependence on key bits by BOTH message and autoclave S-box inputs

---

[2] but not on permutations IP, FP and $PC1$, which only serve to renumber the plaintext, ciphertext, and key bits respectively. The analysis done in this paper ignores these permutations for this reason.

5. Alternatives for the Key Schedule

The purpose of the above rules in designing a key schedule may be summarized as follows:

> to present each key bit to a message input, and to an autoclave input, of each S-box as quickly as possible.

This is achieved by a combination of the key rotation schedule KS, the key permutation $PC2$, and the function $g = S.P.E$. Trials have been performed in which each of these is varied in turn, to analyse the effect of each.

In addition to that underlying design purpose, there is a pragmatic decision on the size of the key registers. In the current scheme, the key is divided into two halves. An alternate form could be to have a single large key schedule register. We also wish to extend the size of the key, in order to ensure it is large enough to withstand any foreseeable exhaustive search style attack. One way of providing a measure of this, whilst still maintaining compatibility with existing protocols, would be to remove the notion of parity bits in the key, and use all 64-bits. Combining these two ideas we have the following possibilities for $PC2$:

28 -> 24 bit
56 -> 48 bit
32 -> 24 bit
64 -> 48 bit

Initially, a key schedule with the same form as the current DES was examined, in order that comparisons with the effectiveness of the current DES scheme could be made. Having obtained some guidelines from these trials, key schedules involving some of the alternatives were then tried.

6. Some Trials on New Key Schedules

In Brown [Bro89], some empirical design criteria for permutation $PC2$ and the Key Rotation Schedule were presented. The authors have subsequently used these rules to generate a set of permutations $PC2$. Since all possible 28->24 bit permutations could not be tried, permutations with the form shown in Table 3 were tried (that is all arrangements of the 4 excluded bits, subject to the rules set, were found). This form was chosen in order to distribute key bits to each of the 4 S-boxes being fed by each half of the key schedule as quickly as possible. A total of 7315 permutations were found.

Table 3 - form of generated Permutations PC2
C: 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 + X X X X
D: 5 6 7 8 5 6 7 8 5 6 7 8 5 6 7 8 5 6 7 8 5 6 7 8 + X X X X

Ciphertext-Key Dependences (CKdep) tests on these permutations produced results shown in Table 4 (with comparisons to the current and worst case $PC2$ supplied for comparison).

Table 4 - Dependency of Ciphertext bits on Key bits Using Current DES Permutation P and Key Schedule					
Round	Std $PC2$	Worst $PC2$	Generated $PC2$	Regular X 2	Regular X 1, X 3, X 4
1	5.36	5.36	5.36	5.36	5.36
2	39.17	42.19	38.50-39.06	39.06	38.62
3	82.25	81.47	80.25-82.37	82.37	81.47
4	98.44	91.29	96.65-98.66	98.66	98.21
5	100.00	96.21	99.55-100.00	100.00	100.00
6	100.00	99.55	100.00	100.00	100.00
7	100.00	100.00	100.00	100.00	100.00
8	100.00	100.00	100.00	100.00	100.00

Some of these permutations performed better than the $PC2$ used in the current DES. The best of these were selected, 15 being found. These 15 permutations were all found to have a special form, namely that the excluded bits always fell between bits permuted to S-box 1 and S-box 2 (or 5 and 6 in the D-side). There are thus exactly 15 since 15 = $^6C_4$. In order to investigate these permutations with a regular placing of the excluded bits, all 60 such permutations were generated. A CKdep analysis of these permutations resulted in only two results, one for permutations with the excluded bit before a bit permuted to S-box 2 (Regular X 2), and one for the others (Regular X 1, X 3, X 4). These results are also shown in Table 4.

So far, we have used the first of the two criteria presented earlier, namely the growth of overall bit dependence of output bits on key bits. If we now consider the alternate measure, namely growth in dependence of output bits on key bits by both message and autoclave S-box inputs, then the results become less clear. As shown in Table 5, whilst growth of overall dependence is greater with the regular $PC2$'s, growth of both is worse.

Table 5 - Dependency of Ciphertext bits on Key bits Using Current DES Permutation P and Key Schedule by Both Message and Autoclave S-box Inputs								
Round	1	2	3	4	5	6	7	8
CKdep	Both,Either	Both,Either	Both,Either	Both,Either	Both,Either	Both,Either	Both,Either	Both,Either
PC2.std	0.0,5.36	2.01,39.17	36.50,82.25	81.03,98.44	95.87,100.0	99.33,100.0	100.0,100.0	100.0,100.0
PC2.worst	0.0,5.36	0.0,42.19	33.71,81.47	73.88,91.29	84.38,96.21	92.86,99.55	98.66,100.0	100.0,100.0
PC2 X 1	0.0,5.36	0.22,38.62	29.91,81.47	65.07,98.21	73.66,100.0	80.13,100.0	87.28,100.0	93.97,100.0
PC2 X 2	0.0,5.36	0.45,39.06	30.36,82.37	67.08,98.66	77.01,100.0	83.26,100.0	90.18,100.0	95.87,100.0
PC2 X 3	0.0,5.36	0.45,39.06	30.13,81.92	66.74,98.21	76.79,100.0	83.04,100.0	89.96,100.0	95.76,100.0
PC2 X 4	0.0,5.36	0.45,39.06	30.13,81.92	66.52,98.21	75.67,100.0	80.36,100.0	86.38,100.0	92.41,100.0

