⌽WN33
Powers of t

Powers of θ	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	4	−12
2	0	0	0	−4	24

⌽WN33
Powers of t

Powers of θ	5	6	7	8	9
0	0	0	0	0	0
1	8	8	−12	4	0
2	−60	80	−60	24	−4

⌽WD33
Powers of t

Powers of θ	0	1	2	3	4
0	0	0	0	8	−16
1	0	0	0	−4	12

⌽WD33
Powers of t

Powers of θ	5	6	7	8
0	0	16	−8	0
1	−16	16	−12	4

WT3 ← .5 × PLIB ELIMFACT WN33 RATPOL WD33

Powers of θ

Powers of t	0	1	2	
0	0	1	−1	
1	0	−1	4	
2	0	−1	−6	Numerator
3	0	1	4	
4	0	0	−1	

Powers of t				
0	2	−1	0	
1	0	1	0	
2	−2	−1	0	Denominator
3	0	1	0	
4	0	0	0	

W3 ← (2 2ρ1 1 1 −1)ELIMFACT(1 2 1ρ1 1)SUBST⌽WT3

Powers of t

0	1	2	3	
0	4	−4	0	Numerator
1	5	5	1	Denominator

Appendix C. Orbit Problem Solution

$$
\begin{aligned}
F11 = {} & 2320275\,UVW^4 - 42723300\,UV^3W^3 + 266431410\,UV^5W^2 \\
& - 695674980\,UV^7W + 654729075\,UV^9 + 3479700\,U^2VW^3 \\
& - 53057340\,U^2V^3W^2 + 234084492\,U^2V^5W - 318715236\,U^2V^7 \\
& + 1189902\,U^3VW^2 - 14873940\,U^3V^3W + 36761454\,U^3V^5 \\
& + 93660\,U^4VW - 878268\,U^4V^3 + 1023\,U^5V
\end{aligned}
$$

$$
\begin{aligned}
G11 = {} & - 308745\,UW^4 + 11213100\,UV^2W^3 - 93324150\,UV^4W^2 \\
& + 290768940\,UV^6W - 310134825\,UV^8 - 255000\,U^2W^3 \\
& + 9297840\,U^2V^2W^2 - 57948120\,U^2V^4W + 97257888\,U^2V^6 \\
& - 40446\,U^3W^2 + 1657140\,U^3V^2W - 6379326\,U^3V^4 \\
& - 1008\,U^4W + 53640\,U^4V^2 - U^5
\end{aligned}
$$

Technical Note
Programming Techniques and Data Structures
D. McIlroy Editor

On the Security of Multiple Encryption

Ralph C. Merkle
Elxsi, Int.
Martin E. Hellman
Stanford University

Double encryption has been suggested to strengthen the Federal Data Encryption Standard (DES). A recent proposal suggests that using two 56-bit keys but enciphering 3 times (encrypt with a first key, decrypt with a second key, then encrypt with the first key again) increases security over simple double encryption. This paper shows that although either technique significantly improves security over single encryption, the new technique does not significantly increase security over simple double encryption. Cryptanalysis of the 112-bit key requires about 2^{56} operations and words of memory, using a chosen plaintext attack. While DES is used as an example, the technique is applicable to any similar cipher.

Key Words and Phrases: encryption; encrypt; cipher; encipher; cryptography; DES; data encryption standard; cryptanalysis; multiple encryption

CR Categories: 3.56, 3.57, 4.9

Introduction

Diffie and Hellman [2] have argued that the 56-bit key used in the Federal Data Encryption Standard (DES) [9] is too small and that current technology allows an exhaustive search of the 2^{56} keys. Although there is controversy surrounding this issue [1, 5, 7, 8, 10, 13], there is almost universal agreement [12, 2] that multiple

This work was supported under NSF Grants ENG 10173 and ELS 7916161.

