# SUBTRACTIVE ENCRYPTORS - ALTERNATIVES TO THE DES

by

D. R. Morrison

Department of Computer Science

The University of New Mexico

Presented to the Workshop on Cryptology

UNIVERSITY OF CALIFORNIA - SANTA BARBARA

August 24, 1981

## SUBTRACTIVE ENCRYPTORS - AN ALTERNATIVE TO THE DES

In this paper, we introduce a family of encryptors, called subtractive encryptors, which includes the DES and other encryptors which, we submit, are easier to understand and to encode, and harder to cryptanalyze than the DES.

In suggesting alternatives to the DES, we are, of course, moving away from standardization of encryption. Standardization, like motherhood, is a highly desireable goal in appropriate circumstances, but totally undesireable in others. We believe that the protection of privacy is one of the others. If your goal is to protect your secrets from me, then it cannot be much comfort to you to know that my computer can execute the same programs as yours. And even less comfort to know that all or most of the people whose secrets I hope to steal, are using the same algorithm as you.

Keeping all the ducks in a row is more likely to benefit the foxes than the ducks.

1. Features borrowed from DES

   a. Subtractive transformations. These have the form

      $$W := f - W.$$

      A subtractive transformation is its own inverse.

   b. End-around shifts of vectors. A right end-around shift shifts each component of the vector one place to the right, and the rightmost component to the leftmost position. The inverse of an end-around shift is an end-around shift in the opposite direction.

c. Compositions of subtractive transformations and end-around shifts. These have inverses which are the composition of the inverses of their compqnents in reverse order.

d. Secret inputs to the computation of f used in the subtractive transformation. If the intruder does not know the value of f, then he can't perform the subtractive transformation.

e. Mixing operations in the computation of f. These insure that every part of the output depends on every part of the input. This discourages cryptanalysis by the divide-and-conquer technique, which seeks first to find part of the input by looking at part of the output, and then to find the rest of the input, with the first part known. A more subtle form of divide-and-conquer is one which applies a homomorphism of the operations used in the encryption, cryptanalyzes the homomorphic image, and then uses the result of this cryptanalysis to limit the search for the key or the input message. Use of the Chinese Remainder Algorithm is an example of this kind. Mixing should discourage this technique as well, by utilizing operations for which there are no homomorphisms but the identity.

2. The parameters.

The parameters of a subtractive encryptor are:

a. An interval (0..M-1) of integers, called words.

b. An operation, +, on pairs of words to words, such that (W, +) is a group.

c. An integer, n, called the block length. Members of the set $W^n$, of sequences of n words, are called blocks.

d. A function, f, called the scrambler function.

e. A positive integer, k, called the number of steps.

f. A sequence, $K_1\ K_2\ K_3\ \ldots\ K_k$ of words, called the keys.

g. A function E, called the enblocker, which maps messages into blocks, and

h. The inverse, D, of E, called the deblocker, which maps blocks in the range of E into messages.

M is chosen, typically, to be a little less than half the maximum integer upon which the basic arithmetic of the computer can be done. The operation, +, is, typically, addition mod M, or exclusive or or any other operation which the computer can perform easily, under which W is a group. The inverse of + is called -. If + is exclusive or then - is the same.

The encryptor operates as follows.

E converts a message to a block, B.

The encryptor converts B to a block, C, called the cipherblock.

The cipherblock is transmitted.

A decryptor, the inverse of the encryptor, converts C to B.

message → block → cipherblock → cipherblock → block → message
E encryptor transmitter decryptor D

The deblocker, D, converts B to the message. E and D are not secret. They are standard encodings of character strings, by means of their ASCII or other numerical equivalents.

3. The one-step encryptor.

A one-step encryptor has as its input, a block,

$$B = (W_1\ W_2\ W_3\ \ldots\ W_n)$$

and a key word, K. Its output is a block

$$B' = (W_2\ W_3\ W_4\ \ldots\ W_{n+1})$$

where

$$W_{n+1} = f(W_2\ W_3\ W_4\ \ldots\ W_n\ K) - W_1$$

The inverse of the one-step encryptor is the one-step decryptor, whose input is B' and whose output is B. It calculates $W_1$ by

$$W_1 = f(W_2\ W_3\ W_4\ \ldots\ W_n\ K) - W_{n+1}.$$

4. The k-step encryptor. The one-step encryptor is, of course, not useful by itself, especially if n is large, since n-1 of the words in the input block are still plainly visible in the output block.

