Symmetric and Asymmetric Encryption

GUSTAVUS J. SIMMONS

Sandia Laboratories, Albuquerque, New Mexico 87185

All cryptosystems currently in use are symmetric in the sense that they require the transmitter and receiver to share, in secret, either the same piece of information (key) or one of a pair of related keys easily computed from each other, the key is used in the encryption process to introduce uncertainty to an unauthorized receiver. Not only is an asymmetric encryption system one in which the transmitter and receiver keys are different, but in addition it is computationally infeasible to compute at least one from the other. Asymmetric systems make it possible to authenticate messages whose contents must be revealed to an opponent or allow a transmitter whose key has been compromised to communicate in privacy to a receiver whose key has been kept secret—neither of which is possible using a symmetric cryptosystem.

This paper opens with a brief discussion of encryption principles and then proceeds to a comprehensive discussion of the asymmetric encryption/decryption channel and its application in secure communications.

Keywords and Phrases: cryptography, secure communications, asymmetric encryption, computational complexity, public-key cryptosystems, authentication

CR Categories. 3.81, 5.25, 5.6

INTRODUCTION

The object of secure communications has been to provide privacy or secrecy, i.e., to hide the contents of a publicly exposed message from unauthorized recipients. In contemporary commercial and diplomatic applications, however, it is frequently of equal or even greater concern that the receiver be able to verify that the message has not been modified during transmission or that it is not a counterfeit from an unauthorized transmitter. In at least one important class of problems message authentication is needed at the same time that the message itself is revealed.

In this paper secure communications are discussed with emphasis on applications that cannot be satisfactorily handled by present cryptographic techniques. Fortunately, an entirely new concept—the asymmetric encryption/decryption channel—solves the new requirements in secure communications. For perspective, the reader should keep in mind that all current cryptosystems are *symmetric* in the sense that either the same piece of information (key) is held in secret by both communicants, or else that each communicant holds one from a pair of related keys where either key is easily derivable from the other. These secret keys are used in the encryption process to introduce uncertainty (to the unauthorized receiver), which can be removed in the process of decryption by an authorized receiver using his copy of the key or the "inverse key." This means, of course, that if a key is compromised, further secure communications are impossible with that key. The new cryptosystems are *asymmetric* in the sense that the transmitter and receiver hold different keys at least one of which it is computationally infeasible to derive from the other.

This article was sponsored by the U.S Department of Energy under Contract DE-AC04-76DP00789.

CONTENTS

It is possible to communicate in secrecy and to "sign" digital messages using either symmetric or asymmetric techniques if both the receiver and transmitter keys can be secret. One of these functions can be accomplished with an asymmetric system even though the transmitter or the receiver key has been revealed. It is also possible to communicate privately without a prior covert exchange of keys and to authenticate messages even when the contents cannot be concealed from an opponent—neither of which is possible with a symmetric cryptosystem. The current revolution in secure communications is based on the ability to secure communications even when one terminal (and the key) is located in a physically unsecured installation.

1. CLASSICAL CRYPTOGRAPHY

Classical cryptography seeks to prevent an unauthorized (unintended) recipient from determining the content of the message. In this section we illustrate the concepts of all cryptosystems, such as key, stream or block ciphers, and unicity point. A more detailed account can be found in the paper by Lempel [LEMP79] and in Kahn's encyclopedic *The Codebreakers, the Story of Secret Writing* [KAHN67].

A primitive distinction among cryptosystems is the structural classification into stream and block ciphers. The plaintext message is a sequence of symbols from some alphabet $\mathscr{A}$ (letters or numbers). A stream cipher operates on the plaintext symbol by symbol to produce a sequence of cipher symbols from an alphabet $\mathscr{C}$. ($\mathscr{C}$ and $\mathscr{A}$ are frequently the same.) Symbolically, if π is a nonsingular mapping $\pi: \mathscr{A} \rightarrow \mathscr{C}$, and M is a plaintext message

$$M = (a_1 a_2 \cdots a_k \mid a_i \in \mathscr{A}),$$

then the stream cipher $C = \pi(M)$ is given by

$$C = (\pi(a_1), \pi(a_2), \ldots, \pi(a_k) \mid \pi(a_i) \in \mathscr{C}).$$

The mapping π is commonly a function of previous inputs—as in the rotor cryptomachines of the World War II period. The various versions of Vigenére encryption to be discussed shortly are all examples of stream ciphers, some of which use a fixed mapping and others, such as the running key and autokey systems, a usage-dependent mapping.

In a block cipher a block of symbols from $\mathscr{A}$ is operated on jointly by the encryption algorithm, so that in general one may view a block cipher as a nonsingular[1] mapping from the set of plaintext n-tuples $\mathscr{A}^n$ into the set of cipher n-tuples $\mathscr{C}^n$. For cryptosystems which use the same key repeatedly, block ciphers are cryptographically stronger than stream ciphers. Consequently, most contemporary cryptosystems are block ciphers, although one-time key systems are used in applications where the very highest security is required. Examples of block ciphers are the Playfair digraph substitution technique, the Hill linear transformation scheme, and the NBS Data Encryption Standard (DES). The distinction between block and stream ciphers is more apparent than real since a block cipher on n-tuples from $\mathscr{A}$ is equivalent to a stream cipher over the enlarged alphabet $\mathscr{A}^n$.

Since much of the discussion relies on the concept of a "key" in the cryptosystem, we shall present several examples that illustrate keys and possible attacks to discover them.

[1] Nonsingular simply means that every cipher decrypts to a unique message. In Section 6.2 an example of a singular cryptomapping is described.

In the most general terms possible, an encryption system must combine two elements: some information—called the *key*—known only to the authorized communicants, and an *algorithm* which operates on this key and the message (plaintext) to produce the cipher. The authorized receiver, knowing the key, must be able to recover the message (decrypt the cipher); an unauthorized receiver should not be able to deduce either the message or the unknown key. The key as defined here is very general: It is the total equivocation of everything that is kept secret from an opposing cryptanalyst. By this definition, a key can be much longer than the bit stream serving as the key in some cryptodevices.

The encryption algorithm must be so constructed that even if it becomes known to the opponent, it gives no help in determining either the plaintext messages or the key. This principle, first formulated by Kerchoffs in 1883, is now universally assumed in determining the security of cryptosystems.

Preprocessing a text by encoding into some other set of symbols or symbol groups by an unvarying rule is not considered to be a part of the encryption process, even though the preprocessing may complicate the cryptanalyst's task. For example, The Acme Commercial Code [ACME23] replaces entire phrases and sentences by five-letter groups; the preprocessed text EJEHS OHAOR CZUPA, which is derived from (BUDDY) (CAN YOU SPARE) ((A) DIME(S)), would be as baffling to the cryptanalyst as a cipher. Continued use of fixed preprocessing codes, however, destroys this apparent cryptosecurity, which is therefore considered to be nonexistent from the beginning. Common operations which compress text by deleting superfluous symbols or expand text with null symbols are considered to be part of the encoding of the text rather than part of the encryption process.

The encryption process itself consists of two primary operations and their combinations, *substitution* and *transposition*.[2] A substitution cipher or *cryptogram* simply replaces each plaintext symbol by a cipher symbol; the key specifies the mapping. An example is the Caesar cipher, in which each letter is replaced by the letter occurring k places later in the alphabet (considered cyclically); when $k = 3$,

COMPUTING SURVEYS
= FRPSXWLQJ VXUYHBV.

Simple transposition permutes symbols in the plaintext. The permutation is the key. For example, if the permutation $(15327468)^3$ is applied to the two blocks of eight symbols above,

COMPUTING SURVEYS
= NMUICPOTS UVYGRSE.

In either of these simple cases the frequency of occurrence of symbols is unaffected by the encryption operation. The cryptanalyst can get a good start toward breaking the code by a frequency analysis of cipher symbols [KULL76]. In secure systems complicated usage-dependent combinations of the two primitive encryption operations are used to cause all cipher symbols to occur with equal frequency.

It might seem that such simple systems would offer reasonable cryptosecurity since there are $26! \approx 4 \times 10^{26}$ substitutions possible on the 26 alphabetic characters in the first case and $n!$ permutations on n-symbol blocks in the second. But the redundancy of English (indeed, any natural language) is so great that the $\log_2(26!) \approx 88.4$ bits of equivocation introduced by the encryption algorithm can be resolved by a cryptanalyst, using frequency of occurrence counts on symbols, with approximately 25 symbols of cipher text! This illustrates how deceptive the appearance of large numbers of choices to the cryptanalyst can be in judging the cryptosecurity of a cryptosystem.

An obvious means of strengthening substitution ciphers is to use not one but several monoalphabetic substitutions, with the key specifying which substitution is to be used for each symbol of the cipher. Such systems are known as polyalphabetics. The

[2] Kahn [KAHN67, p. 764] has analogized substitution and transposition ciphers with continuous and batch manufacturing processes, respectively.

[3] This notation means: move the first symbol to the fifth place, the fifth symbol to the third place, the third symbol to the second place, and so on.

best known are the simple Vigenére ciphers wherein the substitutions are taken as the mod 26 sum of a symbol of the message m_i and a symbol of the key k_i, with the convention A = 0, . . . , Z = 25. Depending on the complexity of the substitution rule (key) chosen, the equivocation of such a Vigenére-type system can be made as great as desired, as we see later in examining the random key Vernam–Vigenére system. The following examples illustrate how the key complexity can affect the security of a cryptosystem.

In the simplest Vigenére-type systems, the key is a word or phrase repeated as many times as necessary to encrypt the message; for example, if the key is COVER and the message is THE MATHEMATICS OF SECRECY, the resulting cipher is

Message	THE MATHEMATICS OF SECRECY
Key	COV ERCOVERCOVE RC OVERCOV
Cipher	VVZ RQVVZRQVWXW FH GZGIGQT.

Kasiski's general solution of repeated key Vigenére ciphers starts from the fact that like pairings of message and key symbols produce the same cipher symbols; these repetitions are recognizable to the cryptanalyst [KAHN67]. The example above shows the group VVZRQ repeated twice; the length of the repeated group reveals that the key length is five. The cipher symbols would then be partitioned into five monoalphabets each of which is solved as a substitution cipher.

To avoid the problems of the preceding example, one can use a nonrepeating text for the key. The result is called a running-key Vigenére cipher. The running key prevents the periodicity exploited by the Kasiski solution. However, there are two basic types of solution available to the cryptanalyst in this case [KAHN66]. One can apply statistical analysis by assuming that both cipher text and key have the same frequency distributions of symbols. For example, E encrypted with E occurs with a frequency of ≈ 0.0169 and T by T occurs only half as often. A much longer segment of cipher test is required to decrypt a running-key Vigenére cipher; however, the methods, based on recurrence of like events, are similar.

The other technique for attacking running-key ciphers is the so-called *probable word* method in which the cryptanalyst "subtracts" from the cipher words that are considered likely to occur in the text until fragments of sensible key text are recovered; these are then expanded using either of the two techniques just discussed. The vital point is that although the equivocation in the running text can be made as large as desired, the redundancy in the language is so high that the number of bits of information communicated per bit of cipher exceeds the rate at which equivocation is introduced by the running key. Therefore, given sufficient cipher text, the cryptanalyst will eventually have enough information to solve the cipher.

The most important of all key variants to the Vigenére system was proposed in 1918 by the American engineer G. S. Vernam [VERN26]. Messages for transmission over the AT&T teletype system were at that time encoded in Baudot code, a binary code consisting of marks and spaces. Vernam recognized that if a random sequence of marks and spaces were added mod 2 to the message, then all of the frequency information, intersymbol correlation, and periodicity, on which earlier successful methods of attack against various Vigenére systems had been based, would be totally lost to the cryptanalyst. In this judgment Vernam's intuition was absolutely right, as would be proved two decades later by another AT&T scientist, Claude Shannon [SHAN49]. Vernam proposed to introduce uncertainty at the same rate at which it was removed by redundancy among symbols of the message. Unfortunately, this ideal requires exchanging impractical amounts of key in advance of communication, i.e., one symbol of key must be provided for every symbol of message. In Vernam's invention the keys were made up in the form of punched paper tapes which were read automatically as each symbol was typed at the keyboard of a teletypewriter and encrypted "on line" for transmission. An inverse operation at the receiving teletype decrypted the cipher using a copy of the tape. Vernam at first thought that a short random key could safely be used over and over; however, the resulting periodicity of the key permits a simple Kas-

iski-type solution. A second proposed solution was to compute a key of n_1n_2 bits in length by forming the logical sum, bit by bit, of two shorter key tapes of relatively prime lengths n_1 and n_2, so that the resulting key stream would not repeat until n_1n_2 bits of key had been generated. This form of Vernam system was used for a time by the U.S. Army.