Authors' present addresses: R.C. Merkle, 1134 Pimento Ave, Sunnyvale, CA 94087, M.E. Hellman, Department of Electrical Engineering, Stanford University, Stanford, CA 94305.

encryption using independent keys can increase the strength of DES. But, as noted in [2], the increase in security can be far less than might first appear.

Double Encryption

The simplest approach to increasing the key size is to encrypt twice, with two independent keys K1 and K2. Letting P be a 64-bit plaintext, C a 64-bit ciphertext, and K a 56-bit key, the basic DES encryption operation can be represented as

$$C = S_K(P), \tag{1}$$

and simple double encryption is obtained as

$$C = S_{K2}[S_{K1}(P)]. \tag{2}$$

While exhaustive search over all 2^{112} keys (K1–K2 pairs) requires 2^{112} operations and is clearly infeasible, this cipher can be broken under a known plaintext attack (where corresponding plaintext and ciphertext are both known) with 2^{56} operations, and 2^{56} words of memory [2]. The time required is therefore no greater than is needed to cryptanalyze a single 56-bit key exhaustively (although there is very significant additional cost for memory). If P and C represent a known plaintext–ciphertext pair, then the algorithm for accomplishing this [2] encrypts P under all 2^{56} possible values of K1, decrypts C under all 2^{56} values of K2, and looks for a match. For obvious reasons, this is called a "meet in the middle" attack; it is given in detail by the following algorithm (where n is the number of keys in the key space; for DES, $n = 2^{56}$):

(1) For $i = 1$ to n Do
  (a) Table[i] = $\langle S_i(P), i,$ "encrypt"$\rangle$
  (b) Table[$n + i$] = $\langle S_i^{-1}(C), i,$ "decrypt"$\rangle$

(2) Sort the table on the first field.

(3a) Search the table for adjacent entries of the form
  $\langle$value, $\hat{K}1$, "encrypt"$\rangle$
  $\langle$value, $\hat{K}2$, "decrypt"$\rangle$

(3b) Test to see if $\hat{K}1$ and $\hat{K}2$ are the correct keys by encrypting one additional plaintext–ciphertext pair.

Unicity distance arguments [4, 11] indicate that step (3a) will produce about 2^{48} false alarms: each of the 64 bits of known plaintext corresponds to one binary equation (bit of redundancy) and there are 112 binary unknowns (the bits of the key). Unicity distance arguments therefore predict $2^{112-64} = 2^{48}$ false alarms. A similar argument indicates that 64 bits of additional known plaintext suffices to reduce the overall false alarm rate at step (3b) to $2^{48-64} = 2^{-16}$, which is small.

While sorting causes the above algorithm to run in time $n \log n$, it could be rewritten using hash tables to run in essentially linear time. In any event, the present analysis will neglect logarithmic factors.

The use of double encryption provides an increase in security because the algorithm for cryptanalysis requires 2^{56} words of memory, as well as 2^{56} operations. The cost of a machine to perform 2^{56} operations in approximately a day has been estimated by Diffie and Hellman [2] to be about $20 million. The cost of 2^{56} 64-bit words of memory on 6250 cpi reels of magnetic tape, assuming 2400 foot reels that cost $20 each, is about $80 billion.

While the cost of implementing this search is high enough to prevent its use today, the danger of cheaper technology or shortcuts [5] in the future prompted Diffie and Hellman to suggest triple encryption with three independent keys, K1, K2, and K3. A generalized meet in the middle attack would then require 2^{112} operations and be well beyond the foreseeable technology for at least 50 years, and possibly forever.

Triple Encryption

At the 1978 National Computer Conference, Tuchman [12] proposed a triple encryption method which uses only two keys, K1 and K2. The plaintext is encrypted with K1, decrypted with K2, then again encrypted with K1, so that

$$C = S_{K1}\{S_{K2}^{-1}[S_{K1}(P)]\}. \tag{3}$$

This method seems to avoid the "meet in the middle" attack outlined above and is upwardly compatible with a single encryption by setting K1 = K2 to produce

$$C = S_{K1}\{S_{K1}^{-1}[S_{K1}(P)]\} = S_{K1}(P). \tag{4}$$

This allows users of the new (two key) system to decrypt data encrypted by users of the old (single key) system.