A closer look at the structure of the regular permutations shows that the autoclave input bits are clustered, due to the method used to assign them to S-box inputs. By altering the order of inputs within each S-box, a more regular arrangement of autoclave inputs was obtained. When these were tested, the growth of dependence on both was much greater, thus emphasizing the importance of this criterion on the design of $PC2$.

To obtain an indication of the relative influences of each of the components in the key schedule, a series of trials were run, in which each of the following three components were varied with the specified alternatives:

P based on the results in [BrS90], two permutations P were used:

• the current DES *P* and

• a strictly regular permutation generated by a difference function on the S-box number of [+1 -2 +3 +4 +2 -1]. Because of its very regular structure, the propagation of dependencies may be more easily calculated.

*PC*2 from the above work, the 4 best performing regular *PC*2 were extracted. Then these were processed to provide three levels of clustering of the autoclave inputs.

KS the key variant in the key rotation schedule appears to be the distribution of shifts of 1 verses 2 places. A set of key schedules with various numbers of shifts of 1 initially were derived as shown in Table 6.

Table 6 - Trial Key Schedules

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
KS	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1
KS	1	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1
KS	1	1	2	2	2	2	2	2	2	2	2	2	2	2	1	1
KS	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	1
KS	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2

When this test was run, the following conclusions were made:

- the current permutation *P* performed better, possibly because its less than regular structure assisted the distribution of dependencies between the autoclave and message inputs.
- permutations *PC*2 with the best spread of autoclave inputs, performed best as expected.
- a key schedule with as many shifts of 1 initially performed best. This again would appear to be a function of the best method for spreading bits to as many S-box inputs as soon as possible.

7. Design Criteria for New Key Schedules

From the above results, the design principles for designing key schedules can now be summarized as follows:

The key schedule ensures that:

1 each key bit is used as input to each S-box in turn

2 no bit is used as autoclave inputs on successive rounds

3 no bit is excluded on successive rounds

4 the final key register value is identical to the original key register value (to enable easy reversal of the key schedule for decryption)

8. An Alternative Key Schedule Design

In the design of the DES, small key rotations were used, which required the use of permutation *PC*2 to provide a fan-out of key-bits across the S-box inputs, in order to satisfy the above principles. An alternative design can be envisaged in which a large key rotation interval is used, along with a null *PC*2 (ie: so called worst case *PC*2), or a local *PC*2 which only permutes bits within each block of 6 S-box inputs. The two *PC*2 permutations used are shown in

Table 7.

Table 7 - Null and Local Permutations PC2 for Alternative Key Schedule
C: 1* 1 1 1 1 1* 2* 2 2 2 2 2* 3* 3 3 3 3 3* 4* 4 4 4 4 4* X X X X
D: 5* 5 5 5 5 5* 6* 6 6 6 6 6* 7* 7 7 7 7 7* 8* 8 8 8 8 8* X X X X
C: 1* 1 1 1 1 1* 2 2 2 2* 2* 2 3 3* 3* 3 3 3 4* 4 4 4 4 4* X X X X
D: 5* 5 5 5 5 5* 6 6 6 6* 6* 6 7 7* 7* 7 7 7 8* 8 8 8 8 8* X X X X

For this design, a constant key rotation of 7 bits was used, both because it is larger than the number of inputs to an S-box, and because after sixteen rounds, the key register contents are the same as the original value (since 7 * 16 = 112 = 4 * 28 = 2 * 56), for both split key registers or a single large key register. This schedule is shown in Table 8.

Table 8 - Alternative Constant Key Schedule																
Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
KS	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

The results obtained for these $PC2$ permutations and this key schedule, using both a split key rotation register, and a single key register, are shown in Table 9.

