# An Introduction to Contemporary Cryptology

JAMES L. MASSEY, FELLOW, IEEE

*Invited Paper*

*An appraisal is given of the current status, both technical and nontechnical, of cryptologic research. The principal concepts of both secret-key and public-key cryptography are described. Shannon's theory of secrecy and Simmons's theory of authenticity are reviewed for the insight that they give into practical cryptographic systems. Public-key concepts are illustrated through consideration of the Diffie–Hellman public-key-distribution system and the Rivest–Shamir–Adleman public-key cryptosystem. The subtleties of cryptographic protocols are shown through consideration of some specific such protocols.*

## I. Preliminaries

### A. Introduction

That cryptology is a "hot" research area hardly needs saying. The exploits of cryptographic researchers are reported today not only in an increasing number of scholarly journals and popular scientific magazines, but also in the public press. One hears of conflicts between cryptologic researchers and government security agencies, insinuations of built-in "trapdoors" in commonly used ciphers, claims about new ciphers that would take millions of years to break and counterclaims that no cipher is secure—all the stuff of high drama. To ferret out the truth in such controversies, one needs a basic understanding of cryptology, of its goals and methods, and of its capabilities and limitations. The aim of this paper is to provide a brief, self-contained introduction to cryptology that may help the reader to reach such a basic understanding of the subject, and that may give him or her additional insight into the more specialized papers on cryptology that form the rest of this special section.

Only scant attention will be given in this paper to the long and rich history of cryptology. For an excellent short history, the reader is referred to that given in a splendid survey of cryptology that appeared earlier in these pages [1] or that in an unusually penetrating encyclopedia article [2]. But Kahn's voluminous history, *The Codebreakers* [3], is indispensable to anyone who wishes to dig deeply into cryptologic history. The abridged paperback edition [4] of Kahn's book can be especially recommended as it packs as much suspense as the best spy fiction has to offer, but will also satisfy the historical curiosity of most readers.

Manuscript received March 3, 1987; revised July 30, 1987.
The author is with the Institute for Signal and Information Processing, Swiss Federal Institute of Technology, CH-8092 Zurich, Switzerland.
IEEE Log Number 8821262.

### B. Cryptologic Nomenclature and Assumptions

The word, cryptology, stems from Greek roots meaning "hidden" and "word," and is the umbrella word used to describe the entire field of secret communications. For instance, the five-year-old scientific society formed by researchers in this field is appropriately called the International Association for Cryptologic Research.

Cryptology splits rather cleanly into two subdivisions: cryptography and cryptanalysis. The cryptographer seeks to find methods to ensure the secrecy and/or authenticity of messages. The cryptanalyst seeks to undo the former's work by breaking a cipher or by forging coded signals that will be accepted as authentic. The original message upon which the cryptographer plies his art is called the plaintext message, or simply the *plaintext;* the product of his labors is called the ciphertext message, or just the *ciphertext* or, most often, the *cryptogram.* The cryptographer always employs a *secret key* to control his enciphering process. He often (but not always) delivers the secret key by some secure means (e.g., in an attaché case handcuffed to the wrists of a courier) to the person (or machine) to whom he expects later to send a cryptogram formed using that key.

The almost universal assumption of cryptography is that the enemy cryptanalyst has full access to the cryptogram. Almost as universally, the cryptographer adopts the precept, first enunciated by the Dutchman A. Kerckhoff (1835–1903), that the security of the cipher must reside entirely in the secret key. Equivalently, *Kerckhoff's assumption* is that the entire mechanism of encipherment, except for the value of the secret key, is known to the enemy cryptanalyst. If the cryptographer makes only these two assumptions, then he is designing his system for security against a *ciphertext-only* attack by the enemy cryptanalyst. If the cryptographer further assumes that the enemy cryptanalyst will have acquired ("by hook or by crook") some plaintext-cryptogram pairs formed with the actual secret key, then he is designing against a *known-plaintext attack.* The cryptographer may even wish to assume that the enemy cryptanalyst can sub-

0018-9219/88/0500-0533$01.00 © 1988 IEEE

mit any plaintext message of his own and receive in return the correct cryptogram for the actual secret key (a *chosen-plaintext attack*), or to assume that the enemy cryptanalyst can submit purported "cryptograms" and receive in return the unintelligible garble to which they (usually) decrypt under the actual key (a *chosen-ciphertext attack*), or to assume both of these possibilities (a *chosen-text attack*). Most cipher systems in use today are intended by their designers to be secure against at least a chosen-plaintext attack, even if it is hoped that the enemy cryptanalyst will never have the opportunity to mount more than a ciphertext-only attack.

### C. *The Need for Cryptology*

Cryptography has been used for millenia to safeguard military and diplomatic communications. Indeed, the obvious need for cryptography in the government sector led to the rather general acceptance, until quite recently, of cryptography as a prerogative of government. Most governments today exercise some control of cryptographic apparatus if not of cryptographic research. The U.S., for instance, applies the same export/import controls to cryptographic devices as to military weapons. But the dawning of the Information Age revealed an urgent need for cryptography in the private sector. Today vast amounts of sensitive information such as health and legal records, financial transactions, credit ratings and the like are routinely exchanged between computers via public communication facilities. Society turns to the cryptographer for help in ensuring the privacy and authenticity of such sensitive information.

While the need for cryptography in both the government and private sectors is generally accepted, the need for cryptanalysis is less well acknowledged. "Gentlemen do not read each other's mail," was the response of U.S. Secretary of State H. L. Stimson in 1929 upon learning that the U.S. State Department's "Black Chamber" was routinely breaking the coded diplomatic cables of many countries. Stimson forthwith abolished the Black Chamber, although as Secretary of War in 1940 he relented in his distaste of cryptanalysis enough to condone the breaking of Japanese ciphers [4, p. 178]. In today's less innocent world, cryptanalysis is generally regarded as a proper and prudent activity in the government sector, but as akin to keyhole-peeping or industrial espionage in the private sector. However, even in the private sector, cryptanalysis can play a valuable and ethical role. The "friendly cryptanalyst" can expose the unsuspected weaknesses of ciphers so that they can be taken out of service or their designs remedied. A paradigm is Shamir's recent breaking of the Merkle-Hellman trapdoor-knapsack public-key cryptosystem [5]. By publishing his ingenious cryptanalysis [6] of this clever and very practical cipher, Shamir forestalled its likely adoption in practice with subsequent exposure to the attacks of cryptanalysts seeking rewards more tangible than scientific recognition. Shamir's reward was the 1986 IEEE W. R. G. Baker Award.

### D. *Secret and Open Cryptologic Research*

If one regards cryptology as the prerogative of government, one accepts that most cryptologic research will be conducted behind closed doors. Without doubt, the number of workers engaged today in such secret research in cryptology far exceeds that of those engaged in open research in cryptology. For only about ten years has there in fact been widespread open research in cryptology. There have been, and will continue to be, conflicts between these two research communities. Open reseach is a common quest for knowledge that depends for its vitality on the open exchange of ideas via conference presentations and publications in scholarly journals. But can a government agency, charged with the responsibility of breaking the ciphers of other nations, countenance publication of a cipher that it could not break? Can a researcher in good conscience publish such a cipher that might undermine the effectiveness of his own government's code-breakers? One might argue that publication of a provably-secure cipher would force all governments to behave like Stimson's "gentlemen," but one must be aware that open research in cryptology is frought with political and ethical considerations of a severity much greater than in most scientific fields. The wonder is not that some conflicts have occurred between government agencies and open researchers in cryptology, but rather that these conflicts (at least those of which we are aware) have been so few and so mild.

One can even argue that the greatest threat to the present vigorous open cryptologic research activity in the U.S. stems not from the intransigence of government but rather from its largesse. A recent U.S. government policy will require governmental agencies to rely on cryptographic devices at whose heart are tamperproof modules incorporating secret algorithms devised by the National Security Agency (NSA) and loaded with master keys distributed by NSA [7]. Moreover, NSA will make these modules available to certified manufacturers for use in private-sector cryptography, and will presumably also supply the master keys for these applications. If, as appears likely, these systems find widespread acceptance in the American private sector, it will weaken the practical incentive for further basic open research in cryptography in the U.S. The main practical application for such research will be restricted to international systems where the NSA technology will not be available.

### E. *Epochs in Cryptology*

The entire period from Antiquity until 1949 can justly be regarded as the *era of prescientific cryptology*; which is not to say that the cryptologic history of these times is devoid of interest today, but rather that cryptology was then plied almost exclusively as an art rather than as a science. Julius Caesar wrote to Cicero and his other friends in Rome more than 2000 years ago, employing a cipher in which each letter in the plaintext was replaced by the third (cyclically) later letter in the Latin alphabet [4, p. 77]. Thus, the plaintext CAESAR would yield the ciphertext FDHVDU. Today, we would express Caesar's cipher as

$$y = x \oplus z \tag{1}$$

where $x$ is the plaintext letter ($A = 0, B = 1, \cdots, Z = 25$), $z$ is the secret key (which Julius Caesar always chose as 3—Caesar Augustus chose 4), $y$ is the ciphertext letter, and $\oplus$ here denotes addition modulo 26 (so that $23 \oplus 3 = 0$, $23 \oplus 4 = 1$, etc.). There is no historical evidence to suggest that Brutus broke Caesar's cipher, but a schoolchild today, who knew a little Latin and who had read the elementary cryptanalysis described in Edgar Allen Poe's masterful short story, "The Gold-Bug," would have no difficulty to succeed in a ciphertext-only attack on a few sentences of ciphertext. In fact, for the next almost two thousand years after Caesar, the cryptanalysts generally had a clear upper hand over the cryptographers. Then, in 1926, G. S. Vernam, an engineer with the American Telephone and Telegraph Company published a remarkable cipher to be used with the binary Baudot code [8]. Vernam's cipher is similar to Caesar's in that it is described by (1), except that now $x$, $y$, and $z$ take values in the binary alphabet $\{0, 1\}$ and $\oplus$ denotes addition modulo-two ($0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 1 = 0$). The new idea advanced by Vernam was to *use the key only one time*, i.e., to encipher each bit of ciphertext with a new randomly-chosen bit of key. This necessitates the secure transfer of as much secret key as one will later have plaintext to encipher, but it yields a truly unbreakable cipher as we shall see later. Vernam indeed believed that his cipher was unbreakable and was aware that it would not be so if the randomly chosen key bits were to be reused later, but he offered no proofs of these facts. Moreover, he cited in [8] field tests that had confirmed the unbreakability of his cipher, something no amount of field testing could in fact confirm. Our reason for calling the period up to 1949 the prescientific era of cryptology is that cryptologists then generally proceeded by intuition and "beliefs," which they could not buttress by proofs. It was not until the outbreak of World War II, for instance, that the English cryptological community recognized that mathematicians might have a contribution to make to cryptology [8, p. 148] and enlisted among others, A. Turing, in their service.

The publication in 1949 by C. E. Shannon of the paper, "Communication Theory of Secrecy Systems" [10], ushered in the *era of scientific secret-key cryptology*. Shannon, educated both as an electrical engineer and mathematician, provided a theory of secrecy systems almost as comprehensive as the theory of communications that he had published the year before [11]. Indeed, he built his 1949 paper on the foundation of the 1948 one, which had established the new discipline of information theory. Shannon not only proved the unbreakability of the random Vernam cipher, but also established sharp bounds on the required amount of secret key that must be transferred securely to the intended receiver.

For reasons that will become clear in the sequel, Shannon's 1949 paper did not lead to the same explosion of research in cryptology that his 1948 paper had triggered in information theory. The real explosion came with the publication in 1976 by W. Diffie and M. E. Hellman of their paper, "New Directions in Cryptography" [12]. Diffie and Hellman showed for the first time that secret communications was possible without any transfer of a secret key between sender and receiver, thus establishing the turbulent *epoch of public-key cryptography* that continues unabated today. R. C. Merkle, who had submitted his paper about the same time as Diffie and Hellman but to another journal, independently introduced some of the essential ideas of public-key cryptography. Unfortunately, the long delay in publishing his paper [13] has often deprived him of due scientific credit.

### *F. Plan of this Paper*

In the next section, we review briefly the theory of secret key cryptography, following essentially Shannon's original approach and making Shannon's important distinction between theoretical and practical security. We also indicate the directions of some contemporary research in secret-key cryptography. Section III gives a short exposition of public-key cryptography, together with a description of some of the most important public-key systems thus far advanced. In Section IV we touch upon the delicate subject of cryptographic protocols, and show how cryptographic techniques can be used to accomplish nonstandard, but very useful, tasks.

