THE IMPACT OF TECHNOLOGY ON CRYPTOGRAPHY*

Ronald L. Rivest
M.I.T. Laboratory for Computer Science
Cambridge, Massachusetts 02139

Abstract

We examine aspects of recent technological developments, in particular the development of large semiconductor memories, on enciphering techniques.

I. Introduction

The invention of electronic communications media has created a concurrent need for the invention of cryptographic techniques to ensure the privacy of messages transmitted on those media. Indeed, Kahn [Ka67] attributes the creation and growth of modern cryptography in large part to the new communications technologies:

"The telegraph made cryptography what it is today." [Ka67, p. 189]

"The First World War marks the great turning point in the history of cryptography. Before, it was a small field; after, it was big. ... The direct cause of this development was the enormous increase in radio communications." [Ka67, p. 348]

The invention of the telegraph (Morse, 1840) and the radio (Marconi, 1895) provided inexpensive communication on a global scale. However, radio (and to a much lesser extent telegraphy) suffers from being a "broadcast" communications channel: it is very easy to intercept radio transmissions. Because of this defect, cryptography has blossomed during the 20th century as the way to achieve private communications.

The field is unfortunately littered with examples of enciphering techniques that failed. Time and again a new technique would be put in use, only to be broken a few months later. The breaking of the German "Enigma" code during World War II is probably the best-known example. The breaking of this code had an enormous influence on the course of the war.

These repeated failures can be attributed largely to the lack of adequate computational resources. That is, the available technologies for performing the enciphering and deciphering transformations were not sufficiently advanced to permit the practical implementation of complicated functions. With a little bit of theory, some luck, and a slight edge in computing power, the cryptanalyst could unravel the relatively simple enciphering techniques employed.

To express our point another way, we may assert that of the twin sisters of information science (communications and computation) it has been communications that has developed far more rapidly and extensively. The technology to compute quickly and cheaply has only developed more recently. During World War II, for example, while global radio communication was taking place, some of the most advanced enciphering was done using relatively primitive mechanical rotor techniques. (The Enigma used such rotors.)

*Supported by NSF grant MCS76-14294

The importance of computational ability for effective cryptography is easily discovered. The only unconditionally secure cryptographic system (the "one-time pad") requires enormous amounts of key; however, practical systems are forced to use short keys. Shannon has shown [Sh49] that it is possible in theory to recover a short key from a long ciphertext if the message has redundancy (as English text does). The only thing that keeps the cryptanalyst from performing in practice what Shannon assures him he can do in theory is the intrinsic computational difficulty of doing so. A cryptographic system which is not unconditionally secure (that is, which uses short or fixed-length keys), but which poses inordinate computational difficulties, may be considered "unbreakable in practice", or "computationally secure." It seems to be the case that any enciphering scheme which is computationally secure requires a fairly complex enciphering function.

The relative backwardness of computing technology as compared with communications technology has now been eliminated, due to developments in semiconductor technology. Today, a single $20 microprocessor chip provides more computing power than was available in the entire world in 1940, and a single 16K memory chip costing $20 can store as much information as a room full of relays could store then. Much of this progress has occurred in the last 10 years: the first microprocessor was introduced in 1971 (the Intel 4004) and 1K read/write memories became available in 1975.

We would like to know how this incredible availability of cheap computing power will affect cryptography. Will it enable the design of computationally secure enciphering techniques, or does it only aid the cryptanalyst to the same extent that it aids the cryptographer?

We will present evidence for the case that the availability of cheap semiconductor memories and microprocessors is overwhelmingly to the advantage of the cryptographer. There is no reason to expect that with careful design, codes can not be constructed which will defy any attempt at cryptanalysis, and that these codes can be implemented very cheaply with current technology.

There is perhaps a measure of irony in this, since the cryptanalysts have been among those responsible for the technological advances. The Colossus [Ra76] was perhaps the world's first operational computer; it was built to help decipher the Enigma code. Since World War II the National Security Agency, responsible for cryptanalytic (and cryptographic) efforts in the U.S., has reportedly been very active in supporting and purchasing advanced computers, and thus has no doubt contributed to the technological breakthroughs described above.