The greatest contribution of the two-tape Vernam system came from its successful cryptanalysis, which led to the recognition of the unconditional cryptosecurity of one-time keys or pads. Major J. O. Mauborgne of the U.S. Army Signal Corps showed that cipher produced from key generated by the linear combination of two or more short tapes could be successfully analyzed by techniques essentially the same as those used against running-key systems. The unavoidable conclusion was that the Vernam-Vigenére system with either a repeating single key tape or with linear combinations of repeating short tapes to form a long key sequence were both insecure. The truly significant conclusion was arrived at by Friedman and Mauborgne: The key in an unconditionally secure stream cipher[4] must be *incoherent* (the uncertainty, or entropy, of each key symbol must be at least as great as the average information content per symbol of the message). Such a cryptosystem is referred to as a random one-time key or pad.[5] In other words, the system is unconditionally secure—not because of any failure on the cryptanalyst's part to find the right technique, but rather because the equivocation faced by the cryptanalyst leaves an irresolvable number of choices for key or plaintext message. While it is often stated that a Vernam-Vigenére cryptosystem with a nonrepeating random key is unconditionally secure, it is necessary to add the qualification that each symbol of the key introduce at least as much uncertainty as is removed by a symbol of the cipher.

An interesting example of the need for the key to introduce uncertainty, even with a nonrepeating random key, appears in a recent article by Deavours on the unicity point[6] of various encryption systems [DEAV77]. In Deavours's example, the key introduces exactly 1 bit per symbol using the random binary stream 001100110010000010111011 1 ··· to encipher a message in the Vigenére scheme with B as key if $k_i = 0$ and C as key if $k_i = 1$. Deavours's cipher is

TPOGD JRJFS UBSFC SQLGP COFUQ
NFDSF CLVIF TONWG T.

The first four letters, for example, could decrypt sensibly to either SOME or ROME, etc., but the reader should have no difficulty determining the intended message to be: SOME CIPHERS ARE BROKEN AND SOME BREAK THEMSELVES.

All of the preceding examples are of stream ciphers, illustrating the way in which the key equivocation appears in each case, and also the concepts of unicity point and one-time pad or key. We turn now to block ciphers, of which we will describe two. Block ciphers attempt to deny to the cryptanalyst the frequency statistics which have proved so useful against stream ciphers. One way to accomplish this is to operate on pairs of symbols (digraphs), triples (trigraphs), or, in general, on blocks (polygraphs). For manageability, manual block cryptosystems are limited to digraph substitutions. The best known manual digraph system is Wheatstone's Playfair cipher, in which a 25-symbol alphabet[7] is written in a 5 × 5 array with a simple geometric rule [GAIN56] specifying the cipher digraph to be substituted for each digraph in the message.

[4] This condition applies to both block and stream ciphers, although at the time the conditions were stated, block ciphers were not considered because of the difficulty of manual implementation.

[5] One needs to clearly distinguish between two kinds of undecipherability In one kind the equivocation is too high even if the analyst makes perfect use of all available information. This may be because of the brevity of cipher or of a lost key, as with the famous Thomas Jefferson Beale book ciphers, numbers 1 and 3 [HART64]. In the other, the code can be deciphered in principle but not in practice, as is probably the case with the MIT challenge cipher [GARD77].

[6] The unicity point was defined by Shannon to be the length of cipher beyond which only a single plaintext message could have produced the cipher, i.e , the point of zero equivocation to the cryptanalyst [SHAN49].

[7] The letter J is usually dropped in the Playfair cipher since it occurs infrequently and can almost always be filled in by context or by substituting I in the text

TABLE 1

Letter	*Number of Occurrences*	*Letter*	*Number of Occurrences*	*Letter*	*Number of Occurrences*
E	540	C	212	Y	57
T	479	M	177	B	44
O	384	D	168	U	42
A	355	H	145	K	33
N	354	U	136	Q	11
I	326	P	114	X	7
R	317	F	87	Z	4
S	308	G	67	J	1
L	219	W	65		

The cornerstone of modern mathematical cryptography was laid by Hill [HILL29, HILL31, ALBE41] in 1929. Hill recognized that nearly all the existing cryptosystems could be formulated in the single model of linear transformations on a message space. Hill identified a message n-tuple with an n-tuple of integers and equated the operations of encryption and decryption with a pair of inverse linear transformations. The simplest representation for such transformations is multiplication of an n-tuple (message) by a nonsingular $n \times n$ matrix to form the cipher and by the inverse matrix to decrypt and recover the message. For example, let the digits zero–nine be represented by the numbers 0–9, blank by 10, and the 26 letters of the alphabet by 11–36. The number of symbols, 37, is a prime; the encoding and decoding can be carried out with arithmetic modulo 37. If the encrypting matrix is

$$\mathscr{E} = \begin{pmatrix} 7 & 6 \\ 3 & 11 \end{pmatrix}$$

and the decrypting matrix is

$$\mathscr{E}^{-1} = \begin{pmatrix} 19 & 30 \\ 15 & 2 \end{pmatrix},$$

then the message LULL = (22, 31, 22, 22) would encrypt to the cipher

$$\begin{pmatrix} 7 & 6 \\ 3 & 11 \end{pmatrix}\begin{pmatrix} 22 & 31 \\ 22 & 22 \end{pmatrix} = \begin{pmatrix} 27 & 16 \\ 12 & 2 \end{pmatrix}$$

(all computations mod 37).

Similarly, the cipher (27, 16, 12, 2) decrypts to yield the message LULL by,

$$\begin{pmatrix} 19 & 30 \\ 15 & 2 \end{pmatrix}\begin{pmatrix} 27 & 16 \\ 12 & 2 \end{pmatrix} = \begin{pmatrix} 22 & 31 \\ 22 & 22 \end{pmatrix} \pmod{37}.$$

Note that the three L's in LULL encipher into different symbols. This illustrates the cryptographic advantage of polygraphic systems: The raw frequency-of-occurrence statistics for blocks up to size n are obscured in the encryption process; in the limit (with n), they are lost completely.

Table 1 shows the number of occurrences of each letter in 4652 letters of an English language computing science article. These patterns, which survive any monographic substitution, are invaluable clues to the cryptanalyst. For instance, he knows that T is one of the most frequently occurring letters and can be quite sure that T is one of the eight most frequently seen letters. Figure 1 shows the frequency-of-occurrence data for single symbols in the cipher, for a simple monographic encryption, and for polygraphic encryption distributions with matrix sizes 2×2, 3×3, and 4×4. A perfect encryption system would have a flat distribution for all n-tuples; i.e., all possible n-tuples would be equally likely.[8]

Tuckerman [TUCK70] in his analysis of Vigenére-Vernam cryptosystems has shown that Vigenére systems using nonrandom transformations are always subject to statistical attack. This is to be expected

[8] Hill's system using an nth-order transformation resists simple statistical methods of cryptanalysis based on the frequency of occurrence of t-tuples in the cipher for t less than n; however, if the cryptanalyst has two ciphers resulting from the encryption of a single message with two involutory transformations $\mathscr{D}_1$ and $\mathscr{D}_2$ in $\mathscr{A}^n$ so that for all messages $M \in \mathscr{A}^n$, $\mathscr{D}_1(\mathscr{D}_1(M)) = \mathscr{D}_2(\mathscr{D}_2(M) = M$, and if he knows $\mathscr{A}$, he can recover $\mathscr{D}_1$ and $\mathscr{D}_2$. It was not this cryptanalytic weakness, however, which prevented the adoption of Hill's cryptosystem, but rather the difficulty of carrying out the manual encryption/decryption operations he had defined

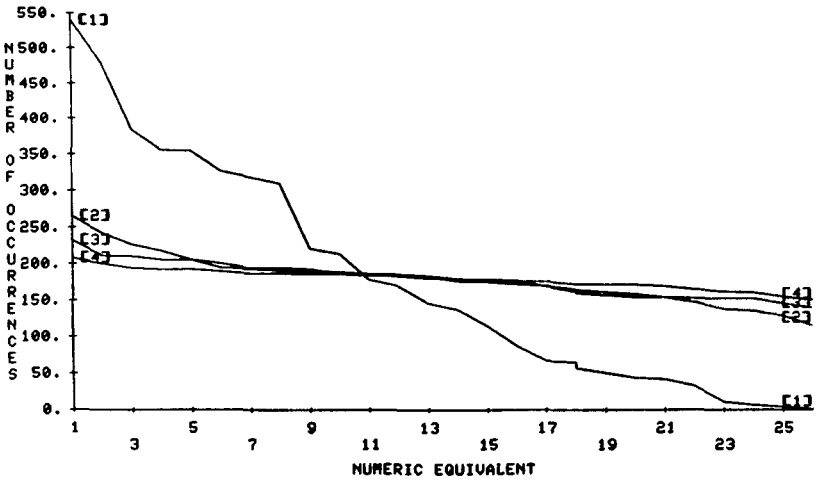

FIGURE 1 [1] Monographic substitution, [2] polygraphic substitution, matrix size 2 × 2, [3] polygraphic substitution, matrix size 3 × 3; [4] polygraphic substitution, matrix size 4 × 4

since the initial equivocation to the opponent must eventually be eroded by usage. Tuckerman provides the neat proof of this intuitive statement.

The reader wishing a more complete treatment is referred to GAIN56, KAHN67, or BRIG77 for further details of cryptanalysis. In a later section we take up current cryptotechnology, which has developed since World War II.

2. READER'S GUIDE

Because of the unavoidable length and detail of the subsequent sections, a brief outline of the development is given here. First, a parallel between the classical noisy communications channel and the general encryption/decryption channel is drawn. The reason for doing this is that error detecting and correcting codes and message or transmitter authentication are mathematically dual problems. In both cases redundancy, i.e., extra symbols, is introduced in the message, but the way in which this redundancy is used to communicate through the channel is different in the two applications. This is true whether the cryptosystem is symmetric or asymmetric.

Second, computationally infeasible problems are the source of cryptosecurity for both symmetric and asymmetric systems. One of the important points to this paper is to make clear how these computationally complex problems are embedded in an encryption/decryption process. To illustrate this, a frequently rediscovered encryption scheme dependent on maximal length linear feedback shift registers (LFSRs) is discussed to show how computational feasibility can destroy cryptosecurity. In the discussion of asymmetric encryption two examples of computationally infeasible problems are described in detail.

Linear feedback shift registers provide not only a simple illustration of the relationship between cryptosecurity and computational feasibility, but they also illustrate how redundancy is used in error detecting and correcting codes. The main text emphasizes these points, while a brief discussion of these devices is given in the appendix.

The ultimate objective of the paper is to impart to the reader a clear perception of how secrecy and authentication are accomplished in both symmetric and asymmetric encryption systems. This implies a clear understanding of which forms of secure communication can only be realized through asymmetric techniques, and which forms can be realized by either symmetric or asymmetric cryptosystems.

3. THE COMMUNICATIONS CHANNEL

A transmitter draws a message M from a space of possible messages $\mathcal{M}$ and sends it to a receiver over a noisy communications channel. It is possible that some $M' \neq M$

may be received. In 1948 Shannon [SHAN48] proposed the concept of the *entropy* of a message, which measures its information content. He showed how to introduce redundancy by means of a code; the extra symbols could be used to detect (and correct) errors in the received message M'. For example, Hamming codes add $2k + 1$ bits for each k errors to be detected [MACW77]. How this redundancy is introduced and utilized is a function of the way in which the errors occur in transmission, i.e., the statistics of the communications channel shown schematically in Figure 2. Essentially one wishes to impose a metric on the message space $\mathcal{M}$ so that the set of messages most apt to result from errors in the transmission of a given message M is also the one "closest" to M in $\mathcal{M}$. For example, if the errors in the binary symmetric channel are independent and uniformly distributed, the Hamming metric is a natural one to use; however, if adjacent symbol errors are more apt to occur, Berlekamp [BERL68] has shown the Lee metric[9] to be preferable. Coding theory is concerned with finding a partitioning of $\mathcal{M}$ into a collection of disjoint subsets (ideally "spheres") with all points in the ith set less than some specified distance from a central point C_i in the set. The code then consists of the labels (code words) of the collection of central points in the subsets of $\mathcal{M}$, with the maximum likelihood error correction rule being to decode any received point in $\mathcal{M}$ as the central point of the class that it belongs to in the partition.