## II. Secret-Key Cryptography

### *A. Model and Notation*

By a secret-key cryptosystem, we mean a system that corresponds to the block diagram of Fig. 1. The essential feature of such a system is the "secure channel" by which the

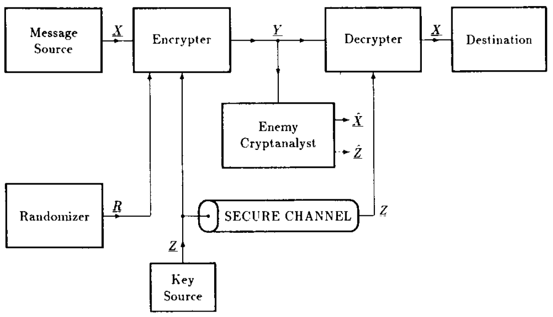

**Fig. 1.** Model of a secret-key cryptosystem.

secret key, $\boldsymbol{Z} = [Z_1, Z_2, \cdots, Z_K]$, after generation by the *key source*, is delivered to the intended receiver, protected from the prying eyes of the enemy cryptanalyst. To emphasize that the same secret key is used by both the encrypter and decrypter, secret-key cryptosystems have also been called *one-key cryptosystems* and *symmetric cryptosystems*. The $K$ digits of the key are letters in some finite alphabet that we will often choose to be the binary alphabet $\{0, 1\}$. The *message source* generates the plaintext, $\boldsymbol{X} = [X_1, X_2, \cdots, X_M]$; the *randomizer* (whose purpose will soon be explained) generates the *randomizing sequence*, $\boldsymbol{R} = [R_1, R_2, \cdots, R_J]$. The encrypter forms the cryptogram, $\boldsymbol{Y} = [Y_1, Y_2, \cdots, Y_N]$, as a function of $\boldsymbol{X}$, $\boldsymbol{R}$, and $\boldsymbol{Z}$. We write this encrypting transformation as

$$\boldsymbol{Y} = E_{\boldsymbol{ZR}}(\boldsymbol{X}) \tag{2}$$

to emphasize that we wish to think of the cryptogram $\boldsymbol{Y}$ as some function of only the plaintext $\boldsymbol{X}$, the particular function being determined by the value of the secret key $\boldsymbol{Z}$ and of the randomizing sequence $\boldsymbol{R}$. As Fig. 1 implies, the decrypter must be able to invert this transformation without knowledge of the randomizing sequence. That is

$$\boldsymbol{X} = D_{\boldsymbol{Z}}(\boldsymbol{Y}) \tag{3}$$

which expresses the fact that the plaintext $\boldsymbol{X}$ must be some function of the cryptogram $\boldsymbol{Y}$ where the particular function is determined by the secret key $\boldsymbol{Z}$ alone. The enemy cryptanalyst observes the cryptogram $\boldsymbol{Y}$ and nothing else, and forms his estimate $\hat{\boldsymbol{X}}$ of the plaintext $\boldsymbol{X}$ and/or his estimate $\hat{\boldsymbol{Z}}$ of the secret key $\boldsymbol{Z}$. The enemy cryptanalyst, in accordance with Kerckhoff's precept, is assumed to know all details of the encrypter and decrypter, but of course to have no knowledge of $\mathbf{X}$, $\mathbf{R}$, and, in particular, of $\mathbf{Z}$.

Our Fig. 1 differs from the "Schematic of a general secrecy system" that appears as Fig. 1 in Shannon's 1949 paper [10] only in that we have included a "randomizer" in the enciphering process. Such randomization is an old trick of the cryptographer (and it also plays a special role in the theory of authenticity that we will soon consider). In English text, the letter e appears much more frequently than any other letter. If English text is first converted into text in some larger alphabet by replacing e each time with a randomly chosen letter from the large "e-group" of letters in the larger alphabet, and similarly replacing other frequently chosen English letters with random choices of a letter from appropriately-sized groups in the larger alphabet, one obtains a new text in which all letters of the larger alphabet have (approximately) the same frequency. Enciphering of this randomized text frustrates a single-letter frequency analysis by the enemy cryptanalyst. But, after deciphering the randomized text, the legitimate receiver can remove the randomization merely by replacing each letter in the e-group of the larger alphabet by the letter e, and so on—he does not need to be told in advance which random substitutions would be made. Such randomized ciphers are known as "multiple-substitution ciphers" and also as "homophonic ciphers." The great mathematician, Gauss, deceived himself into believing that, by using homophonic substitution, he had devised an unbreakable cipher [2]; but, without question, randomization is a useful cryptographic tool. As its inclusion hardly complicates Shannon's theory of secrecy, we have included it in our Fig. 1.

It is important to recognize that $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are random quantities. The statistics of the plaintext $\mathbf{X}$ are of course determined by the message source, but the statistics of the secret key $\mathbf{Z}$ and of the randomizing sequence $\mathbf{R}$ are under the control of the cryptographer. As Fig. 1 suggests, we shall always assume that $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{R}$ are statistically independent of one another.

## B. Theoretical and Practical Security

Shannon considered two very different notions of security for cryptographic systems. He first considered the question of *theoretical security*, by which he meant, "How secure is a system against cryptanalysis when the enemy has unlimited time and manpower available for the analysis of intercepted cryptograms?" [10, p. 658]. Shannon's theory of theoretical security, which we shall next review, casts much light into cryptography, but leads to the pessimistic conclusion that the amount of secret key needed to build a theoretically secure cipher will be impractically large for most applications. Thus, Shannon also treated the question of *practical security*, by which he meant: Is the system secure against a cryptanalyst who has a certain limited amount of time and computational power available for the analysis of intercepted cryptograms? Public-key systems, to be discussed in Section III, are intended to provide practical security—they cannot provide theoretical security.

## C. Perfect Secrecy

The first assumption in Shannon's theory of theoretical security is that the secret key will be used only one time, or equivalently that the $M$ digits of the plaintext $\mathbf{X}$ form the total of messages that will be enciphered before the secret key $\mathbf{Z}$ and the randomizer $\mathbf{R}$ are changed. The second assumption is that the enemy cryptanalyst has access only to the cryptogram $\mathbf{Y}$ and thus is limited to a ciphertext-only attack. Shannon then defined *perfect secrecy* to mean that the plaintext $\mathbf{X}$ is statistically independent of the cryptogram $\mathbf{Y}$, i.e., that $P(\mathbf{X} = \mathbf{x} \mid \mathbf{Y} = \mathbf{y}) = P(\mathbf{X} = \mathbf{x})$ for all possible particular plaintexts $\mathbf{x} = [x_1, x_2, \cdots, x_M]$ and particular cryptograms $\mathbf{y}$. This is the same as saying that the enemy cryptanalyst can do no better estimating $\mathbf{X}$ with knowledge of $\mathbf{Y}$ than he could do without knowing $\mathbf{Y}$, no matter how much computing time and power he has available for the processing of $\mathbf{Y}$. Having made the right mathematical formulation of the problem, it was then child's play for Shannon to show that perfect secrecy systems exist.

Consider the case of a nonrandomized cipher in which the plaintext, ciphertext, and key digits all take values in the $L$-ary alphabet $\{0, 1, \cdots, L-1\}$, and in which the length $K$ of the key and length $N$ of the cryptogram coincide with the length $M$ of the plaintext, i.e., $K = N = M$. Suppose that the key is chosen to be *completely random*, i.e., $P(\mathbf{Z} = \mathbf{z}) = L^{-M}$ for all $L^M$ possible values $\mathbf{z}$ of the secret key, and that the enciphering transformation is

$$Y_i = X_i \oplus Z_i, \qquad i = 1, 2, \cdots, M \tag{4}$$

where $\oplus$ denotes addition modulo $L$. Because for each possible choice $x_i$ and $y_i$ of $X_i$ and $Y_i$, respectively, there is a unique $z_i$ such that $Z_i = z_i$ satisfies (4), it follows that $P(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x}) = L^{-M}$ for every possible particular $\mathbf{y}$ and $\mathbf{x}$, no matter what the statistics of $\mathbf{X}$ may be. Thus $\mathbf{X}$ and $\mathbf{Y}$ are statistically independent, and hence this *modulo-L Vernam system* (to use Shannon's terminology) provides perfect secrecy. The modulo-$L$ Vernam system is better known under the name, the *one-time pad*, from its use shortly before, during and after World War II by spies of several nationalities who were given a pad of paper containing the randomly chosen secret key and told that it could be used for only one encipherment. There appears to have been a general belief in cryptological circles that this cipher was unbreakable, but Shannon seems to have been the first to publish a proof of this theoretical unbreakability.

It is worth noting here that the one-time pad offers perfect secrecy no matter what the statistics of the plaintext $\mathbf{X}$ may be. In fact, we will show shortly that it also uses the least possible amount of secret key for any cipher that provides perfect secrecy independent of the statistics of the plaintext—this is a most desirable attribute; one would not usually wish the security of the cipher system to depend on the statistical nature of the message source. But the fact that the one-time pad requires one digit of secret key for each digit of plaintext makes it impractical in all but the few cryptographic applications, such as encrypting the Moscow–Washington hotline, where the need for secrecy is paramount and the amount of plaintext is quite limited.

## D. Key Requirements for Perfect Secrecy

To go further in the study of theoretical security, we need to make use of some properties of "uncertainty" (or "entropy"), the fundamental quantity in Shannon's information theory [11]. Uncertainty is always defined as the mathematical expectation of the negative logarithm of a corresponding probability distribution. For instance,

$H(\mathbf{X} \mid \mathbf{Y})$ (which should be read as "the uncertainty about $\mathbf{X}$ given knowledge of $\mathbf{Y}$") is the expectation of the negative logarithm of $P(\mathbf{X} = x \mid \mathbf{Y} = y)$, i.e.,

$$H(\mathbf{X} \mid \mathbf{Y}) = \sum_x \sum_y P(\mathbf{X} = x, \mathbf{Y} = y)\,[-\log P(\mathbf{X} = x \mid \mathbf{Y} = y)]$$

where the sums are over all possible values $x$ and $y$ of the random quantities $\mathbf{X}$ and $\mathbf{Y}$, respectively. Uncertainties obey intuitively-pleasing rules, such as $H(\mathbf{X}, \mathbf{Y}) = H(\mathbf{X}) + H(\mathbf{Y} \mid \mathbf{X})$, which we will use in our discussion of theoretical secrecy without further justification—the reader is referred to [11] or to the introductory chapters of any standard textbook on information theory for proofs of the validity of these "obvious" manipulations of uncertainties.

Equations (2) and (3) above can be written equivalently in terms of uncertainties as

$$H(\mathbf{Y} \mid \mathbf{X}, \mathbf{Z}, \mathbf{R}) = 0 \tag{5}$$

and

$$H(\mathbf{X} \mid \mathbf{Y}, \mathbf{Z}) = 0 \tag{6}$$

respectively, because, for instance, $H(\mathbf{X} \mid \mathbf{Y}, \mathbf{Z})$ is 0 if and only if $\mathbf{Y}$ and $\mathbf{Z}$ together uniquely determine $\mathbf{X}$. The definition of perfect secrecy may be written as

$$H(\mathbf{X} \mid \mathbf{Y}) = H(\mathbf{X}) \tag{7}$$

since this equality holds if and only if $\mathbf{X}$ and $\mathbf{Y}$ are statistically independent.

For any secret-key cryptosystem, one has

$$\begin{aligned} H(\mathbf{X} \mid \mathbf{Y}) &\le H(\mathbf{X}, \mathbf{Z} \mid \mathbf{Y}) \\ &= H(\mathbf{Z} \mid \mathbf{Y}) + H(\mathbf{X} \mid \mathbf{Y}, \mathbf{Z}) \\ &= H(\mathbf{Z} \mid \mathbf{Y}) \\ &\le H(\mathbf{Z}) \end{aligned} \tag{8}$$

where we have made use of (6) and of the fact that the removal of given knowledge can only increase uncertainty. If the system gives perfect secrecy, it follows from (7) and (8) that

$$H(\mathbf{Z}) \ge H(\mathbf{X}). \tag{9}$$

Inequality (8) is *Shannon's fundamental bound for perfect secrecy; the uncertainty of the secret key must be at least as great as the uncertainty of the plaintext that it is concealing.* If the $K$ digits in the key are chosen from an alphabet of size $L_z$, then

$$H(\mathbf{Z}) \le \log (L_z^K) = K \log L_z \tag{10}$$

with equality if and only if the key is completely random. Similarly,

$$H(\mathbf{X}) \le M \log L_x \tag{11}$$

(where $L_x$ is the size of the plaintext alphabet) with equality if and only if the plaintext is completely random. Thus, if $L_x = L_z$ (as in the one-time pad) and if the plaintext is completely random, Shannon's bound (9) for perfect secrecy yields, with the aid of (10) and of equality in (11)

$$K \ge M. \tag{12}$$

That is, the key must be at least as long as the plaintext, a lower bound that holds with equality for the one-time pad.

### *E. Breaking an Imperfect Cipher*

Shannon also considered the question of when the enemy cryptanalyst would be able in theory to break an imperfect cipher. To this end, he introduced the *key equivocation function*

$$f(n) = H(\mathbf{Z} \mid Y_1, Y_2, \cdots, Y_n) \tag{13}$$

which measures the uncertainty that the enemy cryptanalyst has about the key given that he has examined the first $n$ digits of the cryptogram. Shannon then defined the *unicity distance* $u$ as the smallest $n$ such the $f(n) \approx 0$. Given $u$ digits of the ciphertext and not before, there will be essentially only one value of the secret key consistent with $Y_1, Y_2, \cdots, Y_n$, so it is precisely at this point that the enemy cryptanalyst with unlimited time and computing power could deduce the secret key and thus break the cipher. Shannon showed for a certain well-defined "random cipher" that

$$u \approx \frac{H(\mathbf{Z})}{r \log L_y} \tag{14}$$

where

$$r = 1 - \frac{H(\mathbf{X})}{N \log L_y} \tag{15}$$

is the *percentage redundancy* of the message information contained in the $N$ digit cryptogram, whose letters are from an alphabet of size $L_y$. When $N = M$ and $L_x = L_y$ (as is true in most cryptosystems), $r$ is just the percentage redundancy of the plaintext itself, which is about $\frac{3}{4}$ for typical English text. When $L_x = L_z$ and the key is chosen completely at random to maximize the unicity distance, (14) gives

$$u \approx \frac{K}{r}. \tag{16}$$

Thus, a cryptosystem with $L_x = L_y = L_z$ used to encipher typical English text can be broken after only about $N = \frac{4}{3}K$ ciphertext digits are received. For instance, a secret key of 56 bits (8 ASCII 7-bit symbols) can be found in principle from examination of only about 11 ASCII 7-bit symbols of ciphertext.

Although Shannon's derivation of (14) assumes a particular kind of "random" cipher, he remarked "that the random cipher analysis can be used to estimate equivocation characteristics and the unicity distance for the ordinary types of ciphers" [10, p. 698]. Wherever it has been possible to test this assertion of Shannon's, it has been found to be true. Shannon's approximation (14) is routinely used today to estimate the unicity distance of "ordinary" secret-key ciphers.

The reader may well be worrying about the validity of (14) and (16) when $r = 0$, as it would in the case when $N = M$, $L_x = L_y$, and the message source emitted completely random plaintext so that $H(\mathbf{X}) = M \log L_x = N \log L_y$. The answer is somewhat surprising: the enemy cryptanalyst can never break the system ($u = \infty$ is indeed the correct unicity distance!), even if $K \ll M$ so that (12) tells us that the system does not give perfect secrecy. The resolution of this paradox is that perfect secrecy demands that $\mathbf{Y}$ provide no information at all about $\mathbf{X}$, whereas breaking the system demands that $\mathbf{Y}$ determines $\mathbf{X}$ essentially uniquely, i.e., that $\mathbf{Y}$ must provide the maximum possible information about

$\boldsymbol{X}$. If the secret key $\boldsymbol{Z}$ were also chosen completely at random in the cipher for the completely-random message source described above, there would always be $L_z^K$ different plaintext-key pairs consistent with any possible cryptogram $\boldsymbol{y}$, and all would be equally likely alternatives to the hapless cryptanalyst. This suggests, as Shannon was quick to note, that *data compression is a useful cryptographic tool.* An ideal data compression algorithm transforms a message source into the completely-random (or "nonredundant") source that we have just been considering. Unfortunately, no one has yet devised a data compression scheme for realistic sources that is both ideal and practical (nor is anyone ever likely to do so), but even a nonideal scheme can be used to decrease $r$ significantly, and thus to increase the unicity distance $u$ significantly. Experience had long ago taught cryptographers that redundancy removal was a useful trick. In the days when messages were hand-processed, cryptographers would often delete from the plaintext many letters and blanks that could be recognized as missing and be replaced by the legitimate receiver. THSISASIMPLFORMOFDATACOMPRESION.

Shannon's derivation of (14) assumed a cryptographic system without the randomizing sequence $\boldsymbol{R}$ in our Fig. 1. When randomization is included in the cipher, $H(\boldsymbol{X})$ in (15) must be replaced by $H(\boldsymbol{X}, \boldsymbol{R})$ in order for (14) still to hold. This suggests that randomization can also be used to reduce the redundancy $r$ in the cryptogram. This, too, old-time cryptographers had learned from experience. They frequently inserted extra symbols into the plaintext, often an $X$, to hide the real statistics of the message. THXISISAXNEXAMXPLE.

## F. Authenticity and Deception

We have several times mentioned that cryptography seeks to ensure the secrecy and/or authenticity of messages. But it is in fact quite a recent realization that secrecy and authenticity are independent attributes. If one receives a cryptogram that decrypts under the actual secret key to a sensible message, cannot one be sure that this cryptogram was sent by one's friend who is the only other person privy to this secret key? The answer, as we shall see, in general is: No! The systematic study of authenticity is the work of G. J. Simmons [14], who has developed a theory of authenticity that in many respects is analogous to Shannon's theory of secrecy.

To treat the theoretical security of authenticity systems as formulated by Simmons, we must give the enemy cryptanalyst more freedom than he is allowed in the model of Fig. 1. Fig. 2 shows the necessary modification to Fig. 1. The enemy cryptanalyst is now the one who originates the "fraudulent" cryptogram $\tilde{\boldsymbol{Y}}$ that goes to the decrypter. The line from the decrypter to the destination is shown dotted in Fig. 2 to suggest that the decrypter might recognize $\tilde{\boldsymbol{Y}}$ as fraudulent and thus not be deceived into passing a fraudulent plaintext $\tilde{\boldsymbol{X}}$ to the destination.

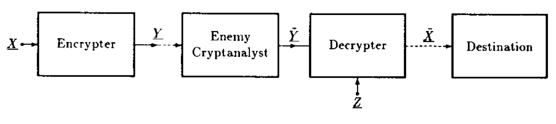

**Fig. 2.** Modifications to Fig. 1 for consideration of authenticity attacks.

As did Shannon, Simmons assumes that the secret key $\boldsymbol{Z}$ will be used only one time, i.e., to form only one authentic cryptogram $\boldsymbol{Y}$. But Simmons recognized that even in this case the enemy cryptanalyst can choose either of two quite different attacks. He can choose to form his fraudulent cryptogram $\tilde{\boldsymbol{Y}}$ without waiting to see the authentic cryptogram $\boldsymbol{Y}$ (the *impersonation attack*), which explains why the input line to the enemy cryptanalyst in Fig. 2 is shown dotted. The enemy cryptanalyst is said to succeed in this impersonation attack if the decrypter accepts $\tilde{\boldsymbol{Y}}$ as authentic (even if it happens that $\tilde{\boldsymbol{Y}}$ turns out later to coincide with the authentic cryptogram $\boldsymbol{Y}$). The enemy cryptanalyst can also wait to see $\boldsymbol{Y}$ before he forms $\tilde{\boldsymbol{Y}}$ (the *substitution attack*), in which case he is said to succeed if the decrypter accepts $\tilde{\boldsymbol{Y}}$ as authentic *and* $\tilde{\boldsymbol{Y}}$ decrypts to $\tilde{\boldsymbol{X}}$ with $\tilde{\boldsymbol{X}} \neq \boldsymbol{X}$. Let $P_I$ and $P_S$ denote the enemy cryptanalyst's best-possible probability of success in an impersonation attack or in a substitution attack, respectively. Assuming that the enemy cryptanalyst will choose the attack that is more likely to succeed, Simmons defines $P_d = \max(P_I, P_S)$ to be the *probability of deception*, i.e., the probability that the enemy cryptanalyst succeeds in defrauding the decrypter.

The theory of authenticity is in many ways more subtle than the corresponding theory of secrecy. In particular, it is not at all obvious how "perfect authenticity" should be defined. Let $\#\{\boldsymbol{Y}\}$ denote the number of cryptograms $\boldsymbol{y}$ such that $P(\boldsymbol{Y} = \boldsymbol{y}) \neq 0$, and let $\#\{\boldsymbol{X}\}$ and $\#\{\boldsymbol{Z}\}$ be similarly defined as the number of plaintexts and cryptograms, respectively, with nonzero probability. It follows from (3) that, for every $\boldsymbol{z}$, there must be at least $\#\{\boldsymbol{X}\}$ different cryptograms $\boldsymbol{y}$ such that $P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{Z} = \boldsymbol{z}) \neq 0$. Hence, if the enemy cryptanalyst in an impersonation attack selects $\boldsymbol{Y}$ completely at random from the $\#\{\boldsymbol{Y}\}$ cryptograms with nonzero probability, his probability of success will be at least $\#\{\boldsymbol{X}\}/\#\{\boldsymbol{Y}\}$. Thus, $P_I$, the probability of success in an optimal impersonation attack satisfies

$$P_I \geq \#\{\boldsymbol{X}\}/\#\{\boldsymbol{Y}\}. \tag{17}$$

This shows that good protection against an impersonation attack demands that $\#\{\boldsymbol{Y}\}$ be much greater than $\#\{\boldsymbol{X}\}$, and shows also that *complete protection* (i.e., $P_I = 0$) *is impossible*. We note further that (17) can hold with equality only when there are exactly $\#\{\boldsymbol{X}\}$ valid cryptograms $\boldsymbol{y}$ for each key $\boldsymbol{z}$, which means that a randomized cipher cannot achieve equality in (17).

Because complete protection against deception is impossible, the only recourse is to define "perfect authenticity" to mean as much protection against deception as is possible given the size of the space of valid cryptograms (even if this means that we must call a system "perfect" for which $\#\{\boldsymbol{Y}\} = \#\{\boldsymbol{X}\}$ and hence $P_d = 1$). This is what Simmons has done, but we must develop the theory a little further before introducing his precise definition of "perfect authenticity."

Let the *authentication function*, $\phi(\boldsymbol{y}, \boldsymbol{z})$ be defined to be 1 if $\boldsymbol{y}$ is a valid cryptogram for the secret key $\boldsymbol{z}$ and to be 0, otherwise. Note that if $\boldsymbol{Z} = \boldsymbol{z}$, the decrypter will accept $\tilde{\boldsymbol{Y}} = \boldsymbol{y}$ as a valid cryptogram just when $\phi(\boldsymbol{y}, \boldsymbol{z}) = 1$. To simplify notation, we now write $P(\boldsymbol{y})$, $P(\boldsymbol{z})$ and $P(\boldsymbol{y}|\boldsymbol{z})$ as shorthand for $P(\boldsymbol{Y} = \boldsymbol{y})$, $P(\boldsymbol{Z} = \boldsymbol{z})$, and $P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{Z} = \boldsymbol{z})$, respectively. The probability that a particular $\boldsymbol{y}$ is a valid cryptogram can then be written

$$P(\boldsymbol{y} \text{ valid}) = \sum_{\boldsymbol{z}} \phi(\boldsymbol{y}, \boldsymbol{z}) P(\boldsymbol{z}) \tag{18}$$

which is just the total probability of the keys $\boldsymbol{z}$ for which $y$ is a valid cryptogram. The best impersonation attack is for the enemy cryptanalyst to choose $\tilde{Y} = y$ for that $y$ that maximizes $P(y \text{ valid})$. Thus

$$P_I = \max_y P(y \text{ valid}). \qquad (19)$$

Noting that $P(Y = y, \boldsymbol{Z} = \boldsymbol{z}) = P(\boldsymbol{z})\, P(y|\boldsymbol{z}) \neq 0$ implies that $\phi(y, \boldsymbol{z}) = 1$, we can write

$$P(y) = \sum_{\boldsymbol{z}} P(\boldsymbol{z})\, P(y|\boldsymbol{z})\, \phi(y, \boldsymbol{z}). \qquad (20)$$

We are now in position to derive Simmons's lower bound on $P_I$. The first and least obvious step is to define

$$Q_y(\boldsymbol{z}) = P(\boldsymbol{z})\, \phi(y, \boldsymbol{z})/P(y \text{ valid}). \qquad (21)$$

This is a nonnegative function of $\boldsymbol{z}$ which, because of (18), sums over $\boldsymbol{z}$ to 1, i.e., $Q_y(\boldsymbol{z})$ can be considered as a probability distribution over $\boldsymbol{z}$. Then, because $t \log t$ is a strictly convex function for $t \geq 0$, we can use Jensen's inequality, cf. [15, pp. 151–152], to obtain

$$\begin{aligned} &\sum_{\boldsymbol{z}} Q_y(\boldsymbol{z})\, P(y|\boldsymbol{z}) \log P(y|\boldsymbol{z}) \\ &\quad \geq \left[\sum_{\boldsymbol{z}} Q_y(\boldsymbol{z})\, P(y|\boldsymbol{z})\right] \log \left[\sum_{\boldsymbol{z}} Q_y(\boldsymbol{z})\, P(y|\boldsymbol{z})\right] \end{aligned} \qquad (22)$$

with equality if and only if $P(y|\boldsymbol{z})$ has the same value for all $\boldsymbol{z}$ for which $Q_y(\boldsymbol{z}) \neq 0$, i.e., for which $\phi(y, \boldsymbol{z}) \neq 0$. From (20) and (21), we have

$$P(y) = P(y \text{ valid}) \sum_{\boldsymbol{z}} Q_y(\boldsymbol{z})\, P(y|\boldsymbol{z}) \qquad (23)$$

so that

$$\begin{aligned} P(y) \log P(y) &= P(y) \log P(y \text{ valid}) \\ &\quad + P(y) \log \sum_{\boldsymbol{z}} Q_y(\boldsymbol{z})\, P(y|\boldsymbol{z}). \end{aligned} \qquad (24)$$