We present our evidence in the form of a simple

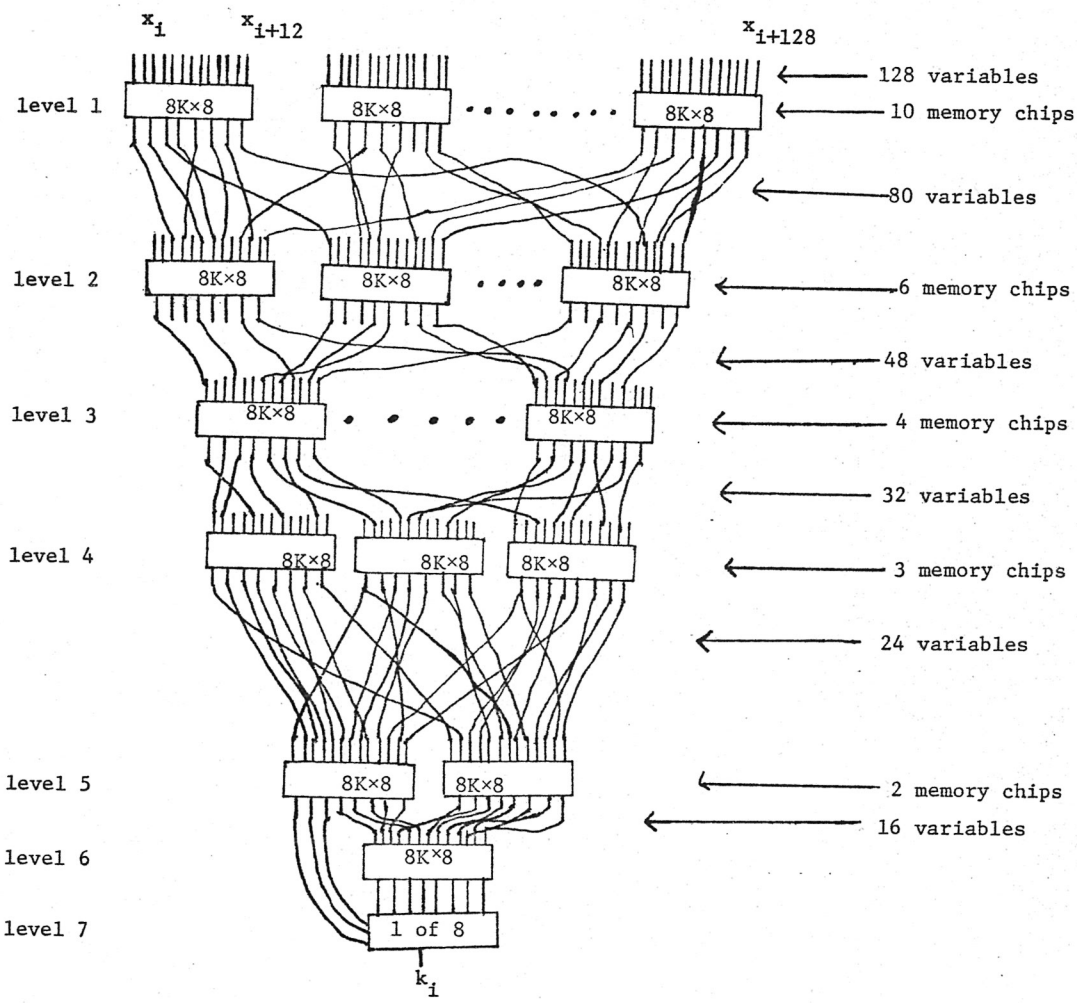
x_i
x_{i+12}
x_{i+128}
level 1
level 2
level 3
level 4
level 5
level 6
level 7
8K×8
1 of 8
k_i
128 variables
10 memory chips
80 variables
6 memory chips
48 variables
4 memory chips
32 variables
3 memory chips
24 variables
2 memory chips
16 variables

Figure 5.

example, showing how a well-known cryptographic system (the Vernam system) could be implemented with current technology.

