June 9, 1987

(Manuscript of invited paper for E & M, the journal of the Oesterreichische Verband für Elektrotechnik)

PROBABILISTIC ENCIPHERMENT

James L. Massey
Inst. for Signal & Info. Proc.
Swiss Federal Institute of Technology
CH-8092 Zürich

Abstract

An encipherment process is probabilistic if the plaintext and secret key do not determine the ciphertext uniquely. A "randomizer" is introduced to account for the randomness of the enciphering process. It is shown that plaintext padding and homophonic substitution are probabilistic encipherment techniques for which the randomizer plays the role of "extra plaintext". Two examples are given of computationally-secure probabilistic ciphers for which the randomizer plays the role of "extra secret key".

I. INTRODUCTION

In a secret-key cryptosystem, the plaintext, $\underline{X}$, and the secret key, $\underline{Z}$, are random variables. The statistics of $\underline{X}$ depend on the nature of the plaintext source, whereas the statistics of $\underline{Z}$ are under the control of the cryptographer. In a conventional cipher, however, the enciphering process itself is deterministic, i.e., the ciphertext, $\underline{Y}$, is uniquely determined by $\underline{X}$ and $\underline{Z}$. This enciphering function is usually written as

$$\underline{Y} = E_{\underline{Z}}(\underline{X}) \tag{1}$$

to emphasize that one may think of a particular value, say $\underline{z}$, of the secret key, $\underline{Z}$, as selecting the particular enciphering function $E_{\underline{Z}}(.)$ that is applied to the plaintext, $\underline{X}$, to produce the ciphertext, $\underline{Y}$. The only general restriction on these enciphering functions is that they must be invertible,

i.e. $E_{\underline{z}}(.)$ must have an inverse $D_{\underline{z}}(.)$ for each allowed key $\underline{z}$; hence, one can write

$$\underline{X} = D_{\underline{Z}}(\underline{Y}). \tag{2}$$

Our intention in this paper is to point out the usefulness of making the enciphering process probabilistic, i.e., of making it such that $\underline{X}$ and $\underline{Z}$ do not uniquely determine the ciphertext, $\underline{Y}$. To describe such probabilistic encipherment conveniently, we introduce a random variable $\underline{R}$, which we shall call the randomizer. The statistics of the randomizer, like those of the secret key, are under the control of the cryptographer. The essential feature that distinguishes the randomizer, $\underline{R}$, from the secret key, $\underline{Z}$, is that $\underline{R}$ is known only at the encrypting site whereas $\underline{Z}$ is known at both the encrypting and decrypting sites. We demand that $\underline{X}$, $\underline{Z}$ and $\underline{R}$ uniquely determine $\underline{Y}$, i.e., that $\underline{R}$ account for all the randomness of the enciphering process. This enciphering process can thus equally well be described by

$$\underline{Y} = E_{\underline{Z}}(\underline{X},\underline{R}) \tag{3a}$$

or by

$$\underline{Y} = E_{\underline{Z}\underline{R}}(\underline{X}). \tag{3b}$$

The only general restriction on this enciphering function is that it must be possible to recover $\underline{X}$ at the decrypting site, i.e., with knowledge of only $\underline{Y}$ and $\underline{Z}$, i.e., we again require that

$$\underline{X} = D_{\underline{Z}}(\underline{Y}).$$

[It will often happen that $\underline{R}$ can also be recovered at the decrypting site, but we do not wish to make this a condition on probabilistic encipherment.] The form (3a) of the equation for probabilistic encipherment suggests that the randomizer, $\underline{R}$, can be considered as "extra plaintext", but this extra plaintext need not be recoverable from the ciphertext and secret key. The form (3b) suggests that the randomizer can alternatively be considered as "extra secret key", but this extra secret key is not known in advance at the decrypting site. These dual roles of the randomizer correspond to the two

essentially different ways that the randomizer can serve the cryptographer and that we explore in the following sections.

II. THE RANDOMIZER AS EXTRA PLAINTEXT

Consider a substitution cipher in which the plaintext $\underline{X}$ and ciphertext $\underline{Y}$ are binary sequences of the same length N. The Data Encryption Standard (DES) in its "electronic codebook mode" is such a cipher with N = 64, i.e., it is a substitution cipher on an alphabet of 2^{64} letters. Such a large alphabet is necessary to thwart a simple frequency-analysis attack on the cipher. But suppose that the plaintexts that one wishes to encipher with DES are single 8-bit bytes that have only $2^8 = 256$ different values. Every competent cryptographer would "pad" the plaintext with a random sequence $\underline{R}$ of 56 bits in this situation, and would thus be making use of probabilistic encipherment. Such "plaintext padding" is an old cryptographic trick, but its power goes beyond what is generally realized. For suppose that the true plaintext of length n has redundancy ρ, i.e., it contains only $(1-\rho)$ bits of information or, equivalently, has an entropy of only $(1-\rho)n$ bits, cf. [1]. The padding with N-n completely random bits gives the padded plaintext an entropy of $N-n + (1-\rho)n$ bits and hence reduces the redundancy to only

$$\rho' = \frac{n}{N}\,\rho. \tag{4}$$

For our DES example with n = 8 and N = 64, the plaintext redundancy is reduced by a factor of 8 or, equivalently, Shannon's unicity distance [1] is increased by a factor of 8. It is much harder to break the padded cipher than its unpadded version! The price, of course, is that the amount of true plaintext sent with each cryptogram is reduced by the factor n/N, but this may be an acceptable tradeoff when great security is required.