Since we shall later wish to contrast the partitioning of $\mathcal{M}$ for message authentication to the kind of partitioning useful for error detection and correction—where the objective in both instances is to detect an incorrect message—we give in Table 2 an example of a Hamming code that adds three extra bits to each 4-bit block of message code [MASS69]. This code can be generated by taking as code words the 7-bit subsequences having the 4-bit messages in the low-order bit positions from the output of the linear feedback shift register (see appendix). If any single bit of the 7-bit code word is altered in transmission, the receiver can recover the message correctly by finding the code word that differs from the received block in the fewest number of bits.

TABLE 2

Message	*Code Word*
0000	000,0000
0001	011,0001
0010	110,0010
0011	101,0011
0100	111,0100
0101	100,0101
0110	001,0110
0111	010,0111
1000	101,1000
1001	110,1001
1010	011,1010
1011	000,1011
1100	010,1100
1101	001,1101
1110	100,1110
1111	111,1111

Figure 3 is a schematic diagram of the Shannon channel. The codes in $\mathscr{C}$ are so designed that the likelihood of an altered message being misinterpreted by the receiver is minimum. In the case of error correction, the code is designed to maximize the likelihood that the receiver will be able to transform the received message to the message actually sent correctly.

4. THE ENCRYPTION/DECRYPTION CHANNEL

The encryption channel also consists of a transmitter who wishes to send a message M to a receiver. But now the channel is assumed to be under surveillance by a hostile opponent. Cryptographic theory seeks to devise codes that cannot systematically be distinguished from purely random bit strings by the opponent. The statistical communications channel of the coding/decoding model has been replaced by a game-theoretic channel; nature has been replaced by an intelligent opponent. The opponent can have one or more of the following purposes:

a) To determine the message M.
b) To alter the message M to some other

[9] Whereas the Hamming metric is the number of symbol differences between two words, the Lee metric is the sum of the absolute differences of the symbols: for $W_1 = (0, 1, 2)$ and $W_2 = (2, 0, 1)$, $H(W_1, W_2) = 3$ and $L(W_1, W_2) = 4$. For binary code words the Hamming and Lee metrics are identical.

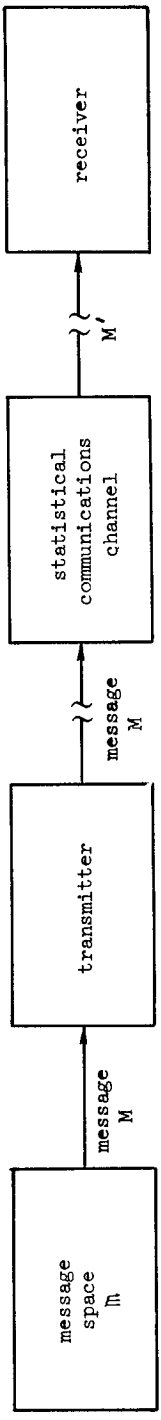

FIGURE 2. Classical communications channel.

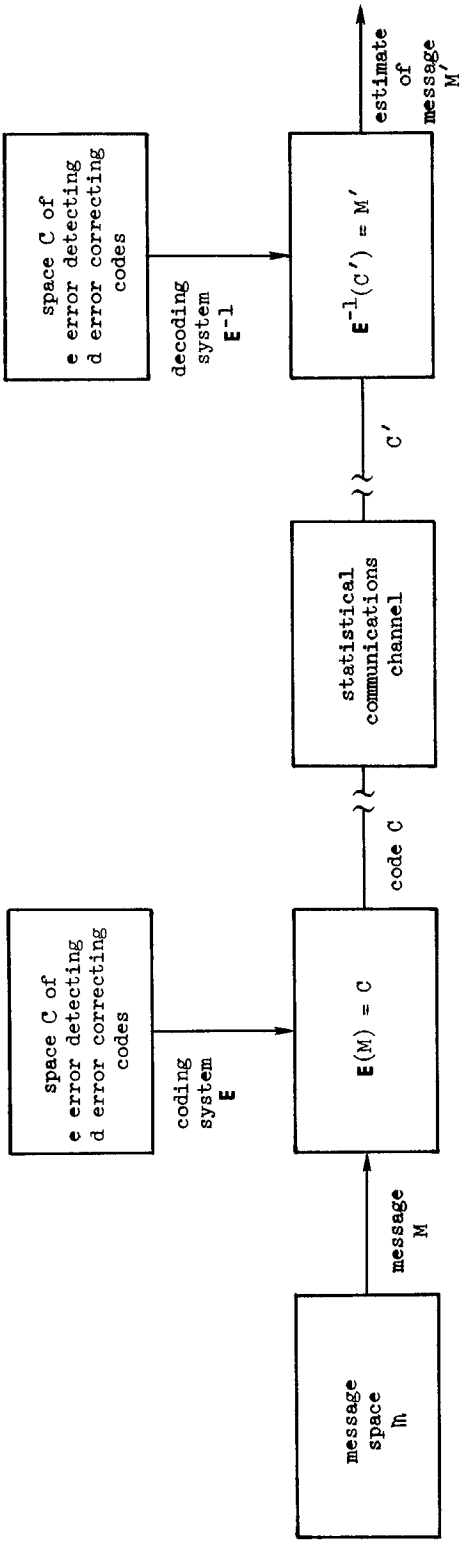

FIGURE 3 Functional schematic for the Shannon coding/decoding channel.

message M' and have M' accepted by the receiver as the message actually sent.

c) To impersonate the transmitter.

Thwarting a), i.e., ensuring secrecy, is the best known purpose of cryptographic systems, but modern data processing systems with controlled log-in and access to business files are greatly concerned with authenticating the "transmitter" (thwarting c)) and ensuring the integrity of the received messages (thwarting b)) [FEIS73, HOFF77, LIPT78, MART73]. In many cases the privacy or secrecy of communications is a secondary objective. An intelligent opponent could easily defeat the fixed strategies underlying error detecting codes by making improbable changes such that the received code words would be interpreted as incorrect messages. Moreover the opponent's task of "breaking" the code is not difficult because the code space is partitioned into spheres, which reduces the search. A perfectly secure code is one in which each cipher symbol is produced with equal probability by any message symbol when averaged over all possible keys. Deavours's example [DEAV77] was not secure because each cipher symbol could have been produced by only two message symbols rather than all 26 message symbols.

To be perfectly secure, an encryption system should randomly map the message space onto itself such that the opponent must consider all points in $\mathcal{M}$ to be equally likely candidates for the plaintext corresponding to the received ciphertext. Whereas a satisfactory "random" number generator need not be a good encryption function (as we shall see in an example a little later), a good encryption system is necessarily a good random number generator. In fact, Gait [GAIT77] has used the DES algorithm for random number generation with considerable success.

As Shannon pointed out [SHAN49], this implies that a perfect encryption scheme is equivalent to a latin square where rows correspond to messages, entries to keys, and columns to ciphers. However, a perfect cryptosystem may be unable to authenticate messages. Suppose that $\mathcal{M}$ is the space of *all* n-bit binary numbers, and that encryption consists in adding, modulo 2, a random n-bit binary number. In this case every proposed decipherment produces an acceptable message. When there is no redundancy in the messages, there is no basis on which to deduce the authenticity of a received cipher. An authentication system must introduce redundancy such that the space of ciphers is partitioned into the images (encryptions) of the messages in $\mathcal{M}$ and a class of unacceptable ciphers. If authentication is to be perfect, then the encryption scheme must consist of a family of partitions of the cipher space such that on learning any message-cipher pair, the opponent who does not know the key will be unable to do any better than pick a cipher at random from the cipher space. In other words, the objective is to diffuse the unacceptable ciphers throughout the entire cipher space. This is precisely the opposite of the error defeating code's objective, which is the *clustering* of the incorrect codes about an acceptable (correct) code.

Figure 4 is a schematic diagram of the abstract encryption/decryption channel. The parallel with the Shannon coding/decoding channel is apparent. Figure 4 is more general than the secrecy systems described by Shannon [SHAN49], Albert [ALBE41], or Feistel [FEIS73]; Shannon's and Albert's models were concerned only with secrecy, and Feistel's model dealt with a restricted form of message authentication. The model of Figure 4 encompasses all the objectives for secure communications. It should be noted that a cipher can be encoded to allow for the detection and correction of errors in transmission. This requires that the receiver first decode and correct errors before decrypting. In fact, such compound encryption/encoding is routinely used with satellite communications systems.

In encryption/decryption systems, the functions **E** and **D** (encryption and decryption) are assumed known to the opponent. If the system were to depend completely on **E** and **D**, the opponent would have sufficient information to defeat it. Therefore, something must be unknown if the opponent is to be unable to duplicate the actions performed by the authorized receiver. The unknown information is called the cryptographic *key*. The authorized receiver can use his secret deciphering key K' to decrypt the encrypted message.

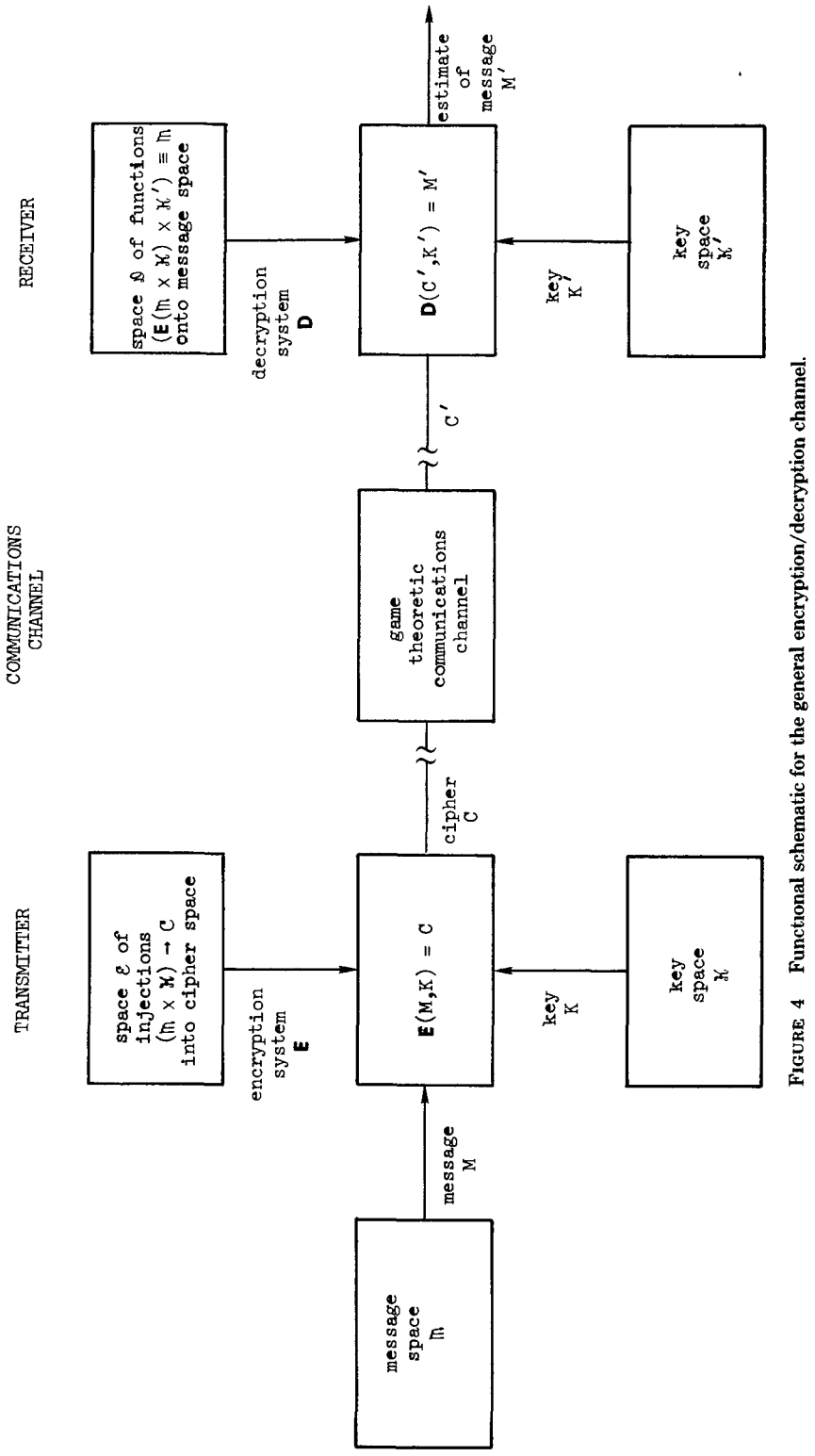

FIGURE 4 Functional schematic for the general encryption/decryption channel.