But the second term on the right of (24) with the aid of (23) becomes

$$\begin{aligned} &P(y \text{ valid}) \left[\sum_{\boldsymbol{z}} Q_y(\boldsymbol{z})\, P(y|\boldsymbol{z})\right] \log \left[\sum_{\boldsymbol{z}} Q_y(\boldsymbol{z})\, P(y|\boldsymbol{z})\right] \\ &\quad \leq P(y \text{ valid}) \sum_{\boldsymbol{z}} Q_y(\boldsymbol{z})\, P(y|\boldsymbol{z}) \log P(y|\boldsymbol{z}) \\ &\quad = \sum_{\boldsymbol{z}} P(\boldsymbol{z})\, P(y|\boldsymbol{z})\, \phi(y, \boldsymbol{z}) \log P(y|\boldsymbol{z}) \\ &\quad = \sum_{\boldsymbol{z}} P(\boldsymbol{z})\, P(y|\boldsymbol{z}) \log P(y|\boldsymbol{z}) \end{aligned} \qquad (25)$$

where we have used (22) to obtain the inequality, and then made use of (21) and of the fact that $P(y|\boldsymbol{z})\, P(\boldsymbol{z}) \neq 0$ implies $\phi(y, \boldsymbol{z}) = 1$. Using (25) in (24) and summing over $y$ gives

$$-H(Y) \leq \sum_y P(y) \log P(y \text{ valid}) - H(Y|\boldsymbol{Z}). \qquad (26)$$

In information theory, the quantity $H(Y) - H(Y|\boldsymbol{Z})$ is called the (average) *mutual information* between $\boldsymbol{Y}$ and $\boldsymbol{Z}$ and is denoted $I(\boldsymbol{Y}; \boldsymbol{Z})$. Thus (26) can equivalently be written as

$$\sum_y P(y) \log P(y \text{ valid}) \geq -I(Y; \boldsymbol{Z}). \qquad (27)$$

The final step is to note that the "average" of $\log P(y \text{ valid})$, which appears on the left of (27), is less than or equal to the maximum of $\log P(y \text{ valid})$, so that (19) together with (27) yields

$$\log P_I \geq -I(\boldsymbol{Y}; \boldsymbol{Z}) \qquad (28)$$

which is Simmons's bound. The somewhat lengthy derivation, which nonetheless is a shortening of that in [14], permits one to see that equality holds in (28) if and only if *both* (i) $P(y \text{ valid})$ is independent of $y$ so that the left sides of (27) and (28) are equal (or, equivalently, so that choosing $\boldsymbol{Y}$ completely at random is an optimum impersonation attack) *and* (ii) for each cryptogram $y$, $P(y|\boldsymbol{z})$ has the same value for all $\boldsymbol{z}$ for which $\phi(y, \boldsymbol{z}) = 1$. Because of the definition of $P_d$ as

$$P_d = \max(P_I, P_S) \qquad (29)$$

the bound (28) implies the further bound

$$\log P_d \geq -I(\boldsymbol{Y}; \boldsymbol{Z}); \qquad (30)$$

moreover, conditions (i) and (ii) are necessary (but in general no longer sufficient) conditions for equality to hold in (30).

Simmons [14] has defined *perfect authenticity* to mean that equality holds in (30). Even with perfect authenticity, however, it must be remembered that the probability of deception $P_d$ will be small only when $I(\boldsymbol{Y}; \boldsymbol{Z})$ is large, i.e., only when the cryptogram provides the enemy cryptanalyst with much information about the key! The information that $\boldsymbol{Y}$ gives about $\boldsymbol{Z}$ is a measure of how much of the secret key is used to provide authenticity.

The theory of the theoretical security of authenticity systems is less well developed than is that of secrecy systems. In particular, it is not known in general under what conditions systems offering perfect authenticity exist, although constructions of particular such systems have been given. Thus, we will content ourselves here with giving a series of simple examples that illuminate the main ideas of authentication theory and show the relation between authentication and secrecy.

In the following examples, the plaintext is always a single binary digit $X$, the cryptogram $\boldsymbol{Y} = [Y_1, Y_2]$ is a binary sequence of length 2, the key $\boldsymbol{Z} = [Z_1, \cdots, Z_K]$ is a completely-random binary sequence so that $P(\boldsymbol{Z} = \boldsymbol{z}) = 2^{-K}$ for all $\boldsymbol{z}$, and all logarithms are taken to the base 2 so that $H(\boldsymbol{Z}) = K$ bits.

*Example 1:* Consider the encipherment scheme with a key of length $K = 1$ described by the following table.

| $Z \backslash X$ | 0 | 1 |
|---|---|---|
| 0 | 00 | 10 |
| 1 | 01 | 11 |

The meaning is that, for instance, $Y = [1, 0]$ when $X = 1$ and $Z = 0$. The enciphering rule is simply $\boldsymbol{Y} = [X, Z]$, i.e., the key is appended as a "signature" to the plaintext to form the cryptogram. Thus, this system provides *no secrecy* at all. Moreover, $H(Z|\boldsymbol{Y}) = 0$ so that $I(\boldsymbol{Y}; Z) = 1$ bit, and the bound (28) becomes $P_I \geq \frac{1}{2}$. But $P(y \text{ valid}) = \frac{1}{2}$ for all $y$ so that in fact $P_I = \frac{1}{2}$, which is as small as possible. But upon observing $\boldsymbol{Y} = y$, the enemy cryptanalyst always knows the other valid cryptogram so that he can always succeed in a substitution attack. Hence $P_S = 1 = P_d > 2^{-I(Y;Z)} = \frac{1}{2}$, i.e., the authenticity is not perfect.

*Example 2:* Consider the randomized encipherment system.

| $z$ | $r$ \ $x$ | 0 | 1 |
|---|---|---|---|
| 0 | 0 | 00 | 10 |
| 0 | 1 | 01 | 11 |
| 1 | 0 | 00 | 11 |
| 1 | 1 | 01 | 10 |

Note that $Y_1 = X$ so that again there is *no secrecy*. Given $\boldsymbol{Y} = \boldsymbol{y}$ for any $\boldsymbol{y}$, the two possible values of $Z$ are equally likely so that $H(Z \mid \boldsymbol{Y}) = 1$, and thus $I(\boldsymbol{Y}; Z) = 0$. It follows then from (28) that this system must have $P_I = 1 = P_d = 2^{-I(Y;Z)}$ and thus trivially provides perfect authenticity. But, upon observing, say, $\boldsymbol{Y} = [0, 0]$, the enemy cryptanalyst is faced with two equally likely alternatives, [1, 0] and [1, 1], for the other valid cryptogram, only one of which will be accepted by the receiver, who knows $Z$, as authentic. Thus $P_S = \frac{1}{2}$. This example shows that a randomized cipher can satisfy (30) with equality, and also that $-I(\boldsymbol{Y}; \boldsymbol{Z})$ is *not* in general a lower bound on $\log P_S$.

Examples 1 and 2 show that *the substitution attack can be stronger than the impersonation attack, and vice versa.*

*Example 3:* Consider the same system as in Example 2 except that $z$ and $r$ are now the two digits $z_1$ and $z_2$, respectively, of the secret key, and hence both are known to the legitimate receiver. There is still *no secrecy* because $Y_1 = X$. Given $\boldsymbol{Y} = \boldsymbol{y}$ for any $\boldsymbol{y}$, there are still two equally likely possibilities for $\boldsymbol{Z}$ so that $H(\boldsymbol{Z} \mid \boldsymbol{Y}) = 1$ and hence $I(\boldsymbol{Y}; \boldsymbol{Z}) = 1$ bit. But $P(\boldsymbol{y} \text{ valid}) = \frac{1}{2}$ for all four cryptograms $\boldsymbol{y}$ and thus $P_I = \frac{1}{2}$. Moreover, given that he observes $\boldsymbol{Y} = \boldsymbol{y}$, the enemy cryptanalyst is faced with two equally likely choices for the other valid cryptogram so that $P_S = \frac{1}{2}$. Thus $P_d = \frac{1}{2} = 2^{-I(Y;Z)}$ and hence this system offers (nontrivial) *perfect authenticity*, no matter what the statistics of the plaintext $X$ may be.

*Example 4:* Consider the encipherment system.

| $z_1$ | $z_2$ \ $x$ | 0 | 1 |
|---|---|---|---|
| 0 | 0 | 00 | 11 |
| 0 | 1 | 01 | 10 |
| 1 | 0 | 10 | 01 |
| 1 | 1 | 11 | 00 |

Because $P(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{X} = \boldsymbol{x}) = \frac{1}{4}$ for all $\boldsymbol{x}$ and $\boldsymbol{y}$, the system provides *perfect secrecy*. By the now familiar arguments, $I(\boldsymbol{Y}; \boldsymbol{Z}) = 1$ and $P_I = \frac{1}{2}$, the corresponding best possible protection against impersonation. But, upon observing $\boldsymbol{Y} = \boldsymbol{y}$, the enemy can always succeed in a substitution attack by choosing $\boldsymbol{Y}$ to be the complement of $\boldsymbol{y}$. Thus $P_S = 1 = P_d$ and hence this system provides *no protection against deception* by substitution.

*Example 5:* Consider the encipherment system.

| $z_1$ | $z_2$ \ $x$ | 0 | 1 |
|---|---|---|---|
| 0 | 0 | 00 | 10 |
| 0 | 1 | 01 | 00 |
| 1 | 0 | 11 | 01 |
| 1 | 1 | 10 | 11 |

This cipher provides *perfect secrecy* and has $I(\boldsymbol{Y}; \boldsymbol{Z}) = 1$ bit. Moreover, $P(\boldsymbol{y} \text{ valid}) = \frac{1}{2}$ for all $\boldsymbol{y}$ so that $P_I = \frac{1}{2}$. Upon observing that $\boldsymbol{Y} = \boldsymbol{y}$, say $\boldsymbol{y} = [0, 0]$, the enemy cryptanalyst is faced with the two alternatives [1, 0] and [0, 1] for the other valid cryptogram with the probabilities $P(X = 0)$ and $P(X = 1)$, respectively. Thus, $P_S \geq \frac{1}{2}$ with equality if and only if $P(X = 0) = \frac{1}{2}$. It follows that $P_d = P_S \geq 2^{-I(Y;Z)} = \frac{1}{2}$ with equality if and only if $P(X = 0) = \frac{1}{2}$. Thus, if $P(X = 0) = \frac{1}{2}$, this cipher also provides *perfect authenticity*. (The reader should have no trouble adding a third bit to the key to obtain a cipher that provides both perfect secrecy and perfect authenticity, independent of the plaintext statistics.)

Examples 3, 4, and 5 illustrate the fact that *secrecy and authenticity are independent attributes of a cryptographic system*—a lesson that is too often forgotten in practice.

### G. Practical Security

In Section II-E, we noted the possibility for a cipher system with a limited key [i.e., with $K \ll H(\boldsymbol{X})$] to have an infinite unicity distance and hence to be theoretically "unbreakable." Shannon called such ciphers *ideal*, but noted that their design poses virtually insurmountable practical problems [10, p. 700]. Most practical ciphers must depend for their security not on the theoretical impossibility of their being broken, but on the practical difficulty of such breaking. Indeed, Shannon postulated that every cipher has a *work characteristic* $W(n)$ which can be defined as the average amount of work (measured in some convenient units such as hours of computing time on a CRAY 2) required to find the key when given $n$ digits of the ciphertext. Shannon was thinking here of a ciphertext-only attack, but a similar definition can be made for any form of cryptanalytic attack. The quantity of greatest interest is the limit of $W(n)$ as $n$ approaches infinity, which we shall denote by $W(\infty)$ and which can be considered the average work needed to "break the cipher." Implicit in the definition of $W(n)$ is that the *best possible cryptanalytic algorithm* is employed to break the cipher. Thus to compute or underbound $W(n)$ for a given cipher, we are faced with the extremely difficult task of finding the best possible way to break that cipher, or at least of proving lower bounds on the work required in the best possible attack. There is no practical cipher known today (at least to researchers outside the secret research community) for which even an interesting lower bound on $W(\infty)$ is known. Such practical ciphers are generally evaluated in terms of what one might call the *historical work characteristic*, $W_h(n)$, which can be defined as the average amount of work to find the key from $n$ digits of ciphertext when one uses the *best of known attacks* on the cipher. When one reads about a "cipher that requires millions of years to break," one can be sure that the writer is talking about $W_h(\infty)$. When calculated by a cryptographer who is fully acquainted with the techniques of cryptanalysis, $W_h(\infty)$ can be a trustworthy measure of the real security of the cipher, particularly if the cryptographer includes a judicious "margin of error" in his calculations. But there always lurks the danger that $W(\infty) \ll W_h(\infty)$, and hence that an enemy cryptanalyst might devise a new and totally unexpected attack that will, when it is ultimately revealed, greatly reduce $W_h(\infty)$—the history of cryptography is rife with examples!

### H. Diffusion and Confusion

Shannon suggested two general principles, which he called diffusion and confusion [10, p. 708], to guide the design of practical ciphers. By *diffusion*, he meant the spreading out of the influence of a single plaintext digit over many ciphertext digits so as to hide the statistical structure of the plaintext. An extension of this idea is to spread the influence of a single key digit over many digits of ciphertext so as to frustrate a piecemeal attack on the key. By *confusion*, Shannon meant the use of enciphering transformations that complicate the determination of how the statistics of the ciphertext depend on the statistics of the plaintext. But a cipher should not only be difficult to break, it must also be easy to encipher and decipher when one knows the secret key. Thus, a very common approach to creating diffusion and confusion is to use a *product cipher*, i.e., a cipher that can be implemented as a succession of simple ciphers, each of which adds its modest share to the overall large amount of diffusion and confusion.