Consider the Vernam system depicted in Figure 1.

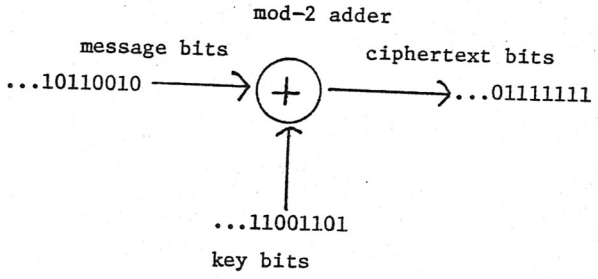

Figure 1.

Here each message bit m_i is added (modulo 2) to a key bit k_i to obtain a ciphertext bit c_i. Since we need as many key bits as message bits, some way must be found to expand a short key into a long key if we are to avoid having to use a long key. We depict this in Figure 2; here an n-bit key $s_1,s_2,\ldots s_n$ is expanded

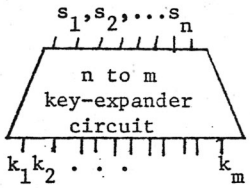

Figure 2.

into a key $k_1,\ldots,k_m$ of length m. One poor way of doing this is merely to repeat the short key $s_1,\ldots,s_n$ over and over. Vernam combined a repetition of a 999-bit key with a 1000-bit key to obtain a key pattern that was periodic with period 999,000. Either way, however, discovery of a few key bits $k_i,\ldots,k_j$ gives much information about other key bits, so that deciphering (or even guessing) a fragment of the message would enable a large part of the entire message to be decrypted.

How might one design a key expander using modern technology? That is, what would Vernam have done with his system if he had access to microprocessors and memory chips? We shall see that a key-expander (which might also be called a good pseudo-random number generator) is easily designed.

A large value of m is required if the key-expander is to produce sufficient key bits; suppose we select $m=2^{128}-1$. A sequence of this period can be easily produced by a linear shift-register technique, [Go67] illustrated in Figure 3. Here the i+128th bit x_{i+128} of the sequence is a linear combination (mod 2) of a fixed subset of the preceding 128 bits; the initializing vector $x_1\ldots x_{128}$ is the "short key" which is expanded.

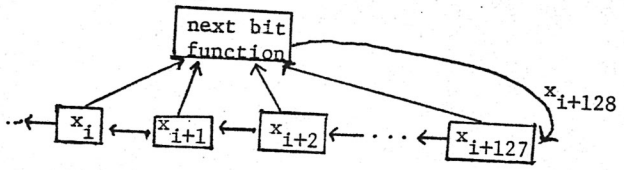

Figure 3.

It is well-known that the sequence produced by a linear shift register is very weak when used in a Vernam system [Me76]; how can one transform its output to yield a stronger code? We propose the use of a "compressor circuit" which computes a very complicated function of the shift register bits $x_i,\ldots,x_{i+127}$ to yield k_i as in Figure 4. Instead of using $k_i=x_i$, we let

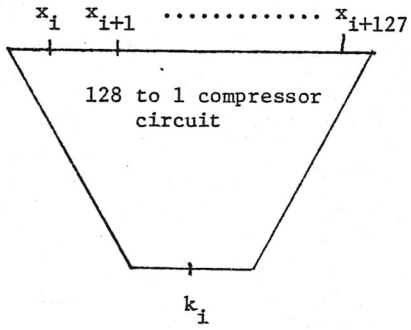

Figure 4.

k_i depend on all of $x_i,\ldots x_{i+127}$ in such a complicated way that even knowledge of a large subsequence of $k_i,\ldots,k_{i+j}$ of the key bits will not easily determine k_{i+j+1} or any of $x_i\ldots x_{i+j}$.