An encryption system can be described formally with the help of the message space $\mathcal{M}$, the key spaces $\mathcal{K}$ and $\mathcal{K}'$, the cipher space $\mathcal{C}$, a space $\mathcal{E}$ of mappings from $\mathcal{M} \times \mathcal{K}$ into $\mathcal{C}$, and a related space $\mathcal{D}$ of inverse mappings. For a particular mapping $\mathbf{E}$ from $\mathcal{E}$, M from $\mathcal{M}$, and K from $\mathcal{K}$, $\mathbf{E}(M, K) = C$ is the encipherment of message M by key K. There must be a deciphering function $\mathbf{D_E}$ corresponding to $\mathbf{E}$ and a key K' corresponding to K such that messages can be uniquely recovered:

$$M = \mathbf{D_E}(\mathbf{E}(M, K), K') = \mathbf{D_E}(C, K') \quad \text{for all } M. \quad (1)$$

By itself (1) does not describe a secure encryption system. For example, if $\mathcal{M} \equiv \mathcal{C}$ and $\mathbf{E}$ is the identity function, then (1) is trivially satisfied with $C = M$ for all M; obviously there is no cryptosecurity for any choice of K. Shannon [SHAN49] defines a secrecy system $\mathbf{E}$ to be perfect (unconditionally secure) if an opponent knowing $\mathbf{E}$ and arbitrarily much cipher C is still left with a choice from among all possible messages M from $\mathcal{M}$. For this to be true, there must be as many keys as there are messages. Moreover the uncertainty about the key K must be essential: The opponent's uncertainty about messages must be at least as great as his uncertainty about the key. In Shannon's model $\mathcal{K} \equiv \mathcal{K}'$ and $\mathcal{E} \equiv \mathcal{D}$, and only objective a), secrecy, is considered. Under these constraints, $\mathbf{E}$ is a mapping from the message space $\mathcal{M}$ into the cipher space $\mathcal{C}$, and $\mathbf{D}$ is $\mathbf{E}^{-1}$, the inverse function to $\mathbf{E}$; the key K then acts as an index for a pair $(\mathbf{E}, \mathbf{D})$. Perfect security is achieved by having one key for each possible $(\mathbf{E}, \mathbf{D})$ pair. Contemporary cryptosystems seldom realize this level of unconditional security. In fact, most of current cryptology deals with systems which are secure in the sense that exploiting the available information is computationally infeasible; but these systems are not unconditionally secure in Shannon's sense. The important exceptions include the Washington–Moscow hot line and various high-level command circuits. In the remainder of this paper, we are concerned with computationally secure systems, but not unconditionally secure ones.

5. COMPUTATIONAL COMPLEXITY AND SYMMETRIC ENCRYPTION

A fundamental change in the practice of cryptography began in the early 1950s. We have already pointed out that a perfectly secure cryptosystem requires impractical quantities of key for most applications. Almost all of cryptography has been devoted to finding ways of "diffusing" smaller, manageable amounts of uncertainty in order to approximate longer keys, that is, keys which appear to have come from a key space with greater uncertainty. This is usually done with an easily computed function of an input sequence, the true key, which produces as output a much longer sequence, the pseudokey. The pseudokey is used as K in Figure 4.

If such a procedure is to be cryptosecure, it must be infeasible to invert the function to recover the true key from the pseudokey; that is, it must be intractable to compute the future output of the function even though the function itself is known and lengthy observations of the output are available. From World War II until the early 1950s these objectives were met on an ad hoc basis through the intuitive judgment of cryptosystem designers. However, electronic computing and the theory of computational complexity transformed the idea of "diffusing" a limited amount of uncertainty into an analytical design question.

In Figure 4 the key spaces $\mathcal{K}$ and $\mathcal{K}'$ represent the equivocation to the opponent of the system at any given stage in its operation. For example, in an English alphabet one-time pad of n equally likely symbols, $|\mathcal{K}| = 26^n$; each point in $\mathcal{K}$ represents about $\log_2(26)^n \approx 4.7n$ bits of information, and so a 1000-symbol one-time "key" would be represented as a point in a binary space of 2^{4700} possible sequences. Because keys are as voluminous as the messages they secure, one-time keys are impractical for large-volume communications. In the early 1950s cryptologists recognized that if a (true) key K from a smaller dimensional key space $\mathcal{K}$ was used to generate a much longer (pseudo) key $\hat{K}$ using an algorithm whose inversion was sufficiently complex computationally, then the cryptanalyst would be unable to compute either K or $\hat{K}$.

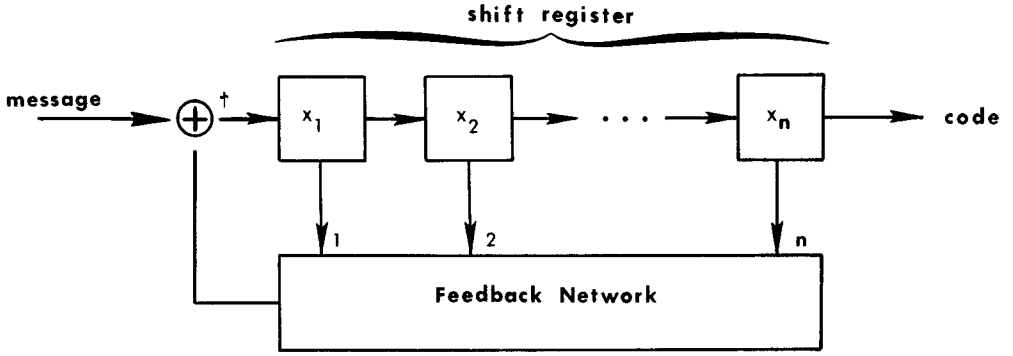

FIGURE 5 † Exclusive OR.

Modern cryptology rests largely on the implementation of this principle.

In terms of Figure 4, the "diffusing" of uncertainty is defined by this condition: For nearly all encryption/decryption pairs (**E**, **D**) and keys K and K', it is computationally infeasible to compute K (or K') from a knowledge of **E**, **D**, C, and M. A system in which either $K = K'$ or one of K and K' is easily computed from knowledge of the other is called a *symmetric* system.

All the examples in the introduction are of symmetric systems. For a one-time key, the two communicants must each have a copy of the same key; $K = K'$ in this case. Similarly, the simple Vigenére and Vernam-Vigenére systems both have $K \equiv K'$. On the other hand, in the Hill linear transformation system, described in Section 1, the receiver must have $\mathbf{E}^{-1}$, not **E**, although it is easy to compute $\mathbf{E}^{-1}$ from a knowledge of **E**.

Maximal length linear feedback shift registers (LFSRs), which are used for error detecting and correcting codes, illustrate that one must take great care in choosing key functions. Some apparently complex functions are not so. Because the $(2^n - 1)$-bit sequence from a maximal length LFSR satisfies many tests for randomness, e.g., the runs property [GOLO67] and lack of intersymbol correlation up to the register length n, numerous suggestions have been made to use these sequences either as key in a Vernam-Vigenére stream cipher mode, as shown in Figure 5, or as block encryption devices on n-bit blocks of message bits [BRIG76, GEFF73, GOLO67, MEYE72]. The feedback network, i.e., the coefficients of the feedback polynomial, and the starting state of the register serve as the key.

Assuming that the cryptanalyst can by some means, such as probable word analysis, recover bits of the cipher (which need not be consecutive), he can set up and solve a system of at most $2n$ linear equations with which to duplicate the future output of the original sequence generator. Berlekamp [BERL68] and Massey [MASS69] have found efficient algorithms for doing this in at most $2n$ steps. Thus the problem of finding K is only of linear complexity (in n); hence K is not well concealed despite the apparently large number of possible feedback functions. A more complete description of LFSRs is given in the appendix.

Another proposed mode of crypto use for LFSRs is for block ciphers: The register is loaded with an n-bit block of plaintext, it is stepped for $k > n$ steps, and the resulting register state is taken as the cipher. Figure 6 shows an example of the state diagram for such an LFSR. Using $k = 7$, for example, the message 00001 encrypts to 11010. To decrypt, one uses the "inverse feedback function," which reverses the stepping order of the state diagram of Figure 6, when a 00001 would be the register state resulting from stepping the register seven steps from the starting point (cipher) of 11010. In this example K (forward stepping) and K' (reverse stepping) are easily computable from each other. Although the output is sufficiently random to be useful as a pseudo-random bit sequence generator, the inversion to find K' or K is only of linear computational complexity.

The National Bureau of Standards Data

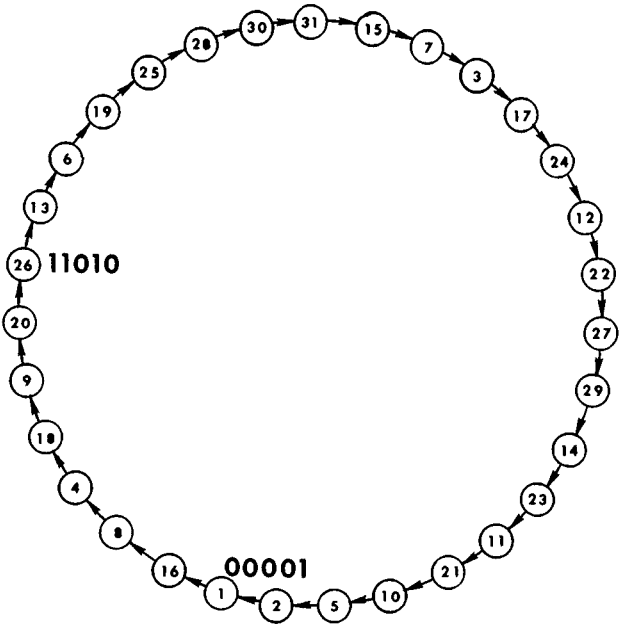

FIGURE 6

Encryption Standard (DES) provides a widely recognized example of a symmetric encryption/decryption whose keys are well concealed by computational complexity. Roberts [ROBE75] states that

> The algorithm is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 64-bit key.[10] Deciphering must be accomplished by using the same key as for enciphering, but with the schedule of addressing the key bits altered so that the deciphering process is the reverse of the enciphering process. A block to be enciphered is subjected to an initial permutation IP, then to a complex key-dependent computation and finally to a permutation which is the inverse of the initial permutation IP^{-1}.

This shows clearly that the system is symmetric. It indicates that the "complex key-dependent computation" conceals the key. The encryption function used in the DES is known as a product cipher [MORR77]; it comprises 16 successive repetitions of a nonlinear substitution (to provide "confusion") alternating with permutations (to provide "diffusion"). There is considerable controversy[11] about the cryptosecurity of the DES [DIFF77, MORR77] centering on the possible brute force attack of a system by enumerating all the keys for the present 56-bit key; yet no one has proposed an inversion of the encryption function itself, which thus far appears to be as computationally complex as its designers believed it to be.

6. COMPUTATIONAL COMPLEXITY AND ASYMMETRIC ENCRYPTION

In symmetric cryptosystems, the keys at the transmitter and receiver, K and K', respectively, either are the same or can be easily computed from each other. We now consider cryptosystems in which this is not the case. There are three possibilities.

a) *Forward asymmetric*: The receiver's

[10] Actually only 56 bits rather than the stated 64, since 8 bits are used for a parity check

[11] The controversy is centered on Hellman's accusation that the National Security Agency has deliberately chosen the DES key to be of a size that it can break. The pros [HELL79a, DAVI79] and cons [TUCH79, BRAN79] of this argument are summarized in the recent editorial debate in the *IEEE Spectrum* [SUGA79]

key (K') cannot easily be computed given the transmitter's key (K).

b) *Backward asymmetric*: The transmitter's key (K) cannot easily be computed given the receiver's key (K').

c) *Bidirectional asymmetric*: Neither K nor K' can be computed given the other.