Product ciphers most often employ both transposition ciphers and substitution ciphers as the component simple ciphers. A *transposition cipher* merely permutes the letters in the plaintext, the particular permutation being determined by the secret key. For instance, a transposition cipher acting on six-letter blocks of latin letters might cause CAESAR to encipher to AESRAC. The single-letter statistics of the ciphertext are the same as for the plaintext, but the higher-order statistics of the plaintext are altered in a confusing way. A *substitution cipher* merely replaces each plaintext letter with another letter from the same alphabet, the particular substitution rule being determined by the secret key. The single-letter statistics of the ciphertext are the same as for the plaintext. The Caesar cipher discussed in Section I-E is a simple substitution cipher with only 26 possible values of the secret key. But if the substitution is made on a very large alphabet so that it is not likely that any plaintext letter will occur more than once in the lifetime of the secret key, then the statistics of the plaintext are of little use to the enemy cryptanalyst and a substitution cipher becomes quite attractive. To achieve this condition, the cryptographer can choose the "single letters" upon which the substitution is applied to be groups of several letters from the original plaintext alphabet. For instance, a substitution upon pairs of Latin letters, in which CA was replaced by WK, ES by LB, and AR by UT, would result in CAESAR being enciphered to WKLBUT. If this ciphertext was then further enciphered by the above-considered transposition cipher, the resulting ciphertext would be KLBTUW. Such interleaving of simple transpositions and substitutions, when performed many times, can yield a very strong cipher, i.e., one with very good diffusion and confusion.

## *I. The Data Encryption Standard*

Perhaps the best example of a cipher designed in accordance with Shannon's diffusion and confusion principles is the Data Encryption Standard (DES). In the DES, the plaintext $\boldsymbol{X}$, the cryptogram $\boldsymbol{Y}$ and the key $\boldsymbol{Z}$ are binary sequences with lengths $M = 64$, $N = 64$, and $K = 56$, respectively. All $2^{64}$ possible values of $\boldsymbol{X}$ are, in general, allowed. Because $M = N = 64$, this means that DES is in fact a substitution cipher, albeit on a very large alphabet of $2^{64} \approx 10^{19}$ "letters"! In its so-called *electronic code book mode*, successive 64 bit "blocks" of plaintext are enciphered using the same secret key, but otherwise independently. Any cipher used in this manner is called a *block cipher*.

The DES is a product cipher that employs 16 "rounds" of successive encipherment, each round consisting of rather simple transpositions and substitutions on 4 bit groups. Only 48 key bits are used to control each round, but these are selected in a random-appearing way for successive rounds from the full 56 bit key. We shall not pursue further details of the DES here; a good short description of the DES algorithm appears in [1] and the complete description is readily available [16]. It suffices here to note that it appears hopeless to give a useful description of how a single plaintext bit (or a single key bit) affects the ciphertext (good diffusion!), or of how the statistics of the plaintext affect those of the ciphertext (good confusion!).

The DES algorithm was submitted by IBM in 1974 in response to the second of two public invitations by the U.S. National Bureau of Standards (NBS) for designers to submit algorithms that might be used as a standard for data encryption by government and private entities. One design requirement was that the algorithm could be made public without compromising its security—a requirement that Kerckhoff would have admired! The IBM design was a modification of the company's older Lucifer cipher that used a 128 bit key. The original design submitted by IBM permitted all $16 \times 48 = 768$ bits of key used in the 16 rounds to be selected independently. A U.S. Senate Select Committee ascertained in 1977 that the U.S. National Security Agency (NSA) was instrumental in reducing the DES secret key to 56 bits that are each used many times, although this had previously been denied by IBM and NBS [17]. NSA also classified the design principles that IBM had used to select the particular substitutions that are used within the DES algorithm. But the entire algorithm in full detail was published by NBS in 1977 as a U.S. Federal Information Processing Standard [16], to become effective in July of that year.

Almost from the beginning, the DES was embroiled in controvery. W. Diffie and M. E. Hellman, cryptologic researchers at Stanford University, led a chorus of skepticism over the security of the DES that focused on the smallness of the secret key. With $2^{56} \approx 10^{17}$ possible keys, a *brute-force attack* or "exhaustive cryptanalysis" (in which the cryptanalyst tries one key after another until the cryptogram deciphers to sensible plaintext) on the DES was beyond feasibility, but only barely so. Diffie and Hellman published the conceptual blueprint for a highly-parallel special-purpose computer that, by their reckoning, would cost about 20 million dollars and would break DES cryptograms by essentially brute-force in about 12 hours [18]; Hellman later proposed a variant machine, that, by his reckoning, would cost only 4 million dollars and, after a year of initial computation, would break 100 cryptograms in parallel each day [19]. Counter-critics have attacked both of these proposals as wildy optimistic. But the hornet's nest of public adverse criticism of DES did lead the NBS to hold workshops of experts in 1976 and 1977 to "answer the criticisms" [17] and did give rise to the Senate hearing mentioned above. The general consensus of the workshops seems to have been that DES would be safe from a Diffie-Hellman-style attack for only about ten years, but that the 56 bit key provided no margin of safety [17]. Ten years have now passed, and the DES appears to have justified the faith of its defenders. Despite intensive scrutiny of the DES algorithm by cryptologic researchers, no one has yet publicly revealed any weakness of DES that could be exploited in an attack that would be significantly better than exhaustive cryptanalysis. The general consensus of cryptologic researchers today is that DES is an extremely good cipher with an unfortunately small key. But it should not be forgotten that the effective size of the secret key can be increased by using multiple DES encryptions with different keys, i.e., by making a product cipher with DES used for the component ciphers. At least three encryptions should be used to foil the "meet-in-the-middle attack" proposed by Merkle and Hellman [20].

## *J. Stream Ciphers*

In a block cipher, a plaintext block identical to a previous such block would give rise to the identical ciphertext block as well. This is avoided in the so-called *stream ciphers* in which the enciphering transformation on a plaintext "unit" changes from unit to unit. For instance, in the *cipher-block chaining* (CBC) mode proposed for the DES algorithm [16], the current 64 bit plaintext block is added bit-by-bit modulo-two to the previous 64 bit ciphertext block to produce the 64 bit block that is then enciphered with the DES algorithm to produce the current ciphertext block. Cipher-block-chaining converts a block cipher into a stream cipher with the advantage that tampering with ciphertext blocks is more readily detected, i.e., impersonation or substitution attacks become much more difficult. But cryptographers generally reserve the term "stream cipher" for use only in the case when the plaintext "units" are very small, say a single Latin letter or a single bit.

The most popular stream ciphers today are what can be called *binary additive stream ciphers*. In such a cipher, the $K$ bit secret key $\boldsymbol{Z}$, is used only to control a *running key generator* (RKG) that emits a binary sequence, $Z'_1, Z'_2, \cdots, Z'_N$, called the *running key*, where in general $N \gg K$. The ciphertext digits are then formed from the binary plaintext digits by simple modulo-two addition in the manner

$$Y_n = X_n \oplus Z'_n, \qquad n = 1, 2, \cdots N. \qquad (31)$$

Because modulo-two addition and substraction coincide, (31) implies

$$X_n = Y_n \oplus Z'_n, \qquad n = 1, 2, \cdots, N \qquad (32)$$

which shows that encryption and decryption can be performed by identical devices. A single plaintext bit affects only a single ciphertext bit, which is the worst possible diffusion; but each secret key bit can influence many ciphertext bits so the key diffusion can be good.

There is an obvious similarity between the binary additive stream cipher and a binary one-time pad. In fact, if $Z_n = Z'_n$ (i.e., if the secret key is used as the running key), then the additive stream cipher is identical to the one-time pad. This similarity undoubtedly accounts in part for the widespread faith in additive stream ciphers that one encounters in many cryptographers and in many users of ciphers. But, of course, in practical stream ciphers, the ciphertext length $N$ greatly exceeds the secret key length $K$. The best that one can then hope to do is to build an RKG whose output sequence cannot be distinguished by a resource-limited cryptanalyst from a completely random binary sequence. The trick is to build the RKG in such a way that, upon observing $Z'_1, Z'_2, \cdots, Z'_n$, the resource-limited cryptanalyst can do no better than to guess $Z'_{n+1}$ at random. If this can be done, one has a cipher that is secure against even a chosen-plaintext attack (by which one would mean that the enemy cryptanalyst could freely select, say the first $n$ bits of the plaintext sequence).

Stream ciphers have the advantage over block ciphers in that analytic measures of their quality are more easily formulated. For instance, stream cipher designers are greatly concerned with the *linear complexity* or "linear span" of the running-key sequence, which is defined as the length $L$ of the shortest linear-feedback shift-register (LFSR) that could produce the sequence. Fig. 3 shows a typical LFSR of length 6. The reason for this concern is that there is a simple algorithm that would quickly find this shortest LFSR after examining only the first $2L$ bits of the running key [21]. Thus, large linear complexity of the running-key sequence is a necessary (but far from sufficient) condition for the practical security of an additive stream cipher. (An up-to-date treatment of linear complexity in connection with stream ciphers may be found in the book by Rueppel [44].) The RKG of an additive stream cipher is often built by the nonlinear combining of the output sequences of several LFSRs, as such combining can create a sequence with large linear complexity. There arises then the danger that individual LFSR sequences will be correlated with the running-key sequence so that the enemy cryptanalyst can attack the cipher piecemeal. Siegenthaler [22] has shown recently that the "correction-immunity" of nonlinear combining functions can be precisely quantified and that the designer has to make an explicit tradeoff between correlation-immunity and linear complexity. There are many other known analytic approaches to stream cipher design. Taken together, they still leave one far from the point where one could say that the true work characteristic of a practical stream cipher is known, but they tend to give many cryptographers and users (perhaps misleadingly) greater trust in the historical work characteristics computed for stream ciphers than in those computed for block ciphers.

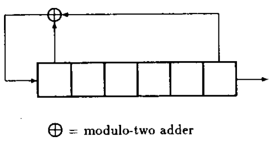

**Fig. 3.** A "typical" linear-feedback shift-register.

## *K. Provably-Secure Ciphers?*

When dealing with the practical security of ciphers, "It is difficult to define the pertinent ideas involved with sufficient precision to obtain results in the form of mathematical theorems," as Shannon said nearly 40 years ago [10, p. 702] in an eloquent understatement that needs no alteration today. It is an open question whether it is even possible to compute the true work characteristic $W(n)$ or its asymptotic value $W(\infty)$. A slender ray of hope lies in a totally impractical cipher proposed by this writer and I. Ingemarsson [23]. This cipher is a randomized stream cipher with a secret key of $K$ bits. One can prove that $W(\infty) \approx 2^{K/2}$ where the unit of computation is a binary test, i.e., a test with 2 outcomes. The "catch" is that the legitimate receiver must wait (during which time he does no testing or other computational work) until about $2^K$ bits have arrived before he begins to decipher. One can easily guarantee that the enemy cryptanalyst will need thousands of years to break the cipher, if one is willing to wait millions of years to read the plaintext! Such a cipher would be tolerable perhaps only to Rip van Winkle, the lazy and sleep-prone hero of Washington Irving's delightful short story, after whom both the story and the cipher have been named. Randomization, which was the feature that allowed the calculation of $W(\infty)$ for the impractical Rip van Winkle cipher, may turn out to be useful in developing a practical provably-secure cipher, if in fact this can be done at all.

## III. Public-Key Cryptography

### A. One-Way Functions

That the publication of Shannon's 1949 paper [10] resulted in no discernible upsurge in open cryptological research is due to several factors. First, the theory of theoretical security of secrecy systems that it provided was virtually complete in itself, and showed conclusively that theoretically-secure secrecy systems demand the secure transfer of far more secret key than is generally practicable. Moreover, the insights that Shannon provided into the practical security of secrecy systems tended to reinforce accepted cryptographic approaches rather than to suggest new ones. But Shannon's observation that "The problem of good cipher design is essentially one of finding difficult problems, subject to certain other conditions.... We may construct our cipher in such a way that breaking it is equivalent to (or requires at some point in the process) the solution of some problems known to be laborious" [10, p. 704] took root in the fertile imaginations of the Stanford cryptologic researchers, W. Diffie and M. E. Hellman. The fruit was their 1976 paper, "New Directions in Cryptography," [12] that stunned the cryptologic world with the startling news that *practically-secure secrecy systems can be built that require no secure transfer of any secret key whatsoever.*

The crucial contribution of the Diffie-Hellman paper lies in two unusually subtle definitions, that of a "one-way function," which was borrowed from work by R. M. Needham on secure computer-login techniques, and that of a "trap-door one-way function," which was totally new. A *one-way function* is defined as a function $f$ such that for every $x$ in the domain of $f$, $f(x)$ is easy to compute; but for virtually all $y$ in the range of $f$, it is computationally infeasible to find an $x$ such that $y = f(x)$. The first thing to note is that this is not a precise mathematical definition. What do "easy," "virtually all" (which we have substituted for Diffie and Hellman's "almost all," as the latter can have a precise mathematical meaning that was not intended in the definition), and "computationally infeasible" mean precisely? Yet the definition is sufficiently precise that one has no doubt as to what Diffie and Hellman essentially meant by a one-way function, and one has the feeling that it could be made completely precise in a particular context. It is less clear how such a function could be of use cryptographically—to build a cipher that not even the legitimate receiver could decipher seems the obvious (and worthless) application! A *trap-door one-way* function is defined as a family of invertible functions $f_z$, indexed by $z$, such that, given $z$, it is easy to find algorithms $E_z$ and $D_z$ that easily compute $f_z(x)$ and $f_z^{-1}(y)$ for all $x$ and $y$ in the domain and range, respectively, of $f_z$; but for virtually all $z$ and for virtually all $y$ in the range of $f_z$, it is computationally infeasible to compute $f_z^{-1}(y)$ even when one knows $E_z$. Again, this is only a semimathematical definition, but this time the cryptological utility is nakedly apparent.