The k-step encryptor, on the other hand, is a composition of k one-step encryptors, each having as its input, the output of the previous step, and each having its own key, $K_i$. Each step hides one more word of the original input, and after n steps, none of the original input remains visible.

The key-space for the k-step encryptor is $W^k$, the set of all sequences of k key words. The difficulty of cryptanalysis by exhaustive search of the key-space is proportional to the size of the key space. If it is larger than $2^{56}$ then it is more difficult than cryptanalysis of the DES, whose key space consists of all bit strings of length 56.

The difficulty of cryptanalysis by means other than exhaustive search depends, among other things, on the choice of f. For very simple choices of f, it may not be very difficult. Some of these choices are described in the next section.

5. The case f is affine. The scrambler function, f, is affine if

$$f = a_2W_2 + a_3W_3 + a_4W_4 + \ldots + a_nW_n + b$$

where the $a_i$ and b are constants or functions of K. In this case, cryptanalysis is not very difficult. An affine f is, therefore, not recommended.

In the affine case, we can represent B and B' as column vectors,

$$B = \begin{pmatrix} W_1 \\ W_2 \\ \ldots \\ W_n \\ 1 \end{pmatrix} \qquad B' = \begin{pmatrix} W_2 \\ W_3 \\ \ldots \\ W_{n+1} \\ 1 \end{pmatrix}.$$

The one-step encryptor, then becomes a simple matrix multiplication,

$$B' = A_i\, B,$$

where

$$A_i = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 \\ \ldots & & & & & \\ 0 & 0 & 0 & \ldots & 1 & 0 \\ -1 & a_2 & a_3 & \ldots & a_n & b \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix},$$

an $(n+1) \times (n+1)$ matrix with determinant $(-1)^n$.

The k-step encryptor is a composition of k such operations, equivalent to

$$\text{output} = A \text{ input},$$

where A is the product of the k matrices $A_i$. A is also, an $(n+1)x(n+1)$ matrix, with determinant $(-1)^{kn}$, and with coefficients that are, presumably, unknown to a cryptanalyst who does not know the keys. The cryptanalyst, however, does not need to know the keys if he can find the matrix A. Having a determinant of $\pm 1$, A is easily inverted, and the cryptanalyst can then decrypt by the formula.

$$\text{input} = A^{-1} \text{ output}$$

To solve for A, or for that matter, for $A^{-1}$, the cryptanalyst need only to have access to n linearly independent inputs and the corresponding n linearly independent outputs. Solving for A, or for $A^{-1}$ requires only the solution of an equation of the form

$$U = A \quad V$$

where U is a matrix whose colums are linearly independent outputs, and V is the matrix whose columns are the corresponding inputs.

6. Non-affine scramblers. Two very simple examples of non-affine scramblers are:

$$f = (W_1 W_2 + K) \bmod M$$

and

$$f = (W_2^2 + K.) \bmod M.$$

Either of these would be classified as a quadratic scrambler since f is a polynomial of degree 2 in the inputs. It follows

that a k-step encryptor built by concatenating k one-step encryptors with a quadratic scrambler creates an output whose words are polynomials of degree up to $2^k$ in the input words.

Since a polynomial of degree $2^k$ is a linear combination of products of up to $2^k$ input words, it follows that this encryptor can also be represented as a matrix multiplication. The cryptanalyst can decrypt if he can find the required matrix. In this case, however, the matrix is much larger than in the affine case, possibly of the order of $2^{nk}$. The number of steps required to solve such a system is on the order of $8^{nk}$. For any but the smallest values of n, this is more difficult than the exhaustive search of the key space.

We don't recommend the quadratic encryptor because multiplication of numbers close to max integer causes overflow if done in standard single precision. It may well be, however, that we can ignore the overflow and derive a "pseudo-quadratic" encryptor which is even more secure than the true quadratic. Whether this is the case or not, depends, among other things, on the manner in which overflow and mod for negative inputs are handled. Almost certainly, they introduce additional non-linearities into f. Although we cannot prove a theorem to this effect, it seems intuitive that this makes the cryptanalysis problem even more difficult. The constraints upon f that make the encryptor and decryptor work are only that f is repeatable and maps blocks into words. Possible system-dependent pitfalls are (1) that the overflow will halt the computation, or (2) that the overflow will

produce a negative result and that the operator mod M operating on a negative number will produce a result not in the range of words.

Barring these pitfalls, it seems that the pseudo-quadratic encryptors, for all but the smallest values of n and k are simpler and at least as secure as the DES.

We have, of course, only discussed upper bounds, not lower bounds on the complexity of cryptanalysis. Lower bounds on complexity are very difficult to come by. Even the most sophisticated results in complexity theory relate the lower bounds on complexity of one problem to unknown lower bounds on complexity of another. Thus our failure to present such a lower bound is not really a reason to discard the suggested algorithm. Any other encryptor, including the DES shares the same flaw.

If, for some reason, the pseudo-quadratic encryptor is of doubtful strength, there is a multitude of other readily available non-linear operations on words. The boolean operators and integer division operators and step functions are examples. Any one of these alone will not provide much security, but most of them, like the quadratic or pseudo-quadratic operations rapidly increase the complexity of cryptanalysis as k increases.

7. A Simple Example. Following is an example of a subtractive encryptor which appears to be simpler and stronger than the DES. We can obviously not certify that it is stronger because we don't know how strong either one of them is. If it is true that neither

can be cryptanalyzed by any method more efficient than exhaustive search, then the method to be shown is stronger because it has a larger key space.

Parameters:

$M = 2^{30}$. (Words are 30-bit positive integers.)

+ is addition mod M

n = 2 (A block is two words.)

k = 4 (The key is four words. The key space has size $2^{120}$.)

f is described below.

Each key, $k_i$ is first converted to a pair of subkeys, $m_1$, the modulator and $m_2$, the second modulator, as follows.

$$m_1 := k_i \text{ div } 2^{16} + 2^{14}$$

$$m_2 := k_i \bmod 2^{14} + 2^{14}$$

The scrambler function, f, is then computed by

$$f(W_2, m_1, m_2) := ((W_2 \bmod m_1)\ (W_2 \bmod m_2) + k_i) \bmod M$$

In a 32-bit machine, this computation produces no overflow, and the result is in the range 0..M-1. Every bit in the output depends on every bit in the input, and this discourages the divide-and-conquer technique of cryptanalysis. While there are non-trival homomorphisms on the integers mod $m_i$ for any particular $m_i$, there are none known to this author which are simultaneous homomorphisms on the integers mod $m_i$ for all $m_i$. Since the $m_i$ are unknown to the cryptanalyst, and there are several of them involved in each encryption, divide-and-conquer appears to be an ineffective tool for the cryptanalyst.

The function f involves, for a fixed $k_i$, a product of two sawtooth functions of the input $W_1$, with periods $m_1$ and $m_2$ and amplitude $m_1$ and $m_2$. This is a highly non-linear function and one which cannot be represented by low order polynomials, nor by linear or affine functions.

The purpose of the added term $2^{14}$ in the formula for $m_i$ is to insure that $m_i$ is not a very small number, and, in particular, is not zero. If $m_i$ were zero, then mod $m_i$ is undefined. If $m_i$ is small, say 2 or 3, then one of the sawtooth functions which defines f would have a very small range, and a method of cryptanalysis might be developed which exploits this small range by an exhaustive search of its possible values.

The purpose of confining $m_1$ and $m_2$ to values less than $2^{15}$ is to preclude very long ramps of the sawtooth functions. If each ramp covers a substantial part of the range of f, then there is a possibility that for substantial numbers of messages, all the values of f fall on one ramp, and the quadratic analysis will apply.

As noted above, we make no claim that a cryptanalysis of this requires an exhaustive search of the key space. There may be a more efficient cryptanalysis or there may not. But this is all we know about the DES. If the only method of cryptanalyzing either is an exhaustive search of the key space, then the suggested encryptor is stronger than the DES.