As usual, the enemy is assumed to know **E**, **D**, M, and C. The term "asymmetric system" refers to all three cases.

The primary applications of (bidirectional) asymmetric encryption systems derive from these two properties:

1) Secure (i.e., secret) communication is possible even if the transmitter's key is compromised.
2) Authentication of the transmitter (message) is possible even if the receiver's key is compromised.

Note that 1) applies to the forward asymmetric encryption system and 2) to the backward encryption system.

Whereas symmetric cryptosystems have been in use for many years, asymmetric encryption systems are a recent development in cryptography. In 1976 Diffie and Hellman [DIFF76] published a conceptual scheme for this kind of cryptosystem, which they called a *public-key cryptosystem* because no pair of potential communicants had to exchange a key secretly in advance. It is essential, however, that the key exchange be secure, so that the communicants can be confident of the keys' owners—otherwise authentication is not possible. Merkle [MERK78a] contemporaneously discovered a related principle that allows the communicants to exchange a key with work $O(n)$, while requiring the opponent to face work $O(n^2)$ to determine the key from monitoring the communicants' exchange. Merkle discovered a forward asymmetric encryption system.

In terms of Figure 4, these conditions must be satisfied by an asymmetric encryption scheme:

1) The keys are concealed by a computationally complex problem from the plaintext and cipher.

2) It is easy to compute matched pairs of keys (K, K') such that

$$\mathbf{D}_E(\mathbf{E}(M, K), K') = M.$$

3) The encryption and decryption functions, **E** and **D** are implemented by fast algorithms.

4) At least one of the keys (K and K') is concealed from a knowledge of the other key by a computationally complex problem.

5) For almost all messages it must be infeasible to find cipher/key pairs that yield that message. That is, the opponent is forced to find the "true" (M, K) that encrypted to the cipher C at hand.

These conditions differ slightly from those imposed on public-key cryptosystems [DIFF76]. Condition 1) is the basic requirement for a practical privacy system; we state it explicitly to exhibit one of the two places in the abstract encryption channel where computational complexity is essential. The public-key cryptosystem was formulated as a two-way communications channel by its inventors, so that the keys are interchangeable: $\mathbf{E}(\mathbf{D}_E(M, K'), K) = M = \mathbf{D}(\mathbf{E}(M, K), K')$[ADLE78, HELL78]. Condition 5) enables detecting deception: The opponent cannot easily find alternate keys giving the same ciphertext [GILB74].

As of 1979, no one had exhibited functions that provably satisfied these conditions. The working approach toward constructing such functions has been to take some problem, known or believed to be exceedingly complex, and make the "obvious" method of finding the keys equivalent to solving the hard problem. Examples of hard problems are factoring a product of very large prime factors, the general knapsack problem, and finding the logarithm of an element in a large field with respect to a primitive element. What is hoped for in such a scheme is that the converse is also true; i.e., decryption is equivalent to solving the hard problem. The first results toward this crucial step in "proving" the cryptosecurity of any asymmetric system were obtained by Rabin [RABI79] and Williams [WILL79b]; they showed that the factorization problem for large moduli is equivalent to decryption for almost all ciphers in Rabin's encryption scheme. We will return to this point later.

6.1 The Knapsack Trapdoor

One of the best known proposals for a forward asymmetric system was made by Merkle and Hellman [MERK78b], who suggested basing asymmetric encryption on the knapsack (or subset sum) problem. The *knapsack problem* is to determine whether a weight S can be realized as the sum of some subset of a given collection of n weights w_i—i.e., to determine whether there exists a binary vector $\mathbf{s}$ for which $S = \mathbf{s} \cdot \mathbf{w}$.[12] Without restrictions on $\mathbf{w}$, solutions need not exist or there may be several. For example, $S = 515$ has three solutions, while $S = 516$ has no solution in the 10-weight knapsack appearing in Hellman's paper [HELL78].[13] The time to verify whether a given vector $\mathbf{s}$ is a solution is $O(n)$. In contrast, the time needed to find a solution vector $\mathbf{s}$ is believed to be of exponential complexity. Horowitz and Sahni [HORO74] have published a search algorithm for the knapsack problem requiring $O(2^{n/2})$ time and $O(2^{n/2})$ memory; and more recently Schroeppel and Shamir [SCHR79] have devised an algorithm of the same time complexity but requiring only $O(2^{n/4})$ memory. The knapsack problem is an NP-complete problem [KARP72].

It is important to remember that the computational complexity of NP-complete problems is measured by the difficulty of solving the worst cases, whereas cryptosecurity is measured by the expected difficulty over all members of the class. Suppose, for example, that the knapsack vector $\mathbf{w}$ is chosen with the w_i in strict dominance, i.e., $w_i > \sum_{j=1}^{i-1} w_j$. In this case $\mathbf{s}$ can either be found or shown not to exist in at most n subtractions: $s_i = 1$ if and only if $S - S_{i-1} \geq w_i$, where S_{i-1} is the partial sum of the first $i - 1$ components of the dot product. Another example is $w_i = 2^{i-1}$, in which case the problem reduces to finding the binary representation of $0 \leq S \leq 2^n - 1$. Both these examples illustrate how simple a knapsack problem can be for special $\mathbf{w}$. An encryption system based on such a simple $\mathbf{w}$ would not be secure.

Merkle and Hellman defined two special classes of vectors $\mathbf{w}$, which they call *trapdoor knapsacks*; with a trapdoor knapsack the designer can easily compute the subset vector $\mathbf{s}$, while the opponent is faced with solving a hard ($O(2^{n/2})$?) problem. The simplest scheme is an "additive trapdoor knapsack," in which the designer starts with any strictly dominating weight vector $\mathbf{w}$ containing n weights, as described above, and derives a related weight vector $\mathbf{v}$, which is believed to be a hard knapsack. This is done by choosing a modulus n and a multiplier e which is relatively prime with respect to n, and then computing the n weights v_i of $\mathbf{v}$ by the rule $ew_i \equiv v_i \pmod{m}$. Since e is relatively prime with respect to m, there exists a d, easily computed using the Euclidean algorithm, such that $ed \equiv 1 \pmod{n}$. The numbers d and m are the receiving key K', and the "hard" knapsack weight vector $\mathbf{v}$ is the transmitting key K. A binary message is broken into n-bit blocks. Each n-bit block becomes a vector $\mathbf{s}$ for the knapsack problem: the transmitter computes the cipher $S' = \mathbf{s} \cdot \mathbf{v}$. Since the cryptanalyst only knows S' and $\mathbf{v}$, he is forced to solve the knapsack problem for $\mathbf{v}$. The authorized receiver, however, computes $dS' \equiv S \pmod{m}$; he then solves the simple knapsack $(S, \mathbf{w})$ in $O(\mathrm{n})$ time because $\mathbf{w}$ is of the dominating form. If m is chosen to strictly dominate the sum of all the weights, then the computations may be done in integer arithmetic as well as in the modular arithmetic.

To further illustrate this simple trapdoor knapsack, use the easy knapsack weight vector $\mathbf{w} = (1, 2, 4, 8)$; choose $m = 17 > 1 + 2 + 4 + 8 = 15$ and $e = 5$. Then $d = 7$ and $\mathbf{v} = (5, 10, 3, 6)$. In this system the subset vector $\mathbf{s} = (0, 1, 0, 1)$ would be transmitted as $S' = \mathbf{s} \cdot \mathbf{v} = 16$. The receiver finds $S = 7 \cdot 16 = 10 \pmod{17}$; since he also knows $\mathbf{w}$, the authorized receiver can solve for $\mathbf{s}$ in three subtractions. The same principles apply to realistic implementations, which use $n = 100$ or larger.

Note that it has not yet been *proved* that the modular derivation of $\mathbf{v}$ from the easy knapsack $\mathbf{w}$ results in a hard knapsack.

---

[12] If $\mathbf{s} = (s_1, \ldots, s_n)$ and $\mathbf{w} = (w_1, \ldots, w_n)$, then the dot product $\mathbf{s} \cdot \mathbf{w} = \sum_{i=1}^{n} s_i w_i$. The vector $\mathbf{s}$, where $s_i = 0$ or 1 such that $S = \mathbf{s} \cdot \mathbf{w}$, selects some of the "objects" to fill a "knapsack" of capacity S

[13] $\mathbf{w} = (14, 28, 56, 82, 90, 132, 197, 284, 341, 455)$, and $\mathbf{s} = (1001111000)$, (0110100010), or (1100010010) for $S = 515$

Shamir and Zippel [SHAM78] have shown that if the opponent knows m as well as $\mathbf{v}$, he can employ a simple algorithm whose output is $\mathbf{w}$ with high probability.

6.2 The Factorization Trapdoor

Another asymmetric system is the public-key encryption scheme proposed by Rivest, Shamir, and Adleman [RIVE78]. The trapdoor in the scheme is based on the difference in computational difficulty in finding large primes as opposed to factoring large numbers. The best algorithms known at the present can find a d-digit prime number in time $O(d^3)$, while the complexity of factoring a large number n exceeds any polynomial bound, currently $O(n^{(\ln(\ln n)/\ln n)^{1/2}})$. In the proposed system, one chooses a pair of primes p and q so large that factoring $n = pq$ is beyond all projected computational capabilities. One also chooses a pair of numbers e and d, where $(e, \varphi(n)) = 1$,[14] and $ed \equiv 1 \bmod \varphi(n)$; $\varphi(n) = (p-1)(q-1)$. In other words, e and d are multiplicative inverses in the group of residue classes modulo $\varphi(n)$. When used as a public-key cryptosystem, e and n are published in the public-key directory and d is kept secret. Because the receiver (designer) knows p and q, the system is forward asymmetric.

A variant of this scheme illustrates a bidirectional asymmetric encryption system. Assume that a higher level of command designs the system, e.g., chooses p, q, and e, computes d, and then gives (e, n) and (d, n) to two subordinate commands that require an asymmetric encryption channel between them. Since computing the multiplicative inverse d of e from a knowledge of e and n is essentially the same as factoring n or determining $\varphi(n)$, d is secure from an opponent knowing only n and e. Conversely, computing e from a knowledge of d and n is of the same difficulty. The two keys (e, n) and (d, n) are separated by a computationally difficult problem. Obviously, the "higher level of command" can be replaced by a volatile memory computing device so that no single party is in possession of the information which could compromise the system.

A message $M \in \mathcal{M}$ is encrypted in this system to the cipher C by the transmitter using key $K = (e, n)$ by the rule

$$M^e \equiv C \pmod{n},$$

and C is decrypted by the authorized receiver using $K = (d, n)$ by the rule

$$C^d \equiv M \pmod{n}.$$

For example, if $p = 421$ and $q = 577$ so that $n = pq = 242{,}917$ and $\varphi(n) = 241{,}920$, then for $e = 101$, $d = 9581$. Using these values $K = (101{:}242{,}917)$ and $K' = (9581{:}242{,}917)$ so that the message $M = 153{,}190$ encrypts by

$$C = 153{,}190^{101} \equiv 203{,}272 \pmod{242{,}917},$$

and C decrypts by

$$M = 203{,}272^{9581} \equiv 153{,}190 \pmod{242{,}917}.$$

Much effort has been devoted to the investigation of whether the scheme just described is secure and whether decryption (for almost all ciphers) is as hard as the factorization of n. Several authors [HERL78, SIMM77, WILL79a] have investigated the restrictions on the primes p and q that must be imposed to ensure cryptosecurity; they conclude that it is not difficult to choose the primes so that the known cryptoweaknesses are avoided [WILL79a]. It is probable that these same steps are also sufficient to ensure that decryption of almost all ciphers is as hard as the factorization of n. However, this crucial result has not been proved. Instead, Rabin [RABI79] has shown that if instead of the encryption function $C = M^e$ one uses

$$C \equiv M(M + b) \pmod{n}, \qquad b \geq 0,$$

which is effectively the same as $e = 2$ where $n = pq$, as in the Rivest et al. scheme, then decryption to an unauthorized user is not simply a consequence of being able to factor n but is actually equivalent. Unfortunately, even the authorized user is left with an ambiguity among four potential messages in this scheme. Williams has completed this work by proving that for suitably chosen primes p and q the ambiguity is removed and that decryption of almost all messages is equivalent to factoring n [WILL79b].