### B. Public Key-Distribution

As a likely candidate for a one-way function, Diffie and Hellman suggested the *discrete exponential function*

$$f(x) = \alpha^x \quad (\text{modulo } p) \tag{33}$$

where $x$ is an integer between 1 and $p - 1$ inclusive, where, as indicated, the arithmetic is done modulo $p$, a very large prime number, and where $\alpha$ ($1 \le \alpha < p$) is an integer such that $\alpha, \alpha^2, \cdots, \alpha^{p-1}$ are, in some order, equal to $1, 2, \cdots, p - 1$. For example, with $p = 7$, one could take $\alpha = 3$ since $\alpha = 3$, $\alpha^2 = 2$, $\alpha^3 = 6$, $\alpha^4 = 4$, $\alpha^5 = 5$, and $\alpha^6 = 1$. (In algebraic terminology, such an $\alpha$ is called a *primitive element* of the finite field $GF(p)$, and such $\alpha$'s are known always to exist.) If $y = \alpha^x$, then it is natural to write

$$x = \log_\alpha(y) \tag{34}$$

so that inverting $f(x)$ is the problem of calculating *discrete logarithms*. Even for very large $p$, say $p \approx 2^{1000}$, it is quite easy to calculate $f(x)$ by the trick of *square-and-multiply*. For instance, to compute $\alpha^{53} = \alpha^{32+16+4+1}$, one would first form $\alpha^2$, $\alpha^4 = (\alpha^2)^2$, $\alpha^8 = (\alpha^4)^2$, $\alpha^{16} = (\alpha^8)^2$, and $\alpha^{32} = (\alpha^{16})^2$, which requires 5 multiplications. Then one would multiply $\alpha^{32}$, $\alpha^{16}$, $\alpha^4$, and $\alpha$ together, which takes 3 more multiplications for a total of 8 multiplications (modulo $p$). Even with $p \approx 2^{1000}$, calculation of $f(x)$ for any integer $x$, $1 \le x < p$, would take less than 2000 multiplications (modulo $p$).

If the discrete exponential function is indeed one-way, then for virtually all integers $y$, $1 \le y < p$, it must be computationally infeasible to compute $\log_\alpha y$. It was soon realized by Hellman and Pohlig that it was not enough that $p$ be large, $p - 1$ must also have a large prime factor (ideally, $p - 1$ would be twice another prime) if the discrete logarithm is indeed to be hard to compute [24]. With this proviso, the best of known algorithms for computing the discrete logarithm require roughly $\sqrt{p}$ multiplies (modulo $p$), compared to only about $2 \log_2 p$ multiplies for discrete exponentiation. If the discrete logarithm is truly this hard to compute, then the discrete logarithm with the proviso on $p - 1$ is indeed a one-way function. But as of this writing *there is no proof that the discrete exponential, or any other function for that matter, is truly one-way.*

Diffie and Hellman suggested an astoundingly simple way in which the discrete logarithm could be used to create secret keys between pairs of users in a network using only public messages. All users are presumed to know $\alpha$ and $p$. Each user, say user $i$, randomly selects an integer $X_i$ between 1 and $p - 1$ that he keeps as his *private secret*. He then computes

$$Y_i = \alpha^{X_i} \quad (\text{modulo } p). \tag{35}$$

Rather than keeping $Y_i$ secret, he places $Y_i$ in a *certified public directory* accessible to all users. If users $i$ and $j$ wish later to communicate secretly, user $i$ fetches $Y_j$ from the directory, then uses his private secret $X_i$ to form

$$Z_{ij} = (Y_j)^{X_i} = (\alpha^{X_j})^{X_i} = \alpha^{X_iX_j} \quad (\text{modulo } p). \tag{36}$$

In a similar manner, user $j$ forms $Z_{ji}$. But $Z_{ij} = Z_{ji}$ so that users $i$ and $j$ can now use $Z_{ij}$ as the secret key in a conventional cryptosystem. If an enemy could solve the discrete logarithm problem he could take $Y_i$ and $Y_j$ from the directory, solve for $X_i = \log_\alpha Y_i$, and then form $Z_{ij}$ in the same manner as did user $i$—there seems to be no other way for an enemy to find $Z_{ij}$ (but there is no proof of this). The scheme just described is the Diffie–Hellman *public key-distribution system*. Although it is the oldest proposal for eliminating the transfer of secret keys in cryptography, it is still generally considered today to be one of the most secure and most practical public-key schemes.

It should not be overlooked that the Diffie–Hellman public key-distribution scheme (and indeed every public-key technique) *eliminates the need for a secure channel to pass along secrets, but does not eliminate the need for authentication*. The custodian of the public directory must be certain that it is indeed user $i$ who puts the (nonsecret) $Y_i$ into the directory, and user $i$ must be certain that $Y_j$ was actually sent to him by the custodian of the public directory. But it must not be forgotten that in secret-key cryptography, cf. Fig. 1, the receiver must not only be sure that the key $\boldsymbol{Z}$ was kept secret en route to him, but also that the key $\boldsymbol{Z}$ was actually sent by the legitimate sender. *Public-key methods* remove one of these two problems; they *do not create a new authentication problem*, but rather make the old authentication problem more apparent.

## C. The RSA Public-Key Cryptosystem

Having defined a trap-door one-way function, it was an easy step for Diffie and Hellman to propose the structure of a *public-key cryptosystem* for a network of many users. Each user, say user $i$, randomly chooses a value $Z_i$ of the index and keeps $Z_i$ as his *private secret*. He next forms the algorithm $E_{Z_i}$ that he then *publishes* in the certified public directory. He also forms the algorithm $D_{Z_i}$ that he *keeps secret* for his own use. If user $j$ wishes to send a secret message $X$ to user $i$, he fetches $E_{Z_i}$ from the directory. He uses this algorithm to compute the cryptogram $Y = f_{Z_i}(X)$ that he then sends to user $i$. User $i$ uses his private algorithm $D_{Z_i}$ to compute $f_{Z_i}^{-1}(Y) = X$. If $f_z$ is truly a trap-door one-way function, this cryptosystem provides unassailable practical security.

When, for every index $z$, the domain and range of $f_z$ coincide, Diffie and Hellman noted that a trap-door one-way function can be used to create *digital signatures*. If user $i$ wishes to send a *nonsecret* message $X$ (to any or all users in the system) that he wishes to "sign" in a way that the recipient will recognize him unmistakably as the author, he merely uses his private algorithm to form $Y = f_{Z_i}^{-1}(X)$ and transmits $Y$. Every user can fetch the public algorithm $E_{Z_i}$ and then compute $f_{Z_i}(Y) = X$; but no one except user $i$ could have known how to write an intelligible message $X$ in the form $Y = f_{Z_i}^{-1}(X)$, since no one except user $i$ can compute $f_{Z_i}^{-1}$. Of course, user $i$ could also send a signed secret message to user $j$ by encrypting $Y$ in user $j$'s public key $E_{Z_j}$, rather than sending $Y$ in the clear (he might first need to break $Y$ into smaller pieces if $Y$ is "too large to fit" into the domain of $f_{Z_j}$).

It was not at all clear to Diffie and Hellman in 1976 whether trap-door one-way functions existed, and they did not hazard a conjectured such function in their paper. It was left to R. L. Rivest, A. Shamir, and L. Adleman (RSA) of M.I.T. to make the first proposal of a possible trap-door one-way function in their remarkable 1978 paper, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems" [25]—it is interesting to note that authentication received higher billing than secrecy in their title. The RSA trap-door one-way function is the essence of simplicity, but to describe it we need a few ideas from elementary number theory.

Let gcd $(i, n)$ denote the greatest common divisor of the integers $i$ and $n$ (not both 0). For example, gcd $(12, 18) = 6$. The *Euler totient function* $\phi(n)$, where $n$ is a positive integer, is defined as the number of positive integers $i$ less than $n$ such that gcd $(i, n) = 1$ (except that $\phi(1)$ is defined to be 1). For instance, $\phi(6) = 2$ since for $1 \le i < 6$ only $i = 1$ and $i = 5$ give gcd $(i, 6) = 1$. One sees immediately that for a prime $p$, $\phi(p) = p - 1$; and just a little thought more shows that if $p$ and $q$ are distinct primes, then

$$\phi(pq) = (p - 1)(q - 1). \tag{37}$$

For instance, $\phi(6) = \phi(2 \times 3) = 1 \times 2 = 2$. A celebrated theorem of Euler (1707–1783) states that for any positive integers $x$ and $n$ with $x < n$

$$x^{\phi(n)} = 1 \quad (\text{modulo } n) \tag{38}$$

provided that gcd $(x, n) = 1$. For example

$$5^2 = 1 \quad (\text{modulo } 6).$$

The last fact from number theory that we need dates back to Euclid (c. 300 B.C.). If $e$ and $m$ satisfy $0 < e < m$ and gcd $(m, e) = 1$, then there is a unique $d$ such that $0 < d < m$ and

$$de = 1 \quad (\text{modulo } m), \tag{39}$$

moreover $d$ can be found in the process of using Euclid's "extended" algorithm for computing gcd $(m, e)$, cf. [26, p. 14].

The *RSA trap-door one-way function* is just the discrete exponentiation

$$f_z(x) = x^e \quad (\text{modulo } n) \tag{40}$$

where $x$ is a positive integer less than $n = pq$ and where the "trap-door" $z = \{p, q, e\}$; here $p$ and $q$ are distinct very large primes such that $\phi(n) = (p - 1)(q - 1)$ has a very large prime factor, and $e$ is a positive integer less than $\phi(n)$ such that gcd $(e, \phi(n)) = 1$. The easy-to-find algorithm $E_z$ to compute $f_z$ easily is exponentiation by square-and-multiply; *publishing this algorithm amounts just to publishing n and e*. The inverse function is

$$f_z^{-1}(y) = y^d \quad (\text{modulo } n) \tag{41}$$

where $d$ is the unique positive integer less than $n$ such that

$$de = 1 \quad (\text{modulo } \phi(n)). \tag{42}$$

The easy-to-find (when one knows $z$) algorithm $D_z$ to compute $f_z^{-1}$ is also exponentiation by square-and-multiply; the decrypting exponent $d$ is found using Euclid's algorithm for computing gcd $(e, \phi(n))$.

That (41) really gives the inverse function for (40) can be seen as follows. Equation (42) is equivalent to the statement (in ordinary integer arithmetic) that

$$de = \phi(n)Q + 1 \tag{43}$$

for some integer $Q$. From (40) and (43), we obtain

$$\begin{aligned}(x^e)^d &= x^{\phi(n)Q+1} \quad (\text{modulo } n)\\ &= (x^{\phi(n)})^Q x \quad (\text{modulo } n)\\ &= x \qquad (\text{modulo } n) \qquad (44)\end{aligned}$$

where at the last step we used Euler's theorem (38). [The wary reader will have noted that Euler's theorem requires gcd $(x, n) = 1$; but in fact (38) holds for all positive integers $x$ less than $n$ in the special case when $n$ is the product of two distinct primes.] Equation (44) shows that raising a number to the $d$ power (modulo $n$) is indeed the inverse of raising a number to the $e$ power (modulo $n$). It remains to show why RSA believed (as do most cryptographers today) that it is comutationally impossible to invert this function $f_z$ when one knows only $n$ and $e$, and also how it it possible easily to choose *randomly* the two distinct and very large primes $p$ and $q$ as must be done for an enemy to be unable to guess $p$ and $q$.

The enemy knows only $n$ and $e$. But if he can factor $n = pq$, then he knows the entire trap-door $z = \{p, q, e\}$, and hence can decrypt just as readily as the legitimate receiver. The security of the RSA public-key cryptosystem depends on the assumption that *any way of inverting $f_z$ is equivalent to factoring $n = pq$*, i.e., given any way to invert $f_z$, one could with at most a little more computational work go on to factor $n$. In their paper [25], RSA show that this is true for the most likely ways that one might try to factor $n$, but the assumption has never been proved. But is the attack by factoring $n$ computationally infeasible? The answer is yes if one chooses $p$ and $q$ on the order of 100 decimal digits each (as RSA suggested ten years ago) *and* if there is no revolutionary breakthrough in factoring algorithms. As Rivest [27] recently pointed out, all of the best factoring algorithms today have running times upper-bounded by the same peculiar-looking function which, for numbers to be factored between 50 and 200 decimal digits, increases by a factor of 10 for every additional 15 digits (roughly) in the number. Today it takes about 1 day on a supercomputer to factor a number of about 80 decimal digits. It would take $10^8$ times that long to factor a 200 digit number $n = pq$, roughly half a million years! One of the by-products of the RSA paper has been a revival of interest in factoring, but this accelerated research effort has produced no revolutionary breakthrough. Proponents of the RSA public-key cryptosystem believe that it never will. An interesting fact is that the best algorithms today for solving the (modulo $p$) discrete logarithm problem [28] and the best algorithms for factoring $n$ [29] require about the same amount of computation when $p \approx n$, so that the RSA trap-door function (49) and the Diffie–Hellman function (33), as of today, have about the same claim to be called "one-way."