A less obvious way to use the randomizer as extra plaintext is by "homophonic substitution", which we illustrate with an example. Suppose the true plaintext is a sequence of statistically independent letters from an alphabet of 3 letters, say {A,B,C}. Suppose further that A,B and C have probabilities 1/4, 1/4 and 1/2, respectively. A single plaintext letter then has entropy 3/2 bits. If we represent these letters in binary as, say,

A = 00, B = 01 and C = 10, the plaintext redundancy ρ will be 1/4. However, if we choose to represent C by either 10 or 11, making a completely random choice from these two "homophones" each time C appears, we create a binary sequence with zero redundancy because each pair of binary digits is equally likely to be any of the four possibilities. No matter what cipher we then use to encrypt this randomized plaintext, the unicity distance will be infinite! At the decrypting site, one merely maps both 10 and 11 to C to recover the true plaintext without knowing the randomizer in advance.

Homphonic substitution is the general technique of mapping letters from a smaller alphabet (3 letters in our example) randomly onto a longer alphabet (4 letters in our example) in such a way that no two letters in the smaller alphabet are mapped to the same letter in the larger alphabet. The goal is to make the letters in the larger alphabet as nearly equally probable as possible and thus to make single-letter frequency analysis by the cryptanalyst worthless.

III. THE RANDOMIZER AS EXTRA SECRET KEY

The real power of random encipherment, in our opinion, is obtained when one considers the randomizer as "extra secret key" that is not known in advance at the decrypting site. The Rip van Winkle cipher [2] that we proposed earlier together with I. Ingemarsson is one illustration of such use of the randomizer. We give now two further illustrations.

We first describe a provably computationally-secure scheme due to W. Diffie [3] in unpublished work. Suppose that there are K + 1 telephone lines from the sender to the legitimate receiver, and that the sender sends completely random binary sequences over the first K lines but, over the last line, sends the modulo-two sum of the binary plaintext and a randomly-chosen one of the K random binary sequences. The secret key $\underline{Z}$ is the telephone number of the line with the chosen random sequence and thus need only be $\lceil \log_2 K \rceil$ bits long. The plaintext has effectively been enciphered with a "one-time pad" whose additive key stream has also been sent in the open. The enemy cryptanalyst has no choice but to try the telephone numbers one at a time, adding the sequence at each number to the (K+1)st sequence moduo-two until he re-

covers meaningful plaintext. On the average, he tries K/2 such numbers before succeeding, and he can do no better. For a 100 bit key, i.e., for $\log_2 K = 100$, he must try about $2^{99} = 10^{30}$ telephone numbers before succeeding. This cipher system is thus provably computationally-secure, but totally impractical because of the vast amount of data (K+1 times the length of the plaintext) that must be transmitted.

For Diffie's scheme, the randomizer R consists of all K random sequences sent over the first K telephone lines. The randomizer is not kept secret at all, but the true secret key selects that part of the randomizer that is used to hide the plaintext; it thus seems appropriate to say, as we have done, that the randomizer is serving as "extra secret key".

We now describe a variation of Diffie's scheme that comes slightly closer to practicality. Suppose the cryptographer generates K random binary digits and sends these serially to the receiver. He then sends a one-bit plaintext to which he adds modulo-two a randomly-chosen one of the K random bits, say the k-th. So far there is no difference from Diffie's scheme. But now the cryptographer sends a second one-bit plaintext to which he adds modulo-two the (k+1)-th random bit, he then sends a third one-bit plaintext message to which he adds modulo-two the (k+2)-nd random bit, etc., until he has sent K one-bit plaintext messages. Here the (K+1)-th random bit is identified with the first random bit, the (K+2)-th random bit with the second random bit, etc. There appears to be no shortcut to the cryptanalyst's trying each of the K random bits one at a time to see which has been added to the first one-bit plaintext message -- but we have not tried to make a strict proof that this is so. The cryptanalyst seemingly must again try about K/2 locations before succeeding, whereas again the secret key is only $\lceil \log_2 K \rceil$ bits long. But for this scheme, only K bits of random sequence are transmitted, the same as the total length of the plaintext messages, so the transmission requirements are only doubled rather than multiplied by K+1 as in Diffie's original scheme.

The only problem with the new scheme is that K must be quite large, say $K \approx 10^{20}$, to obtain the amount of security that most applications of cryptography demand. Thus, transmitting the initial K random bits would impose an

intolerable delay before the plaintext could begin to be sent and decrypted. In fact, this system now closely resembles the Rip van Winkle cipher [2], which also imposes a similar intolerable delay but whose computational security has been strictly proved.

It has been our purpose in this section to show how randomization can be used to provide the basic cryptographic strength of a cipher. It is our conviction that such use of randomization offers the best chance among present research directions of leading to practical, provably computationally-secure ciphers with small keys, and it is our hope that this paper may stimulate the search for such ciphers.

REFERENCES

[1] C.E. Shannon, "Communication Theory of Secrecy Systems", Bell Sys. Tec. J., vol. 28, pp. 656-715, Oct. 1949.

[2] J.L. Massey and I. Ingemarsson, "The Rip van Winkle Cipher -- A Simple and Provably Computationally Secure Cipher with a Finite Key", Abstracts of Papers, IEEE Int.Symp.Info.Th., Brighton, England, June 24-28, 1985, p. 146.

[3] W. Diffie, Private Communication, July 1984.