

Cryptography and Data Protection

Koninklijke Nederlandse Akademie van Wetenschappen
Verhandelingen, Afd. Natuurkunde, Eerste Reeks, deel 38

Cryptography and Data Protection

Proceedings of a Symposium at the Royal Netherlands Academy of
Arts and Sciences on 19th December 1990

J.H. van Lint
R. Tijdeman
editors

North-Holland, Amsterdam/Oxford/New York/Tokyo, 1992

ISBN 0-444-85746-X

Contents

Preface	vii
Contemporary Cryptology: An Introduction James L. Massey	1
Public Key Cryptology and Fundamental Research; their Interaction Henk C.A. van Tilborg	41
Protection of Medical Data J.H. van Bommel	51
Application of Encryption Techniques for Security Purposes in Financial Systems T.W.M. Jongmans	61
Computing and Security; a Task for the Lawyer? H. Franken	71
Security without Identification: Card Computers to make Big Brother Obsolete David Chaum	81

Preface

On 19th December, 1990 the Royal Netherlands Academy of Arts and Sciences organised a symposium on cryptography and data protection. Lecturers from various disciplines presented their point of view. Since the resulting multi-sided view on this contemporary topic is of interest for a broad scientific public, it was decided to publish the proceedings of the symposium.

The first paper, by J.L. Massey, gives an appraisal of the current status of cryptologic research. The principal concepts of both secret-key and public-key cryptography are described. Shannon's theory of secrecy and Simmons' theory of authenticity are reviewed. Some important public-key systems and cryptographic protocols are treated.

Public key cryptosystems are based on mathematical operations that are easy to perform, but without additional information difficult to undo. In the second paper H.C.A. van Tilborg provides some mathematical background to the logarithm system, the RSA System, both of which are explained by Massey, and the knapsack system. He further deals with some factorisation methods which were found in connection with developments in cryptography.

The next two lectures concern practical aspects of data protection. J.H. van Bommel discusses the purposes of computer storage and electronic exchange of medical data and the consequences for data protection. To this end he considers the different types of medical data and their use, the different users of medical data and some legal aspects of privacy.

Application of data protection in financial systems is treated by T.W.M. Jongmans. The urgency of protection here is obvious, since the data themselves are money. He compares the classical basis of security with the modern security techniques. Here theoretical cryptographic systems, discussed by Massey, find a practical implementation. This is illustrated by the security concept for the National Payments Circuit. He further indicates which difficulties of operating encryption techniques arise in practice.

Legal aspects of data protection are discussed by H. Franken. The rapid developments in information technology and telecommunications stimulate abuse and create uncertainty. In particular, it is not clear whether data should be considered as goods and be subject to laws dealing with goods. New criminal and civil laws have to delimit the border between what is permitted and what is not. Franken stresses the limited function of the law, since for genuine enforcement the criminal law should be invoked sparingly and cannot replace measures initiated by the owners and users themselves.

The final paper by D. Chaum is a view in the future. It describes a system of card computers which provides security for the users without the need to reveal their identity. In place of the variety of 'tokens' issued by organisations today one single credit-card sized card computer would suffice. Chaum explains the principles of digital signatures, payment transactions and credential transactions based on cryptographic systems treated by Massey, and discusses the advantages of the new system for individuals and organisations. His paper illustrates some concepts which are basic for developments which will change our financial system.

J.H. van Lint
R. Tijdeman
editors

Contemporary Cryptology: An Introduction

by James L. