PROJECT MAC

May 23, 1975

Computer Systems Research Division

Request for Comments No. 77

SOME COMMENTS ON THE NBS STANDARD ENCRYPTION ALGORITHM

by J. H. Saltzer

In the Federal Register of March 17, 1975, the National Bureau of Standards published a proposed standard data encryption algorithm, based on a simplified version of the IBM "Lucifer" algorithm. Following are some observations upon first reading:

1. The algorithm uses a 56-bit key, allowing about 10^{17} different keys. Considering the speed of modern data processing equipment, these numbers seem too small. If one can acquire (or guess) a sample of clear text known to correspond to a particular enciphered block, a brute force exploration of the key space is just barely feasible. Assume, for example, that the technology used to build a single enciphering chip for $10 is used to build 10^5 such chips for $1M. These 10^5 chips can be put to work in parallel, exploring the key space. If each chip can do one deciphering operation per microsecond, the parallel system can do 10^{17} trials in only ten days. Ten days seems a little too close for comfort, since there may be mathematical transformations that can reduce the work factor by several orders of magnitude.

2. There are several observations related to the problem that the NBS standard algorithm is only part of a larger system, and must be used correctly and carefully within that larger system, or else its strength will be reduced. It would seem that the standard should at least provide discussion of how to use it safely. For example:

   a. The user of this algorithm must be very careful to also use a truly random number generator for his keys. The reason is that if there is any pattern or redundancy whatsoever in his keys, that pattern or redundancy can be used to reduce the work factor of the brute force search of the key space. For example, suppose an attacker has available a sample of a key known to belong to one user of a time-sharing system. Starting with this key and using some knowledge of pseudo-random number generation techniques he may be able to quickly guess a probable range of values of other keys generated at about the same time.

---

b. This algorithm is basically a simple substitution cipher, with a very large alphabet. It is important to its strength that the system spread its usage widely across that alphabet, to avoid possible frequency analysis. For example, if the algorithm is used to secure a full-duplex, character-at-a-time communication link to a terminal, some special measure, such as padding out the one-character messages with changing random numbers, is necessary. Otherwise, the frequency distribution of the underlying communication will show through in the cipher text.

c. Since the proposed standard is a block cipher, it is essential that it be used in a systematic way that guarantees authenticity of all enciphered blocks. Otherwise, there is no way to discover that an interloper has modified some blocks, or possibly replaced them by other blocks enciphered with the same key. The original Lucifer papers proposed a systematic way of initial authentication and of chaining blocks together to counter this threat.[1] Some such strategy should perhaps be part of this standard.

[1] Feistel, H., "Cryptographic coding for databank privacy," IBM Research Report RC 2827, March 18, 1970.