Table 9 - Dependency of Ciphertext bits on Key bits Using Current DES Permutation P and the Alternative Key Schedule by Both Message and Autoclave S-box Inputs

Round	1	2	3	4	5	6	7	8
PC2	Both,Either	Both,Either	Both,Either	Both,Either	Both,Either	Both,Either	Both,Either	Both,Either
Split Key Register Used								
null	0.0,5.36	1.56,39.06	34.82,82.03	76.56,98.33	91.52,100.0	98.21,100.0	100.0,100.0	100.0,100.0
local	0.0,5.36	2.57,39.06	38.17,82.03	83.82,98.33	98.21,100.0	100.0,100.0	100.0,100.0	100.0,100.0
Single Key Register Used								
null	0.0,5.36	1.79,38.73	35.04,81.70	76.56,98.33	91.52,100.0	98.21,100.0	100.0,100.0	100.0,100.0
local	0.0,5.36	2.79,38.73	38.39,81.70	83.82,98.33	98.21,100.0	100.0,100.0	100.0,100.0	100.0,100.0

These results are very similar in performance to the key schedule used in the current DES (see Table 5). The null $PC2$ performs slightly worse, whilst the local $PC2$ performs better. Depending on the efficiency required, a tradeoff between best performance and ease of implementation can be made between these. There is very little difference in performance between the split and single key rotation registers, thus either could be chosen, depending on other constraints.

9. Conclusion

The key schedule in the current DES has been analysed, and some empirical principles which could have been used in its design derived. These were used to test a number of alternative key schedules, which led to the development of a new set of generalized principles to be used in the design of a new algorithm. An alternative key schedule which either eliminates permutation $PC2$, or uses a local $PC2$, was tried and found to be as effective as that used in the current DES. This is thus suggested for use in any new algorithm.

Acknowledgements

To the following members of the Centre for Computer Security Research: Leisa Condie, Thomas Hardjono, Mike Newberry, Cathy Newberry, Josef Pieprzyk, and Jennifer Seberry; and to: Dr. George Gerrity, Dr. Andzej Goscinski, and Dr. Charles Newton; for their comments on, suggestions about, and critiques of this paper.
Thankyou.

References

[ASA85] ASA, "*Electronics Funds Transfer - Requirements for Interfaces, Part 5, Data Encryption Algorithm*," AS2805.5-1985, Standards Association of Australia, Sydney, Australia, 1985.

[Bro89] L. Brown, "A Proposed Design for an Extended DES," in *Computer Security in the Age of Information*, W. J. Caelli (editor), North-Holland, Amsterdam, 1989.

[BrS90] L. Brown and J. Seberry, "On the Design of Permutation P in DES Type Cryptosystems," in *Advances in Cryptology - Eurocrypt'89*, Lecture Notes in Computer Science, no. 434, J. J. Quisquater and J. Vanderwalle (editors), pp. 696-705, |Springer Verlag|, Berlin, 1990.

[DDF83] M. Davio, Y. Desmedt, M. Fosseprez, R. Govaerts, J. Hulsbosch, P. Neutjens, P. Piret, J. Quisquater, J. Vanderwalle and P. Wouters, "Analytical Characteristics of the DES," in *Advances in Cryptology - |Proc.| of Crypto 83*, D. Chaum, R. L. Rivest and A. T. Sherman (editors), pp. 171-202, Plenum Press, New York, |August.| 22-24, 1983.

[GrT78] E. K. Grossman and B. Tuckerman, "Analysis of a Weakened Feistel-Like Cipher," in *|Proc.| 1978 IEEE |Conf.| On Communications*, pp. 46.3.1-5, IEEE, 1978.

[MeM82] C. H. Meyer and S. M. Matyas, *Cryptography: A New Dimension in Data Security*, |John Wiley & Sons, New York, 1982.

[MoS86] J. H. Moore and G. J. Simmons, "Cycle Structure of the Weak and Semi-Weak DES Keys," *Eurocrypt 86 - Abstracts of Papers* , p. 2.1, Linkoping, Sweden, 20-22 |May| 1986.

[MoS87] J. H. Moore and G. J. Simmons, *Advances in Cryptology - |Proc.| of CRYPTO'86*, Lecture Notes in Computer Science, no. 263, pp. 9-32, |Springer Verlag|, Berlin, 1987.

[NBS77] NBS, "*Data Encryption Standard (DES)*," FIPS PUB 46, US National Bureau of Standards, Washington, DC, |January.| 1977.

[PiS89] J. Pieprzyk and J. Seberry, "*Remarks on Extension of DES - Which Way to Go?*," Tech. Rep. CS89/4, |Dept. of Computer Science, UC UNSW, Australian Defence Force Academy|, Canberra, Australia, |February.| 1989.

[Pie89] J. Pieprzyk, "Error Propagation Property and Application in Cryptography," *IEE Proceedings-E, Computers and Digital Techniques*, vol. 136, no. 4, pp. 262-270, |July| 1989.

[SeP89] J. Seberry and J. Pieprzyk, *Cryptography: An Introduction to Computer Security*, |Prentice Hall, Englewood Cliffs, NJ|, 1989.