It remains to consider how one can randomly choose the very large primes, $p$ and $q$, required for RSA. A theorem of Tchebychef, cf. [30, pp. 9–10], states that the fraction of positive integers less than any large integer $m$ that are primes is close to $(\ln m)^{-1}$. For instance, the fraction of integers less than $10^{100}$ that are primes is about $(\ln 10^{100})^{-1} \approx \frac{1}{230}$. Because 90 percent of these integers lie between $10^{99}$ and $10^{100}$, the fraction of primes in this range is also about $\frac{1}{230}$. Thus, if one chooses an integer between $10^{99}$ and $10^{100}$ completely at random the chances that one chooses a prime are about $\frac{1}{230}$. One easily doubles the odds to $\frac{1}{115}$ if one is sensible enough to choose only odd integers. One needs then only about 115 such choices on the average before one has chosen a prime. But how does one recognize a prime? It is a curious fact that one can rather easily test quite reliably whether an integer is a prime or not, even if one cannot factor that integer after one discovers that it is not a prime. Such primality tests rely on a *Theorem of Fermat* (1601–1665) that asserts that for any positive integer $b$ less than a prime $p$

$$b^{p-1} = 1 \quad (\text{modulo } p). \qquad (45)$$

For instance, $2^4 = 1$ (modulo 5). [The reader may have noticed that (45) is a special case of (38), but he should remember that Fermat lived a century before Euler!] If one has an integer $r$ that one wishes to test for primeness, one can choose any positive integer $b$ less than $r$ and check whether

$$b^{r-1} \overset{?}{=} 1 \quad (\text{modulo } r). \qquad (46)$$

If the answer is no, one has the absolute assurance of Fermat that $r$ is not a prime. If the answer is yes, one can begin to suspect that $r$ is a prime, and one then christens $r$ a *pseudoprime to the base b*. If $r$ is not a prime, it turns out that it can be a pseudoprime for less (actually much less) than about half of the possible bases $b$. Thus if $r$ is very large, and one independently chooses $t$ bases $b$ completely at random, the probability is less than about $2^{-t}$ that $r$ will pass Fermat's test (46) for all these bases if $r$ is not truly a prime. If we take, say $t = 100$, then we can be virtually certain that $r$ is a prime if it passes $t$ indepedent Fermat tests. Such "probabilistic tests for primeness" were introduced by Solovay and Strassen, and have been further refined by Rabin [31]. Such tests are today being used to check randomly-chosen odd integers for primeness until one has found the two distinct large primes one needs for the RSA trap-door one-way function, or, more precisely, until one is sufficiently sure that he has found two such primes.

There are VLSI chips today that can implement the RSA encrypting and decrypting function at a data rate of a few kilobits per second. (These same chips can also be used to implement Fermat's test, and thus to find the needed 100 decimal digit primes, $p$ and $q$). Rivest [27] has given convincing arguments that significantly higher data rates will never be achieved. For many cryptographic applications, these data rates are too low. In such cases, the RSA public-key cryptosystem may still desirably be used to distribute the secret keys that will then be used in high-speed secret-key ciphers, such as DES or certain stream ciphers. And the RSA algorithm may still desirably be used for authentication in its "digital signature" mode.

Before closing this section on the RSA system, we should mention that Rabin [32] has developed a variant of the RSA public-key system for which he *proved* that being able to find the plaintext $X$ from the cryptogram $Y$ is *equivalent to factoring $n = pq$*. The system is somewhat more complicated than basic RSA, but Williams [33] refined the variant so that the extra complication is quite tolerable. This might seem to be the ultimate "RSA system," but paradoxically the breaking-is-provably-equivalent-to-factoring versions of RSA have a new weakness that was pointed out by Rivest. The proof of their equivalence to factoring is *constructive*, i.e., one shows that if one could solve $Y = X^e$ (modulo $n$) for $X$ in these systems [which differ from RSA in that now gcd $(e, \phi(n)) \neq 1$], then one could easily go on to factor $X$.

But this means that these systems *succumb to a chosen-ciphertext attack* in which an enemy randomly chooses $X'$, computes $Y = (X')^e$ and then submits $Y$ to the decrypter, who returns a solution $X$ of $Y = X^e$ [where the fact that gcd $(e, \phi(n)) \neq 1$ results in the situation that the solution is not unique so that $X \neq X'$ is possible]. The chances are $\frac{1}{2}$ that the returned $X$ together with $X'$ will give the enemy the information he needs to factor $n = pq$ and thus to break the system. In a public-key environment, such a chosen-ciphertext attack becomes a distinct possibility. The net result is that most cryptographers prefer to use the original RSA public-key cryptosystem, and to pray for the day when a *nonconstructive* proof is given that breaking it is equivalent to factoring.

This is perhaps the appropriate point to mention that a *public-key cryptosystem, if it is secure at all, is secure against a chosen-plaintext attack*. For the enemy cryptanalyst is always welcome to fetch the algorithm $E_z$ from the public directory and then to compute the cryptograms, $y = f_y(x)$, for as many plaintexts $x$ as he pleases. This shows that a trap-door one-way function must necessarily be much more difficult to invert that the encrypting function of a conventional secret-key cipher that is also secure against a chosen-plaintext attack. In the latter case, the enemy can still (by assumption) obtain the cryptograms $y$, for whatever plaintexts $x$, he pleases. But he no longer has the luxury of watching the encryption algorithm execute its encryptions, because the secret key is an ingredient of the algorithm.

### D. Some Remarks on Public-Key Cryptography

The Diffie–Hellman one-way function and the RSA trapdoor one-way function suffice to illustrate the main ideas of public-key cryptography, which is why we have given them rather much attention. But a myriad of other such functions have been proposed. Some have almost immediately been exposed as insecure, others appear promising. But no one has yet produced a proof that any function is a one-way function or a trap-door one-way function. Even the security of the Rabin variant of RSA rests on the unproved (but very plausible) assumption that factoring large integers is computationally infeasible.

There has been some hope that the new, but rapidly-evolving, theory of computational complexity, particularly Karp's theory of NP-completeness, cf. [34], will lead to provably one-way functions or provably trap-door one-way functions. This hope was first expressed by Diffie and Hellman [12], but has thus far led mainly to failures such as the spectacular failure of the Merkle–Hellman trap-door-knapsack public-key cryptosystem. Part of the difficulty has been that NP-completeness is a worst-case phenomenon, not a "virtually all cases" phenomenon as one requires in public-key cryptography. For instance, Even, Lempel, and Yacobi have constructed an amusing example of a public-key cryptosystem whose breaking is equivalent to solving an "NP-hard" problem, but which can virtually always be broken [35]. [A problem is NP-hard if its solution is at least as difficult as the solution of an NP-complete problem.] But the greater difficulty has been to formulate a trap-door one-way function whose inversion would require the solution of an NP-complete problem; this has not yet been accomplished. For instance, the inversion of the Merkle–Hellman trapdoor-knapsack one-way function is actually an easy problem disguised to resemble an NP-hard problem; Shamir broke this public-key cipher, not by solving the NP-hard problem, but by stripping off the disguise.

## IV. Cryptographic Protocols

### A. What is a Protocol?

It is difficult to give a definition of "protocol" that is both precise and general enough to encompass most things to which people apply this label in cryptography and elsewhere. Roughly speaking, we might say that a *protocol* is a specified sequence of actions by which two or more parties cooperatively accomplish some task. A public-key cryptosystem, for instance, can be considered a protocol, based on a trap-door one-way function, by means of which the users of the system and the custodian of the public directory cooperate to ensure the privacy of messages sent from one user to another.

### B. A Key-Distribution Protocol

Many cryptographers, particularly those skeptical of public-key ideas, consider the *key management problem* (i.e., the problem of securely distributing and changing secret keys) to be the main practical problem in cryptography. For example, if there are $S$ users in the system, one will need $S(S-1)/2$ different secret keys if one is to have a dedicated secret key for every possible pair of users—an unwelcome prospect in a large system. It is unlikely that any user will ever wish to send secret messages to more than a few other users, but in advance one usually does not know who will later want to talk secretly to whom. A popular solution to this problem is the following key-distribution protocol that requires the advance distribution of only $S$ secret keys, but still permits any pair of users to communicate secretly; there is a needed new entity, however, the *trusted key distribution center* (TKDC).

*Key Distribution Protocol:*

1) The TKDC securely delivers a randomly-chosen secret key $Z_i$ to user $i$ in the system, for $i = 1, 2, \cdots, S$.

2) When user $i$ wishes to communicate secretly to user $j$, he sends the TKDC a request (which can be in the clear) over the public network for a secret key to be used for this communication.

3) The TKDC randomly chooses a new secret key $Z_{ij}$ which it treats as part of the plaintext. The other part of the plaintext is a "header" in which user $i$ and user $j$ are identified. The TKDC encrypts this plaintext in both key $Z_i$ and key $Z_j$ with whatever secret-key cipher is installed in the system, then sends the first cryptogram to user $i$ and the second to user $j$ over the public network.

4) Users $i$ and $j$ decrypt the cryptograms they have just received and thereby obtain the secret key to be used for encrypting further messages between these two users.

This protocol sounds innocent enough, but its security against a ciphertext-only attack requires more than ciphertext-only security of the system's secret-key cipher. Why? Because in step 3) we see that an enemy cryptanalyst will have access to two cryptograms in different keys for the *same* plaintext. This can be helpful to the cryptanalyst, although it does not give him as much information as he could get in a chosen-plaintext attack on the individual ciphers. Thus, security of the system's cipher against a chosen-plaintext attack will make this protocol also secure against chosen-plaintext attacks. The point to be made here is that when one embeds a cipher into a protocol, *one must be very careful to ensure that whatever security is assumed for the cipher is not compromised by the protocol.*

## C. *Shamir's Three-Pass Protocol*

One of the most interesting cryptographic protocols, due to A. Shamir in unpublished work, shows that secrecy can be obtained with no advance distribution of either secret keys or public keys. The protocol assumes two users connected by a link (such as a seamless optical fiber or a trustworthy but curious postman) that guarantees that the enemy cannot insert, or tamper with, messages but allows the enemy to read all messages sent over the link. The users are assumed to have a secret-key cipher system whose encrypting function $E_Z(\cdot)$ has the *commutative property*, that, for all plaintexts, $x$, and all keys, $\mathbf{z}_1$ and $\mathbf{z}_2$,

$$E_{\mathbf{z}_2}(E_{\mathbf{z}_1}(\mathbf{x})) = E_{\mathbf{z}_1}(E_{\mathbf{z}_2}(\mathbf{x})) \tag{47}$$

i.e., the result of a double encryption is the same whether one uses first the key $z_1$ and then the key $z_2$ or *vice versa.* There are many such ciphers, e.g., the one-time pad (4) fits the bill because $(\mathbf{x} \oplus \mathbf{z}_1) \oplus \mathbf{z}_2 = (\mathbf{x} \oplus \mathbf{z}_2) \oplus \mathbf{z}_1$, where the addition is bit-by-bit modulo-two.

*Shamir's Three Pass Protocol:*

1) Users $A$ and $B$ randomly choose their own private secret keys, $\mathbf{Z}_A$ and $\mathbf{Z}_B$, respectively.

2) When user $A$ wishes to send a secret message $\mathbf{X}$ to user $B$, he encrypts $\mathbf{X}$ with his own key $Z_A$ and sends the resulting crypotogram $\mathbf{Y}_1 = E_{Z_A}(\mathbf{X})$ on the open-but-tamperproof link to user $B$.

3) User $B$, upon receipt of $\mathbf{Y}_1$, treats $\mathbf{Y}_1$ as plaintext and encrypts $\mathbf{Y}_1$ with his own key $\mathbf{Z}_B$. He sends the resulting cryptogram $\mathbf{Y}_2 = E_{\mathbf{Z}_B}(Y_1) = E_{\mathbf{Z}_B}(E_{\mathbf{Z}_A}(\mathbf{X}))$ on the open-but-tamperproof link to user $A$.

4) User $A$, upon receipt of $\mathbf{Y}_2$, decrypts $\mathbf{Y}_2$ with his own key $\mathbf{Z}_A$. Because of the commutative property (47), this removes the former encryption by $\mathbf{Z}_A$ and results in $\mathbf{Y}_3 = E_{\mathbf{Z}_B}(\mathbf{X})$. User $A$ then sends $\mathbf{Y}_3$ over the open-but-tamperproof link to user $B$.

5) User $B$, upon receipt of $\mathbf{Y}_3$, decrypts $\mathbf{Y}_3$ with his own key $Z_B$ to obtain $\mathbf{X}$, the message that $A$ has now successfully sent to him secretly.

What secret-key cipher shall we use in this protocol? Why not the one-time pad, a cipher that gives perfect secrecy? If we use the one-time pad, the three cryptograms become

$$\begin{aligned} \mathbf{Y}_1 &= \mathbf{X} \oplus \mathbf{Z}_A \\ \mathbf{Y}_2 &= \mathbf{X} \oplus \mathbf{Z}_A \oplus \mathbf{Z}_B \\ \mathbf{Y}_3 &= \mathbf{X} \oplus \mathbf{Z}_B. \end{aligned} \tag{48}$$

The enemy cryptanalyst sees all three cryptograms, and hence can form

$$\mathbf{Y}_1 \oplus \mathbf{Y}_2 \oplus \mathbf{Y}_3 = \mathbf{X}$$

where we have used the fact that two identical quantities sum to $\mathbf{O}$ modulo-two. Thus, the 3-pass protocol is completely insecure when we use the one-time pad for the embedded cipher! The reason for this is, as (48) shows, that the effect of the protocol is that each of the two ciphers get used "$1\frac{1}{2}$ times," rather than only once as is required for the security of the "one-time" pad.