Clearly, a 2^{128}-bit memory chip, with 128 address lines and one data output line, would form an ideal compressor circuit (assuming it was filled with random bits). Since this is impossible, one might reasonably consider implementing a compressor with several layers of memory chips, as indicated in Figure 5. Here each of the 26 memory chips is an 8K×8 memory chip holdin 65K bits of information. The interconnections between the various levels are random, and the memory chips each contain a different random pattern of 65K bits.

This sort of mixture of random permutations (the interconnections) with random substitutions (the memory chips) has been proposed before. For example Lucifer, and the NBS Data Encryption Standard system [Fe73, Fe75] use exactly this philosophy, although there only small substitutions are employed (they have only 6-bits of input each, rather than 13). The reason such a scheme is so attractive can perhaps partly be explained by the following theorem.

Theorem 1: Given a knowledge of the size n of a linear shift register, the interconnection pattern and memory contents of a compressor circuit attached to its outputs, and a knowledge of key bits $k_1,\ldots,k_i$, it is a NP-complete problem to find any initial state

vector $x_1,\ldots,x_n$ for the shift register that would yield $k_1,\ldots,k_i$ as outputs.

That is, knowledge of the output does not easily imply knowledge of the state. Even better, we have the following.

Theorem 2. With the givens of Theorem 1, it is an NP-complete problem to determine whether another key bit k_{i+j} could possible have the value 0.

Thus, knowledge of some key bits does not help much in determining what other key bits might be. This property shows that adding a compressor circuit to the outputs of a linear shift register can produce a cipher which is much more difficult to break.

The extensive use of memory circuits in the design of the compressor circuit is not coincidental. Memories are in many ways ideal for this purpose, since they can implement any function with the given number of inputs and outputs. The problem for the cryptanalyst is thus raised to a level that is very difficult.

Summary

We have seen that it seems to be a relatively straightforward matter to construct a pseudo-random number generator with strong cryptographic properties using modern memory technology. While the Vernam system is probably not the cryptographic system of choice today (recent developments [Di76, Ri77] look very promising), this simple example shows how the availability of memory chips enables the simple Vernam system to be transformed into a much stronger system. It is worth mentioning here that the evaluation of the modified Vernam system discussed here is based only on publicly available information; it may be the case that the classified literature on cryptography (to which the author has no access) would throw much light on the potential strengths and weaknesses of this system.

As another example of how the advances in semiconductor technology are favoring the cryptographer, we observe that the cryptographic system of Rivest, Shamir, and Adleman [Ri77], can now be implemented on a few LSI chips, whereas 10 years ago the enciphering computation could only have been done on a large computer.

Roughly we may estimate that the cryptanalysts task should grow exponentially with the amount of work and key used in enciphering. Of course, the design of any cryptographic system should involve the application of any relevant theory (computational complexity, for example) to find weaknesses. With the availability of cheap, powerful computing hardware and the right theoretical tools, developing practical cryptographic systems which are computationally secure becomes possible.

References

[Ka76] Kahn, David. The Codebreakers. (MacMillan Publishing Co., New York).

[Sh49] Shannon, C.E. "Communications Theory of Secrecy Systems" Bell Systems Technical Journal 28 (Oct. 1949), 656-715.

[Ra76] Randell, Brian. "The Colossus" Proceedings of the International Conference on History of Computing. Los Alomos Science Laboratory, University of California, Los Alomos, New Mexico. 1976.

[Fe73] Feistel, Horst. "Cryptography and Computer Privacy," Scientific American 228 (May 1973), 15-23.

[Fe75] Federal Register, 40 (August 1, 1975).

[Di76] Diffie , W. and M. Hellman. "New Directions in Cryptography," IEEE Transactions on Information Theory, IT-22 (Nov. 1976), 644-654.

[Ri77] Rivest, R.L. A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM vol. 21 no. 2 (Feb. 1978),120-126.

[Go67] Golomb, S. "Shift Register Sequences." Modern Algebra Series (Holden Day, 1967).