While the encryption technique (3) provides more security than simple double encryption as in (2), the new method can still be cryptanalyzed using a chosen plaintext attack [3] with about 2^{56} operations and 2^{56} words of memory. We therefore recommend that if triple encryption is used there be three independent keys. If compatibility with single encryption is desired, the operation can be taken to be

$$C = S_{K1}\{S_{K2}^{-1}[S_{K3}(P)]\}. \tag{5}$$

Then, when K1 = K2 = K3 = K, $C = S_K(P)$. Users could also be compatible with Tuchman's suggested method (4) by taking K1 = K3.

Although chosen plaintext attacks can sometimes be mounted on real systems, the following cryptanalysis of Tuchman's proposal should be viewed as a "certificational attack" which is only indicative of a weakness. Use of DES in accordance with proposed federal standards effectively prevents use of a chosen plaintext attack. History, littered with the broken remains of "unbreakable" ciphers, teaches extreme caution in certifying a new one [6], so that today even an indication of weakness is regarded as dangerous. In many cases, ciphers which have yielded to chosen plaintext attacks have later proven vulnerable to known plaintext or ci-

phertext only attacks as well.

We define some useful notation before describing the method of cryptanalysis:

$$\text{Enc(P)} = S_{K1}\{S_{K2}^{-1}[S_{K1}(\text{P})]\}, \quad (6)$$

$$M1 = S_{K1}(\text{P}), \quad (7)$$

$$M2 = S_{K2}^{-1}(M1) \quad (8)$$

$$= S_{K2}^{-1}(S_{K1}(\text{P})) \quad (9)$$

$$S_{K1}^{-1}(\text{C}). \quad (10)$$

$M1$ and $M2$ are intermediate values in the computation of C from P, as shown in Figure 1.

We motivate the method of cryptanalysis with the following observations:

If we knew K1 and a P–C pair, then it would be possible to compute the intermediate values $M1$ and $M2$ from (7) and (10). This would let us mount a known plaintext attack on K2 using (8). There are 2^{56} values of K1, so *if* we could quickly determine the right K2 once we found the right K1, then cryptanalysis would only take 2^{56} operations to search over K1. However, determining K2 using a known plaintext attack requires 2^{56} operations and would result in complexity 2^{112}.

The trick is to change the known plaintext attack on K2 to a chosen plaintext attack (that is, $M1$ is chosen), so we can quickly find K2 with a table lookup based on $M2$. This increases the memory needed to 2^{56} words, the same as is needed by the meet in the middle attack for simple double encryption.

For this attack to work, we must find the plaintext P_0 which results in $M1 = \mathbf{0}$. Equation (7) implies $\text{P}_0 = S_{K1}^{-1}(\mathbf{0})$, so deciphering $M1 = \mathbf{0}$ under all 2^{56} values of K1 is guaranteed to produce P_0. For each $\text{P} = S_i^{-1}(\mathbf{0})$ we therefore request Enc(P) = C (by the chosen plaintext assumption); compute $S_i^{-1}(\text{C}) = \hat{M}2$; and compute $\hat{\text{K}}2$ in one step from $\hat{M}2$ using the precomputed table. Since there are 2^{56} 64-bit values in the table, unicity distance arguments indicate a false alarm rate of 2^{-8} per value of K1 tried, or 2^{48} overall. Again, a single additional plaintext–ciphertext pair suffices to rule these out. The additional effort required is negligible compared to the basic search over 2^{56} K1's

Because $\text{P}_0 = S_{K1}^{-1}(\mathbf{0})$ and the corresponding $M2 = S_{K1}^{-1}(\mathbf{0})$ the algorithm can proceed as follows (Note: $\hat{M}2$ in step (1a) serves as both $\hat{M}2$ from (8) and as P_0 from (7)).