Is there a cipher that can be used in the Shamir 3-pass protocol and still retain its security? There seems to be. Let $p$ be any large prime for which $p-1$ has a large prime factor (to make the discrete logarithm problem in modulo $p$ arithmetic computationally infeasible to solve). Randomly choose a positive integer $e$ less than $p-1$ such that gcd $(e, p-1) = 1$, and let $d$ be the unique positive integer less than $p-1$ such that

$$de = 1 \quad (\text{modulo } p-1). \tag{49}$$

Let $Z = (d, e)$ be the secret key and take the encrypting and decrypting functions to be

$$\begin{aligned} y &= E_Z(x) = x^e \quad (\text{modulo } p) \\ x &= D_Z(y) = y^d \quad (\text{modulo } p) \end{aligned} \tag{50}$$

where $x$ and $y$ are positive integers less than $p$. [The fact that $y^d = x^{de} = x$ (modulo $p$) is an easy consequence of Fermat's theorem (45) and the fact that (49) implies $de = Q(p-1) + 1$ for some integer $Q$.] That this cipher has the commutative property (47) follows from (50) because

$$(x^{e_1})^{e_2} = x^{e_1 e_2} = x^{e_2 e_1} \quad (\text{modulo } p).$$

When this cipher is used in the 3-pass protocol, the three cryptograms become

$$\begin{aligned} y_1 &= x^{e_A} \quad (\text{modulo } p) \\ y_2 &= x^{e_A e_B} \quad (\text{modulo } p) \\ y_3 &= x^{e_B} \quad (\text{modulo } p). \end{aligned} \tag{51}$$

If one can solve the discrete logarithm problem, one can obtain

$$\log_\alpha y_1 = e_A \log_\alpha x \qquad (\text{modulo } p-1) \tag{52a}$$

$$\log_\alpha y_2 = e_A e_B \log_\alpha x \quad (\text{modulo } p-1) \tag{52b}$$

where $\alpha$ is any chosen primitive element for arithmetic modulo $p$, and where we have used the fact that the arithmetic of discrete logarithms is modulo-$(p-1)$ arithmetic—this follows from Fermat's theorem (45) that gives $\alpha^{p-1} = 1 = \alpha^0$. We can now use Euclid's extended gcd algorithm, cf. Section III-C, to find the positive integer $b$ less than $p - 1$ such that

$$b \log_\alpha y_1 = 1 \quad (\text{modulo } p-1)$$

which from (52a) further implies

$$b e_A \log_\alpha x = 1 \quad (\text{modulo } p-1). \tag{53}$$

Multiplying (52b) by $b$ on both sides, then using (53), we obtain

$$b \log_\alpha y_2 = e_B \quad (\text{modulo } p-1). \tag{54}$$

Thus, an enemy who can solve the discrete logarithm problem for modulo-$p$ arithmetic can find $e_B$, hence also $d_B$, and thus read the message $x$ just as well as user $B$. There seems to be no way for the enemy to find $x$ without equivalently solving the discrete logarithm problem, but (like so many other things in public-key cryptography) this has never been proved. This particular cipher for the 3-pass protocol was proposed by Shamir (and independently but later by J. Omura, who was aware of Shamir's 3-pass protocol, but unaware of his proposed cipher for the protocol).

## D. Closing Remarks

There are many protocols that have been proposed recently by cryptologic researchers. One of the most amusing is the Shamir-Rivest-Adleman protocol for "mental poker," a protocol that manages to allow an honest game of poker to be played with no cards [36]. Such frivolous-sounding protocols have a serious cryptographic purpose, however; in this case one could take the purpose to be a protocol for assuring the authenticity of randomly-chosen numbers. Similarly, Chaum [37] has proposed an interesting protocol by which parties making transactions through a bank can do so without the bank ever knowing who is paying what to whom that also suggests a cryptographic application in key distribution. Protocol formulation has recently gained new momentum and has become one of the most active areas of current cryptologic research, as well as one of the most difficult, particularly when one seeks particular cryptographic functions to imbed in the protocol without compromise of their security. The RSA trap-door one-way function is far and away the most frequently used function for this purpose.

We have not mentioned many of the important contributions to cryptology made in the past 10 years. It has *not* been our purpose to *survey* research in cryptology, but rather to sketch the intellectual outlines of the subject. The reader who wishes to bring himself abreast of current research in cryptology, will find the Proceedings of the CRYPTO conference (held annually in Santa Barbara since 1981) and of the EUROCRYPT conference (held annually since 1982) to be invaluable. There are also several recent general textbooks [38]-[42] on cryptology that will give the reader an orderly development of the subject, and a recent text [43] gives a broad treatment of the number-theoretic concepts on which much of present-day public-key cryptology depends. The recent book by Rueppel is a good source of information on stream ciphers.

## References

[1] W. Diffie and M. E. Hellman, "Privacy and authentication: An introduction to cryptography," *Proc. IEEE*, vol. 67, pp. 397-427, Mar. 1979.

[2] G. J. Simmons, "Cryptology," in *Encyclopaedia Britannica*, ed. 16. Chicago, IL: Encyclopaedia Britannica Inc., 1986, pp. 913-924B.

[3] D. Kahn, *The Codebreakers, The Story of Secret Writing.* New York, NY: MacMillan, 1967.

[4] —, *The Codebreakers, The Story of Secret Writing*, abridged ed. New York, NY: Signet, 1973.

[5] R. C. Merkle and M. E. Hellman, "Hiding information and signatures in trapdoor knapsacks," *IEEE Trans. Informat. Theory*, vol. IT-24, pp. 525-530, Sept. 1978.

[6] A. Shamir, "A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem," *IEEE Trans. Informat. Theory*, vol. IT-30, pp. 699-704, Sept. 1984.

[7] D. B. Newman, Jr., and R. L. Pickholtz, "Cryptography in the private sector," *IEEE Commun. Mag.*, vol. 24, pp. 7-10, Aug. 1986.

[8] G. S. Vernan, "Cipher printing telegraph systems for secret wire and radio telegraphic communications," *J. Amer. Inst. Elec. Eng.*, vol. 55, pp. 109-115, 1926.

[9] A. Hodges, *Alan Turing, The Enigma.* New York, NY: Simon and Schuster, 1983.

[10] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst. Tech. J.*, vol. 28, pp. 656-715, Oct. 1949.

[11] —, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, 623-656, July and Oct. 1948.

[12] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Informat. Theory*, vol. IT-22, pp. 644-654, Nov. 1976.

[13] R. C. Merkle, "Secure communication over insecure channels," *Comm. ACM*, pp. 294-299, Apr. 1978.

[14] G. J. Simmons, "Authentication theory/coding theory," in *Advances in Cryptology, Proceedings of CRYPTO 84*, G. R. Blakley and D. Chaum, Eds. Lecture Notes in Computer Science, No. 196. New York, NY: Springer, 1985, pp. 411-431.

[15] W. Feller, *An Introduction to Probability Theory and its Applications*, vol. 2. New York, NY: Wiley, 1966.

[16] "Data encryption standard," FIPS PUB 46, National Tech. Info. Service, Springfield, VA, 1977.

[17] R. Morris, "The data encryption standard—retrospective and prospects," *IEEE Commun. Mag.*, vol. 16, pp. 11-14, Nov. 1978.

[18] W. Diffie and M. E. Hellman, "Exhaustive cryptanalysis of the NBS data encryption standard," *Computer*, vol. 10, pp. 74-84, June 1977.

[19] M. Hellman, "A cryptanalytic time-memory trade-off," *IEEE Trans. Informat. Theory*, vol. IT-26, pp. 401-406, July 1980.

[20] R. C. Merkle and M. E. Hellman, "On the security of multiple encryption," *Comm. ACM*, vol. 24, pp. 465-467, July 1981.

[21] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Informat. Theory*, vol. IT-15, pp. 122-127, Jan. 1969.

[22] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications," *IEEE Trans. Informat. Theory*, vol. IT-30, pp. 776-780, Sept. 1984.

[23] J. L. Massey and I. Ingemarsson, "The Rip van Winkle cipher—A simple and provably computationally secure cipher with a finite key," in *IEEE Int. Symp. on Informat. Theory*, (Brighton, England) (abstr.), p. 146, June 24-28, 1985.

[24] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms in GF(p) and its cryptographic significance," *IEEE Trans. Informat. Theory*, vol. IT-24, pp. 106-110, Jan. 1978.

[25] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Comm. ACM*, vol. 21, pp. 120-126, Feb. 1978.

[26] D. E. Knuth, *The Art of Computer Programming, Vol. 1, Fundamental Algorithms.* Reading, MA: Addison-Wesley, 1973.

[27] R. L. Rivest, "RSA chips (past/present/future)," presented at Eurocrypt 84, Paris, France, Apr. 9-11, 1984.

[28] A. M. Odlyzko, "On the complexity of computing discrete logarithms and factoring integers," to appear in *Fundamental Problems in Communication and Computation*, B. Gopinath and T. Loven, Eds. New York, NY: Springer.

[29] C. Pomerance, "Analysis and comparison of some integer factoring algorithms," in *Computational Number Theory*, H. W. Lenstra, Jr., and R. Tijdeman, Eds. Amsterdam, The Netherlands: Math. Centre Tract, 1982.

[30] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*, ed. 4. London, England: Oxford, 1960.

[31] M. O. Rabin, "Probabilistic algorithm for primality testing," *J. Number Theory*, vol. 12, pp. 128-138, 1980.

[32] —, "Digital signatures and public-key functions as intractable as factorization," Tech. Rep. LCS/TR212, M.I.T. Lab. for Comp. Sci., Cambridge, MA, 1979.

[33] H. C. Williams, "An $M^3$ public-key encryption scheme," in *Advances in Cryptology, Proceedings of CRYPTO 85*, H. C. Williams, Ed. Lecture Notes in Computer Science, No. 218. New York, NY: Springer, 1986, pp. 358-368.

[34] M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness.* New York, NY: Freeman, 1979.

[35] A. Lempel, "Cryptology in transition," *Computing Surveys*, vol. 11, pp. 285-303, Dec. 1979.

[36] A. Shamir, R. L. Rivest, and L. Adleman, "Mental poker," in *Mathematical Gardener*, D. E. Klarner, Ed. New York, NY: Wadsworth, 1981, pp. 37-43.

[37] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Comm. ACM*, vol. 28, pp. 1030-1044, Oct. 1985.

[38] H. Beker and F. Piper, *Cipher Systems, The Protection of Communications.* New York, NY: Van Nostrand, 1982.

[39] D. W. Davies and W. L. Price, *Security for Computer Networks.* New York, NY: Wiley, 1984.

[40] D. E. R. Denning, *Cryptography and Data Security.* Reading, MA: Addison-Wesley, 1982.

[41] A. C. Konheim, *Cryptography, A Primer.* New York, NY: Wiley, 1981.
[42] C. H. Meyer and S. M. Matyas, *Cryptography: A New Dimension in Computer Data Security.* New York, NY: Wiley, 1982.
[43] E. Kranakis, *Primality and Cryptography.* New York, NY: Wiley, 1986.
[44] R. Rueppel, *Analysis and Design of Stream Ciphers.* New York, NY: Springer, 1986.

**James L. Massey** (Fellow, IEEE) was born in Wauseon, OH, on February 11, 1934. He received the B.S.E.E. degree from the University of Notre Dame, Notre Dame, IN, in 1956, and the S.M. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1960 and 1962, respectively.

From 1956 to 1959, he served as a Communications Officer in the U.S. Marine Corps. From 1962 to 1977, he served on the faculty of the University of Notre Dame, the last five years as the Frank Freimann Professor of Electrical Engineering. During 1966–1967, he was a Visiting Associate Professor of Electrical Engineering at M.I.T., and during 1971–1972 he was a Guest Professor at the Technical University of Denmark, Lyngby. He spent 1977–1978 as a Visiting Professor of Electrical Engineering and Computer Science at M.I.T., prior to joining the Systems Science Department at UCLA in 1978. In 1980, he assumed his present position as Professor für Digitaltechnik at the Swiss Federal Technical University, Zurich. He served in 1969 as Chairman of the IEEE Group on Information Theory and was Co-Chairman of the 1974 IEEE International Symposium on Information Theory. He was Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY from 1975 to 1978; he also served this journal from 1972 to 1975 as Associate Editor for Algebraic Coding, and was Guest Editor for the March 1985 Special Issue on Random-Access Communications. He is currently President of the International Association for Cryptologic Research (IACR).

Dr. Massey received the 1964 Paper Award of the IEEE Group on Information Theory for his monograph, "Threshold Decoding," that was published by the M.I.T. Press. He also received a 1975 Western Electric Fund Award from the American Society for Engineering Education for "excellence in the instruction of engineering students." He was elected Professor *causa honoris* by the Northwest Telecommunications Engineering Institute, Xian, China, in 1985.