[14] $\varphi(n)$ is the Euler totient; it is simply the number of integers less than n and relatively prime with respect to n. $(e, \varphi(n)) = 1$ is a notation indicating that e and $\varphi(n)$ are relatively prime.

(Ron Rivest has pointed out that this statement is precisely true for ciphertext-only attack and that it does not hold for chosen-plaintext attack [BRIG77].)

For example, using the same primes and message as above in the simple Rabin scheme, $p = 421$, $q = 577$, and $M = 153{,}190$, and letting $b = 0$, one obtains the cipher

$$C = 153{,}190^2 \equiv 179{,}315 \pmod{242{,}917}.$$

Four messages from $\mathcal{M}$ have C as their square mod n: M, of course, and $-M = 089{,}727$, as well as $M' = 022{,}788$ and $-M' = 220{,}129$.

The important point is that these results are persuasive evidence of equivalence between decryption for almost all messages and the factorization of n in these schemes.

A common misconception is that asymmetric encryption/decryption (public-key encryption) is more secure than its (symmetric) predecessors. For example, Gardner [GARD77] suggests that public-key cryptosystems are more cryptosecure than existing systems, and a lengthy editorial in the *Washington Post*, July 9, 1978, was entitled "The New Unbreakable Codes—Will They Put NSA Out of Business?" [SHAP78]. The discussion in the two previous sections on symmetric and asymmetric encryption demonstrates clearly that asymmetric cryptosecurity depends on precisely the same mathematical condition as most high-quality symmetric cryptosystems—computational work factor. Basing cryptosystems on NP-hard problems opens new worlds of codes which may be as secure as traditional codes. But the new systems are not necessarily more or less secure than existing cryptosystems.

7. AUTHENTICATION

The asymmetric encryption channel serves two functions:

1) Secret communication is possible even if the transmitter's key (K) is public.
2) Authentication of messages is possible by anyone who knows the receiver's key (K'), assuming that K and K' are not easily computed from each other.

The separation of secrecy and authentication in asymmetric systems has a natural counterpart in the different security concerns of the transmitter and receiver: The transmitter wishes assurances that the message cannot be disclosed or altered, whereas the receiver is primarily concerned that the message could only have come from the transmitter.

The different security concerns of transmitter and receiver are well illustrated by the concerns of the various parties involved in a transaction by check. The person writing the check (the transmitter) is not concerned with its authenticity, but he is concerned that no one will be able to alter the amount shown on his signed draft. The person accepting the check (the receiver) is primarily concerned with the authenticity of the check. An intermediate party accepting the check as a second-party draft is concerned with both of these aspects: that the check is unaltered and authentic. The ultimate receiver, the bank, keeps signature cards on file to help verify (if needed) the identity of the person who wrote the check, but its concerns are the same as those of the other intermediate receivers.

Authentication is closely related to error detecting codes. The message $\mathcal{M}$ is partitioned into two classes, acceptable and unacceptable messages, similar to the classes comprising the most probably correct and incorrect messages in the previous case. To realize authentication despite an intelligent opponent, it is essential to conceal these classes in the ciphers. Using an unconditionally secure cryptosystem to encrypt the messages from $\mathcal{M}$ into ciphers from $\mathcal{C}$, every cipher $C \in \mathcal{C}$ would with equiprobability over $\mathcal{K}$ be the encryption of any message in $\mathcal{M}$. But in this ideal case, if the opponent substituted another cipher C' for the correct cipher C, the probability that it would decrypt to a message in the class of acceptable messages would be simply $|\mathcal{A}|/|\mathcal{M}|$, where $\mathcal{A}$ is the class of acceptable messages. For example, if $\mathcal{M}$ is the set of $26^4 = 456{,}976$ four-letter alphabetic sequences and $\mathcal{A}$ is the set of four-letter English words in *Webster's Unabridged International Dictionary*, then the probability that a randomly chosen four-letter cipher will decrypt to an English word is very close to 1/7. In other words, the equivocation to the opponent of this "natural" authentication system is ≈ 2.81 bits.

The point is that authentication is *only* achievable by introducing redundancy into the message—exactly as is done to achieve an error detecting or correcting capability. Simply having the required level of redundancy is not sufficient. The redundancy must be diffused throughout the cipher, lest the signature information be separated from the proper message and appended to another message.

The bidirectional public-key encryption system proposed by Rivest, Shamir, and Adleman can be used by two subscribers, A and B, as a means of authenticating (signing) messages. Assume that A wishes to send a message M to B; B must later be able to prove to a third party (observer or judge) that M originated with A. For example, A is ordering B (his broker) to make a large stock sale which B fears A may disavow if the market value of the stock should increase. A has entered his public-key (e_A, n_A) into the public directory. Similarly B has entered (e_B, n_B). A computes

$$M^{d_A} \equiv C_A \pmod{n_A}$$

using his secret key (d_A, n_A) and then computes

$$C_A{}^{e_B} \equiv C \pmod{n_B}$$

using B's public key. This cipher can only be decrypted by B; A is therefore assured of the secrecy of his message. On receiving C, B computes

$$C^{d_B} \equiv C_A \pmod{n_B}$$

using his secret key and saves C_A as his "signed" version of the message. He then computes

$$C_A{}^{e_A} \equiv M \pmod{n_A}$$

using A's public key. Since this later step can be duplicated by any observer given C_A by using A's public information, the claim is that M could only have come from A.[15]

It has been argued that since M, C_A, and C are all the same length, say k bits, there is no apparent redundancy, as is required for authentication. But this is not true: Suppose that M were perfectly encoded, i.e., a random (equiprobable) k-bit binary number. Now the observer has no way of rejecting any k-bit number as not having been originated by A. A must therefore include in M identifiers, such as his name or ID number, time of day, or transaction number, which serve only to distinguish acceptable from unacceptable messages. The security of the authenticator is still measured by the degree of signature redundancy introduced.

Authentication is possible using either symmetric or asymmetric channels. We noted earlier that with DES, a symmetric block ciphering system, messages can be authenticated using Feistel's block chaining [FEIS73] technique. In this approach successive blocks of 56 bits of the text are used as keys to successively encrypt the ciphers from the preceding step, with one 56-bit initial key unknown to the opponent. The resulting cipher is a "function" of every bit in the message and is resistant to inversion even against a known plaintext attack. The appended authenticator must match an "acceptable" message, usually in a natural language to be accepted.

The unique feature of asymmetric encryption systems for authentication is that a receiver can decrypt but not encrypt; one terminal of the communications link can be intentionally exposed without compromising the other terminal. This is not possible in a symmetric system.

8. SECURE COMMUNICATIONS

Despite the different concerns of the transmitter, the receiver, or the intermediary in authentication, the objective is always an authentication system whose cryptosecurity is equivalent to the security of the transmitter's encryption key. This means that the transmitter can purposely introduce redundancy in such forms as message identifiers prior to encryption, or else he can depend on redundancy inherent in the message format or language to allow the authorized receiver to reject bogus messages.

[15] There is a significant difference between digital signatures and a signature to a document. Once the signer affixes his signature to a document, there is nothing he can do that will interfere with the future verification of the authenticity of the signature. In the digital signature scheme described above, however, A can deliberately expose his secret key d_A and thereby make the authenticity of all digital signatures attributed to him questionable

The cryptosystem may be either symmetric if all communications terminals are secure, or asymmetric if one of the communications terminals is at a physically unsecured site.

There are four possible combinations of security concerns. They are listed in Table 3. Each corresponds to a class of real communications systems.

TABLE 3

Class	*Message/Transmitter Authentication*	*Secrecy*
I	No	No
II	No	Yes
III	Yes	No
IV	Yes	Yes

Class I corresponds to normal, nonsecure communications. We call this the *public channel.*

Class II is the classical case of secret or private communications. We call this the *private channel.* This channel is realizable with symmetric or asymmetric techniques. In the symmetric case a compromise of the key at either end of the communications channel precludes all further secret communications. In a forward asymmetric system secret communications are still possible even if the transmitter's key is public.

The necessity for communicants' using symmetric systems to provide a secure way to exchange keys in advance is a severe restriction. A commercial cryptonet, for example, could have many thousands of subscribers, any pair of whom might wish to communicate. Clearly the number of keys to support symmetric encryption would be unmanageable. In a forward asymmetric encryption system, however, a subscriber S_i could publish his encryption pair $\mathbf{E}_i$ and K_i in a public directory. Anyone wishing to communicate a secret message M to S_i in secrecy transmits $\mathbf{E}_i(M, K_i)$, which can only be deciphered by S_i. It is this application that led to the name "public-key cryptosystem." It is essential, however, that the transmitter be certain that $\mathbf{E}_i$ and K_i are the key entries for S_i. In other words, while a secret exchange of keys is no longer (in an asymmetric system as opposed to a symmetric one) needed, an authenticated exchange of keys is still required! This is an important point since it is frequently said—incorrectly—that there is no key distribution problem for public-key systems.

Class III is an unusual communications system that could not exist in a symmetric cryptosystem. In a system of this type, message and transmitter authentication is required, but secrecy cannot be tolerated. We call this a *signature channel.* An application of this channel for treaty verification has been developed at Sandia Laboratories [SIMM79].

Assume that the United States and the Soviet Union sign a comprehensive test ban treaty in which each party agrees to stop all underground testing of nuclear weapons. Each side wishes to verify that the other is complying, that is, is not surreptitiously carrying out underground tests. One of the most reliable techniques for detecting underground tests uses medium-distance seismic observatories that measure the ground motions resulting from an underground detonation. These techniques are highly reliable; either nation could have confidence in the output message from seismic instruments suitably located in the host (other) nation's territory. It is not difficult to secure the instruments physically in subsurface emplacements; only the data stream sent through an open communications channel is subject to attack. If the host nation could successfully substitute innocuous seismic records for the incriminating records of underground tests, it could cheat undetected. This problem is solvable using either symmetric or asymmetric encryption techniques. The receiver (nation to which the seismic installation belongs) need only encrypt the seismic data along with as many identifiers—station ID number, date, or clocks—as might be needed for authentication. This method of authentication is as secure as the encryption system used to produce the cipher. However this solution would almost certainly be unacceptable to the host nation (in whose territory the seismic observatory is placed), which would be ignorant of the contents of the enciphered messages; it would fear that the cipher contains information other than the agreed-upon seismic data. If the host nation were given the key to a symmetric encryption system (so that it could decrypt the cipher and verify the

message content), it would also, by definition, be able to generate counterfeit ciphers. A compromise solution is to form an authenticator much shorter than the entire message; the authenticator depends on all of the symbols in the message through some hashing function. The authenticator is also encrypted. (The block chaining technique was implemented in such a solution in the late 1960s for a similar application.) The shorter authenticator (cipher) is of course still inscrutable to the host nation, but its smaller size means that less information could be concealed in each transmission. Periodically, the hashing algorithm and key could be changed; the hashing algorithm and key used in the previous period would be given to the host, which could then verify that the authenticators had not concealed unauthorized information in the previous period. After satisfying itself that the system had not been misused, the host would renew the license to operate for one more period. This compromise is not completely satisfying to both parties because the host nation still must trust the other nation not to begin concealing information in the current authenticators.

The problem can be solved completely with either a forward or a bidirectional asymmetric encryption system. The message M and the cipher $\mathbf{E}(M, K)$ are given to the host nation, which has already been given $\mathbf{D}_E$ and K', but not K. The host would compare $\mathbf{D}_E(\mathbf{E}(M, K), K')$ with the purported message M. If the two agree, the host is assured of the content of the message. The other nation also compares $\mathbf{D}_E(\mathbf{E}(M, K), K')$ and M to determine if the message is authentic.