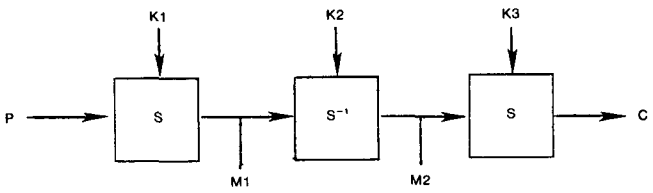
(1) For $i = 1$ to n Do
  (a) $\hat{M}2 = S_i^{-1}(\mathbf{0})$
  (b) Table[i] = $\langle \hat{M}2, i,$ "middle"$\rangle$
  (c) $\hat{M}2' = S_i^{-1}(\text{Enc}(S_i^{-1}(\mathbf{0})))$
  (d) Table[$n + i$] = $\langle \hat{M}2', i,$ "ends"$\rangle$

(2) Sort the table on the first field.

(3a) Search the table for adjacent entries of the form
$\langle$value, $\hat{\text{K}}2$, "middle"$\rangle$
$\langle$value, $\hat{\text{K}}1$, "ends"$\rangle$

(3b) Test to see if $\hat{\text{K}}1$ and $\hat{\text{K}}2$ are the correct keys by checking an additional plaintext–ciphertext pair.

Fig. 1. Diagram Illustrating Triple Encryption.



Conclusion

A second method of multiple encryption has been shown to be less secure than it first appeared. The weakness in both cases came from an ability to separate the key into two halves which did not interact. We conclude that all bits of the key should come into play repeatedly in a complex fashion as they do in the 56-bit DES and that multiple encryption with any cryptographic system is liable to be much less secure than a system designed originally for the longer key.

Received 10/79; revised 3/80; accepted 3/81.

References

1. Branstad, D.K., Gait, J., and Katzke, S. Report of the workshop on cryptography in support of computer security, National Bureau of Standards Rep. NBSIR 77-1291 (Sept. 21–22, 1976).
2. Diffie, W., and Hellman, M. Exhaustive cryptanalysis of the NBS data encryption standard. *Computer* (June 1977), 74–84.
3. Diffie, W., and Hellman, M.E. New directions in cryptography. *IEEE Trans. Info. IT-22*, 6 (Nov. 1976), 644–654.
4. Hellman, M.E., An extension of the Shannon theory approach to cryptography, *IEEE Trans. Info. IT-23*, 3 (May 1977), 289–294.
5. Hellman, M., Merkle, R., Schroeppel, R., Washington, L., Diffie, W., Pohlig, S., and Schweitzer, P. Results of an initial attempt to cryptanalyze the NBS data encryption standard. Information Systems Laboratory SEL 76-042 (Sept. 9, 1976).
6. Kahn, D. *The Codebreakers*. Macmillan, New York, 1976.
7. Kolata, G.B. Computer encryption and the National Security Agency. *Science 197* (July 29, 1977) 438–440.
8. Morris, R., Sloane, N.J.A., and Wyner, A.D. Assessment of the National Bureau of Standards proposed federal data encryption standard. *Cryptologia 1* (July 1977), 281–291.
9. National Bureau of Standards. Federal Information Processing Standards Publication No. 46, Jan 1977.
10. Senate Select Committee on Intelligence. Involvement of the NSA in the development of the data encryption standard. News release (Apr. 12, 1978).
11. Shannon, C.E. Communication theory of secrecy systems. *Bell. Syst. Tech. J. 28* (Oct. 1949), 656–715.
12. Tuchman, W.L. Talk presented at the Nat. Computer Conf., Anaheim, CA., June 1978.
13. Yasaki, E.K. Encryption algorithm: Key size is the thing. *Datamation 22*, 3 (Mar. 1976), 164–166.