Class IV is typified by commercial transactions in which it is essential to be certain both that the message came from the purported transmitter and that it has not been altered in transmission—and also to ensure that outsiders are not privy to the communication. Since all the secure communications objectives are met in such a system, we call this the *secure channel.*

There are many business applications in which a secure channel is desirable, for example, the remote automatic bank teller or the control of access to a computer's unsecured data files. In these cases the user would like to be certain that no one can wiretap the communication link while he is authenticating himself and then later be able to impersonate him to the bank's computer or to the CPU. Secure log-in computer systems require the user to identify himself before granting him access to the operating computer system [HOFF77, MART73], but these systems may be complex. Many low-security systems simply store all user numbers and the corresponding passwords in a file normally inaccessible to users. Anyone gaining (illegal) access to this file could then impersonate any system user. The most common defense is the one-way cipher [EVAN74, PURD74, WILK68], which does not store the user's password W_i, but rather a function $\mathbf{E}(W_i)$, where $\mathbf{E}$ is chosen to be computationally infeasible to invert. Anyone gaining access to the password file would know $\mathbf{E}(W_i)$ for all the authorized users but would be unable to determine any W_i and hence unable to impersonate any user. Obviously, there are requirements other than the difficulty of inverting $\mathbf{E}$; for instance, the file can contain only a vanishingly small fraction of the total number of possible passwords; otherwise the opponent could simply choose a random collection of W_i, form the corresponding $\mathbf{E}(W_i)$, and if a match were found in the file, use that identity. This type of system has generally been adopted by the banking industry for "window identification" of passcard holders for savings accounts.

The requirement for a full-fledged secure channel arises with the brokerage house that responds to either a very large buy or sell order. The house wants the highest possible level of secrecy concerning the details of the order lest it disturb the market. The house also wants full authentication of the giver of the order. Private commercial codes were once used for precisely these purposes; these codes, however, provide little cryptosecurity.

As further illustration of the requirements on secure channels, consider a military commander who sends scouting patrols into enemy territory. A two-way radio communication link exists between each patrol and the command post, and all the patrols use the same asymmetric system.

Before the mission is completed, some of the patrols may have been captured and their cryptosystems divulged. Communication from the uncompromised patrols to headquarters remains secret because only the transmitter's key has been compromised. Moreover, the enemy cannot impersonate the commander's messages because it knows only a receiver's key.

Now, suppose that a hybrid cryptosystem is used. The first communication over the asymmetric channel from a patrol to the commander could be a key, for example, a 56-bit random number for the DES symmetric cryptosystem. This communication is in secret since only the transmitter key could have been compromised for this channel. Thereafter the commander and patrol can engage in a secure two-way communication over the symmetric channel using the new "session" key. This is not possible using the asymmetric system alone because the commander's ciphers may be legible to the enemy. This system is not foolproof, however, because the commander has no way to authenticate the patrol initiating the communication. Some other concealed information, such as a sign or countersign, could be used, but this additional information would be considered to be a part of the key according to the strict definition given earlier and hence may have been divulged to the enemy.

The foregoing discussion assumes that the sender and receiver are sure of each other's identity and keys—for example, a higher level commander has generated the keys, or each user has generated his own pair of keys. Needham and Schroeder [NEED78] have shown that the secure distribution of keys is essential to cryptosecurity and is the same for symmetric and asymmetric systems. The following example illustrates the possibility that completely anonymous communicants can enter into a private conversation. Let $\mathscr{E}$ be a class of commutative encryption functions,[16] i.e., $\mathbf{E}_A, \mathbf{E}_B \in \mathscr{E}$ implies $\mathbf{E}_A(\mathbf{E}_B(M, K_B), K_A) = \mathbf{E}_B(\mathbf{E}_A(M, K_A), K_B)$. If A wishes to communicate a message M to B in secrecy where no advance arrangements such as key distribution or public-key disclosure have been made, A chooses $\mathbf{E}_A$, $\mathbf{D}_A$, and K_A and K_A'. He then transmits the cipher $\mathbf{E}_A(M, K_A)$ to B, who cannot decrypt the cipher. Now B chooses $\mathbf{E}_B$, $\mathbf{D}_B$, and K_B and K_B' from the family of commutative encryption functions and transmits the cipher $\mathbf{E}_B(\mathbf{E}_A(M, K_A), K_B)$ to A. A computes $\mathbf{D}_A(\mathbf{E}_B(\mathbf{E}_A(M, K_A), K_B), K_A')$, which reduces to $\mathbf{E}_B(M, K_B)$ because $\mathbf{D}_A$ "undoes" $\mathbf{E}_A$. Then A relays this cipher back to B, who computes $\mathbf{D}_B(\mathbf{E}_B(M, K_B), K_B')$ to recover M. On the surface it appears that an impossible result has been accomplished because the keys were kept secret all through the exchange. In fact, A has communicated in secret to whomever responded to his original transmission of the cipher $\mathbf{E}_A(M, K_A)$, but A cannot establish the identity of his receiver. In other words, A can only be certain that he has a private communication with an unknown party.

Perhaps the most intriguing example of this paradox of initiating secret communications between two parties who cannot establish each other's identities occurs in Shamir, Rivest, and Adleman's protocol for playing mental poker [SHAM79]. In this case the names of the cards are encrypted by player A and the resulting ciphers passed to B who chooses a random subset (deal), etc., to relay to B using a commutative encryption function as described in the preceding paragraph. The resulting game is self-consistent in the sense that the players can verify that a game of poker is being played fairly—but with an unknown opponent.

The point of the preceding three paragraphs is to illustrate an essential point about asymmetric encryption systems. *It is not true* that "in a public-key cryptosystem[17] there is no need of a secure channel

[16] An example of a commutative cryptosystem is a variant of the Pohlig-Hellman log-antilog scheme over large finite fields [POHL78] Let $\mathscr{M} = \{GF(2^{127})/\{0, 1\}\}$ be the message space known to everyone. A selects an exponent $2 \leq e \leq 2^{127} - 2$ and encrypts M as M^e in $GF(2^{127})$. B chooses an exponent d similarly and relays $(M^e)^d$ (also in $GF(2^{127})$), which A then raises to the e^{-1} power to get $M^d = ((M^e)^d)^{e^{-1}}$, which is retransmitted to B who computes $(M^d)^{d^{-1}}$ to obtain M. An opponent will have seen M^e, M^d, and $(M^e)^d$ and will know the space $\mathscr{M}$, so he is faced with the "known plaintext" decryption problem with the twist that he knows two messages which encrypt to a common cipher.

[17] Read asymmetric cryptosystem

for the distribution of keys" [HELL79b]. What is true is that whereas the secure key distribution system must be able to certify the secrecy of the delivered key for use in symmetric systems, it need only be able to certify the authenticity of the key for asymmetric systems. There is implicit in this statement a distinction between a passive wiretapper (eavesdropper) who only listens to but does not originate ciphers and an active wiretapper who may alter or originate ciphers. An eavesdropper listening to the microwave scatter from a microwave link illustrates the first threat, while a wiretapper in a central switching office illustrates the second. In the case of the active wiretapper, the only way to avoid the "postal chess ploy"[18] is to have the keys delivered securely, either in a face-to-face exchange by the transmitter and receiver or by trusted couriers, etc.

SUMMARY AND CONCLUSION

The primary objectives in this paper have been to develop the concept of the asymmetric encryption/decryption channel and to show some real problems that can only be solved by using such a channel. A secondary objective has been to draw analogies between coding theory and encryption theory in order to clarify the concepts of secrecy and authentication.

Cryptosystems are naturally classified into two classes, symmetric or asymmetric, depending only on whether the keys at the transmitter and receiver are easily computed from each other. The only well-tested operational cryptosystems in 1979 were symmetric. All depend on the computational intractability of working backward from a knowledge of the cipher, plaintext, and encryption/decryption function for their cryptosecurity. Asymmetric cryptosystems are inherently neither more nor less secure than symmetric cryptosystems. Both kinds of system depend on the high "work factor" associated with a computationally infeasible problem to provide computational cryptosecurity. An essential difference between symmetric and asymmetric cryptosystems is that one of the transmitter or receiver keys can be compromised in the asymmetric system with some secure communications still possible. In some instances, such as the public-key cryptosystem, the exposure may be deliberate; in others it cannot be insured against simply because of the physical exposure of one end of the communications link. If in an asymmetric system the receiver key is concealed from a knowledge of the transmitter key, it is still possible to communicate in secrecy even after the transmitter key is exposed. Conversely, if the transmitter key is concealed from a knowledge of the receiver key, it is possible for the transmitter to authenticate himself even though the receiver key is known to an opponent. These unique capabilities of asymmetric systems distinguish them from symmetric systems.

Two vital points need to be restated. First, it is false that key protection and secure key dissemination are unnecessary in an asymmetric system. As Needham and Schroeder [NEED78] have shown for network authentication, the protocols are quite similar, and the number of protocol messages which must be exchanged is comparable using either symmetric or asymmetric encryption techniques. At the end of the section on secure communications we illustrated an anomaly, the establishing of a secret link with a party whose identity cannot be verified, which can arise in the absence of key dissemination. For this reason asymmetric techniques can be used to disseminate a key which is then used in a symmetric system.

The second point is that asymmetric systems are not a priori superior to symmetric ones. The particular application determines which system is appropriate. In the 1979 state of the art, all the proposed asymmetric systems exact a high price for their asymmetry: The higher amount of computation in the encryption/decryption process significantly cuts the channel capacity (bits per second of message information communicated). No asymmetric scheme known to the author has a capacity better than $C^{1/2}$, where C is the channel capacity of a symmetric channel having the same cryp-

[18] In this scheme a third party interposes himself simply to relay moves in the correspondence of two postal chess players with a guarantee of either drawing against both or else winning against one while losing to the other, irrespective of his chess playing abilities

tosecurity and using the same basic clock or bit manipulation rate. Under these conditions, the higher overhead of asymmetric encryption is warranted only for applications in which one of the communications terminals is physically insecure.

APPENDIX

The following brief discussion of LFSRs is included for the benefit of readers who may not be familiar with the inner workings of these devices. Given an nth-order nonhomogeneous polynomial, i.e., $P^n(x) = \sum_{i=0}^{n} c_i x^i$, where $c_0 = c_n = 1$, with binary coefficients,[19] we define an associated n-stage linear feedback shift register by the rules

$$x_1^t = \sum_{i=1}^{n} c_i x_i^{t-1}$$

and

$$x_i^t = x_{i-1}^{t-1}, \qquad i > 1$$

where x_i^t is the state of the ith stage of the register on the tth step and Σ is the modulo 2 sum (binary arithmetic). For example, if $P^4(x) = x^4 + x^3 + x^2 + x + 1$, the shift register is of the form shown in Figure 7 and the sequence of states of the register (depending on the initial fill) is one of four cycles:

0000	1000	0100	1110
	0001	1001	1101
	0011	0010	1011
	0110	0101	0111
	1100	1010	1111

In this case the 16 possible 4-bit binary numbers are divided into three cycles of length 5 and one of length 1. The explanation is that $x^4 + x^3 + x^2 + x + 1$ divides $x^5 + 1$ evenly; i.e.,

$$(x + 1)(x^4 + x^3 + x^2 + x + 1) = x^5 + 1.$$

Note: Remember that the coefficients are treated as residues modulo 2.

A well-known result from algebra says that $P^n(x)$ always divides $x^{2^n-1} + 1$, but

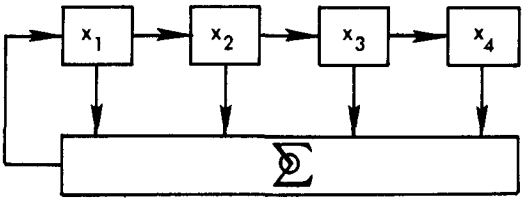

FIGURE 7.

that $P^n(x)$ may also divide $x^d + 1$ where d is a divisor of $2^n - 1$, in which case the maximum period of the sequences from the associated LFSR is also a proper divisor of $2^n - 1$. If the polynomial $P^n(x)$ has no factors and does not divide $x^d + 1$ for any proper divisor d of $2^n - 1$, then $P^n(x)$ is said to be primitive. The important point is that the nonzero cycle generated by the associated linear feedback shift register for any primitive polynomial has the maximum possible period of $2^n - 1$: $00 \cdots 0$ is always in a cycle by itself. For example, $P^4(x) = x^4 + x + 1$ divides $x^{15} + 1$ but not $x^d + 1$ for any $d < 15$; hence $P^4(x)$ is primitive and the maximal length nonzero cycle generated by the associated LFSR is:

1000	0101
0001	1011
0011	0110
0111	1100
1111	1001
1110	0010
1101	0100
1010	

Linear feedback shift registers based on primitive polynomials are therefore said to be maximal length, and the resulting bit sequences have been shown to satisfy many tests for randomness [GOLO67, TAUS65]. For example, 0, 1 and 00, 01, 10, 11, etc. (up to n-tuples), are as nearly uniform in their probability of occurrence as is possible; i.e., since the all-zero n-tuple is not in the cycle, the all-zero k-tuple will occur one time less than do the other k-tuples. Because of these very useful properties and also because of the ease of implementing maximal length LFSRs in either hardware or software, a voluminous literature exists on the subject—including extensive tables of the primitive polynomials [GOLO67, PETE72] needed to compute the feedback functions.

[19] Modulo 2 using the rules

+	0	1
0	0	1
1	1	0

×	0	1
0	0	0
1	0	1

An especially simple class of primitive polynomial [ZIER68, ZIER69], both to analyze and to implement, is the trinomials, $x^n + x^a + 1$, which require only two stages of the feedback shift register to be tapped and combined by an Exclusive OR

⊕	0	1
0	0	1
1	1	0

to compute the feedback sum.

ACKNOWLEDGMENTS

The author wishes to acknowledge the many and valuable contributions of M J. Norris to the ideas presented here. He is also grateful to D. Kahn and H. Bright for careful reviews of a first draft of the manuscript and to the anonymous referees whose detailed suggestions materially shaped the present form of the paper. Finally, he wishes to express his appreciation to R. J. Hanson and P. J. Denning whose assistance has made it possible for this material to be published in *Computing Surveys.*

REFERENCES

ACME23 *Acme commodity and phrase code*, Acme Code Co., San Francisco, Calif., 1923.

ADLE78 ADLEMAN, L. M., AND RIVEST, R L "The use of public-key cryptography in communication system design," *IEEE Trans Commun.* **COM-16,** 6 (Nov 1978), 20-23.

ALBE41 ALBERT, A. A "Some mathematical aspects of cryptography," presented at the AMS 382nd Meeting, Manhattan, Kans., Nov 22, 1941.

BERL68 BERLEKAMP, E. R. *Algebraic coding theory*, McGraw-Hill, New York, 1968.

BRAN79 BRANSTAD, D. "Hellman's data does not support his conclusion," *IEEE Spectrum* **16,** 7 (July 1979), 41

BRIG76 BRIGHT, H S, AND ENISON, R L. "Cryptography using modular software elements," in *Proc AFIPS 1976 NCC,* Vol. 45, AFIPS Press, Arlington, Va , pp 113-123

BRIG77 BRIGHT, H. S. "Cryptanalytic attack and defense. ciphertext-only, known-plaintext, chosen-plaintext," *Cryptologia* **1,** 4 (Oct 1977), 366-370.

DAVI79 DAVIDA, G. I. "Hellman's scheme breaks DES in its basic form," *IEEE Spectrum* **16,** 7 (July 1979), 39.

DEAV77 DEAVOURS, C. A. "Unicity points in cryptanalysis," *Cryptologia* **1,** 1 (Jan 1977), 46-68

DIFF76 DIFFIE, W , AND HELLMAN, M E. "New directions in cryptography," *IEEE Trans Inform. Theory* **IT-22,** 6 (Nov. 1976), 644-654.

DIFF77 DIFFIE, W., AND HELLMAN, M. E "Exhaustive cryptanalysis of the NBS data encryption standard," *Computer* **10,** 6 (June 1977), 74-84.

EVAN74 EVANS, A , JR., AND KANTROWITZ, W. "A user authentication scheme not requiring secrecy in the computer," *Commun ACM* **17,** 8 (Aug. 1974), 437-442.

FEIS73 FEISTEL, H. "Cryptography and computer privacy," *Sci. Am.* **228,** 5 (May 1973), 15-23.

GAIN56 GAINES, H. F. *Cryptanalysis: a study of ciphers and their solution*, Dover, New York, 1956.

GAIT77 GAIT, J "A new nonlinear pseudorandom number generator," *IEEE Trans Softw Eng.* **SE-3,** 5 (Sept. 1977), 359-363

GARD77 GARDNER, M. Mathematical games (section), *Sci. Am.* **237,** 2 (Aug 1977), 120-124.

GEFF73 GEFFE, P. R. "How to protect data with ciphers that are really hard to break," *Electronics* **46,** 1 (Jan. 4, 1973), 99-101.

GILB74 GILBERT, E. N., MACWILLIAMS, F J., AND SLOANE, N. J. A "Codes which detect deception," *Bell Syst Tech. J.* **53,** 3 (March 1974), 405-423.

GOLO67 GOLOMB, S W. *Shift register sequences*, Holden-Day, San Francisco, Calif., 1967.

HART64 HART, G L *The Beale papers*, Roanoke Public Library, Roanoke, Va , 1964

HELL78 HELLMAN, M. E "An overview of public-key cryptography," *IEEE Trans. Commun* **COM-16,** 6 (Nov. 1978), 24-32.

HELL79a HELLMAN, M. E. "DES will be totally insecure within ten years," *IEEE Spectrum* **16,** 7 (July 1979), 32-39.

HELL79b HELLMAN, M. E "The mathematics of public-key cryptography," *Sci. Am.* **241,** 3 (Aug. 1979), 146-157.

HERL78 HERLESTAM, T. "Critical remarks on some public-key cryptosystems," *BIT* **18** (1978), 493-496

HILL29 HILL, L. S "Cryptography in an algebraic alphabet," *Am. Math. Monthly* **36** (June-July 1929), 306-312.

HILL31 HILL, L. S. "Concerning certain linear transformation apparatus of cryptography," *Am Math. Monthly* **38** (March 1931), 135-154.

HOFF77 HOFFMAN, L. J. *Modern methods for computer security and privacy*, Prentice-Hall, Englewood Cliffs, N J., 1977

HORO74 HOROWITZ, E., AND SAHNI, S. "Computing partitions with applications to the knapsack problem," *J. ACM* **21,** 2 (April 1974), 277-292

KAHN66 KAHN, D. "Modern cryptology," *Sci Am.* **215** (July 1966), 38-46

KAHN67 KAHN, D. *The codebreakers, the story of secret writing*, MacMillan, New York, 1967

KARP72 KARP, R. M. "Reducibility among combinatorial problems," in *Complexity of computer computations*, R. E Miller and J. W Thatcher (Eds.), Plenum Press, New York, 1972, pp. 85-104.

KULL76 KULLBACK, S *Statistical methods in cryptanalysis*, Aegean Park Press, Laguna Hills, Calif , 1976.

LEMP79 LEMPEL, A "Cryptology in transition: a survey," *Comput. Surv.* **11,** 4 (Dec. 1979), 285-304.

LIPT78 LIPTON, S M., AND MATYAS, S. M "Making the digital signature legal—and safeguarded," *Data Commun.* **7,** 2 (Feb 1978), 41-52.

MACW77 MACWILLIAMS, F J., AND SLOANE, N. J. A. *The Theory of error-correcting codes*, Vols. I and II, North-Holland, New York, 1977.

MART73 MARTIN, J. *Security, accuracy and privacy in computing systems*, Prentice-Hall, Englewood Cliffs, N J., 1973.

MASS69 MASSEY, J. L "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory* **IT-15,** 1 (Jan. 1969), 122–127.

MERK78a MERKLE, R C. "Secure communications over insecure channels," *Commun. ACM* **21,** 4 (April 1978), 294–299.

MERK78b MERKLE, R. C, AND HELLMAN, M. E "Hiding information and signatures in trapdoor knapsacks," *IEEE Trans. Inform Theory* **IT-24,** 5 (Sept. 1978), 525–530.

MEYE72 MEYER, C, AND TUCHMAN, W. "Pseudo-random codes can be cracked," *Electron Des.* **23** (1972), 74–76.

MORR77 MORRIS, R., SLOANE, N. J A., AND WYNER, A. D "Assessment of the National Bureau of Standards proposed federal Data Encryption Standard," *Cryptologia* **1,** 3 (July 1977), 281–291.

NEED78 NEEDHAM, R. M., AND SCHROEDER, M. D. "Using encryption for authentication in large networks of computers," *Commun. ACM* **21,** 12 (Dec. 1978), 993–999

PETE72 PETERSON, W. W., AND WELDON, E. J *Error correcting codes*, 2nd ed., MIT Press, Cambridge, Mass , 1972

POHL78 POHLIG, S C, AND HELLMAN, M E. "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance," *IEEE Trans Inform Theory* **IT-24,** 1 (Jan 1978), 106–110

PURD74 PURDY, G. B "A high security log-in procedure," *Commun. ACM* **17,** 8 (Aug 1974), 442–445.

RABI79 RABIN, M. O. *Digitalized signatures and public-key functions as intractable as factorization*, Tech Rep MIT/LCS/TR-212, MIT Lab Comput Sci., Cambridge, Mass , Jan 1979.

RIVE78 RIVEST, R., SHAMIR, A., AND ADLEMAN, L. "A method for obtaining digital signatures and public-key cryptosystems," *Commun ACM* **21,** 2 (Feb 1978), 120–126.

ROBE75 ROBERTS, R. W. Encryption algorithm for computer data encryption," (*NBS*) *Fed. Reg.* **40,** 52 (March 17, 1975), 12134–12139

SCHR79 SCHROEPPEL, R., AND SHAMIR, A. "A $T \cdot S^2 = O(2^n)$ time/space tradeoff for certain NP-complete problems," to appear as MIT Lab. Comput Sci Rep.

SHAM78 SHAMIR, A., AND ZIPPEL, R. E *On the security of the Merkle-Hellman cryptographic scheme*, Tech. Rep. MIT/LCS/TM-119, MIT Lab. Comput. Sci., Cambridge, Mass., Dec. 1978.

SHAM79 SHAMIR, A., RIVEST, R. L., AND ADLEMAN, L. M. *Mental poker*, Tech. Rep. MIT/LCS/TM-125, MIT Lab. Comput. Sci., Cambridge, Mass., Feb. 1979.

SHAN48 SHANNON, C. E "A mathematical theory of communication," *Bell Syst. Tech. J.* **27** (July 1948), 379–423; (Oct. 1948), 623–656.

SHAN49 SHANNON, C. E. "Communication theory of secrecy systems," *Bell Syst. Tech. J.* **28** (Oct. 1949), 656–715.

SHAP78 SHAPLEY, D. "The new unbreakable codes—will they put NSA out of business?," *The Washington Post*, Outlook, sec B1, July 9, 1978

SIMM77 SIMMONS, G. J , AND NORRIS, M. J. "Preliminary comments on the M I.T. public-key cryptosystem," *Cryptologia* **1,** 4 (Oct. 1977), 406–414.

SIMM79 SIMMONS, G. J. "Cryptology the mathematics of secure communication," *Math. Intell.* **1,** 4 (Jan 1979), 233–246

SUGA79 SUGARMAN, R "On foiling computer crime," *IEEE Spectrum* **16,** 7 (July 1979), 31–32.

TAUS65 TAUSWORTHE, R. C "Random numbers generated by linear recurrence modulo two," *Math Comput.* **19** (1965), 201–209

TUCH79 TUCHMAN, W "Hellman presents no shortcut solutions to the DES," *IEEE Spectrum* **16,** 7 (July 1979), 40–41.

TUCK70 TUCKERMAN, B. *A study of the Vigenére-Vernam single and multiple loop enciphering systems*, Rep. RC-2879 (#13538), IBM T. J. Watson Res. Ctr., Yorktown Heights, N.Y., May 14, 1970.

VERN26 VERNAM, G. S. "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *J AIEE* **45** (Feb. 1926), 109–115.

WILK68 WILKES, M. V *Time-sharing computer systems*, American Elsevier, New York, 1968

WILL79a WILLIAMS, H. C., AND SCHMID, B. *Some remarks concerning the M.I.T. public-key cryptosystem*, Rep. 91, U. of Manitoba Dep. of Comput Sci., May 22, 1979.

WILL79b WILLIAMS, H. C. *A modification of the RSA public-key encryption procedure*, Rep. 92, U. of Manitoba Dep of Comput. Sci., 1979.

ZIER68 ZIERLER, N., AND BRILLHART, J. "On primitive trinomials (mod 2)," *Inform. Control* **13** (1968), 541–554.

ZIER69 ZIERLER, N., AND BRILLHART, J. "On primitive trinomials (mod 2, II)," *Inform. Control* **14** (1969), 566–569.

RECEIVED NOVEMBER 1978, FINAL REVISION ACCEPTED AUGUST 1979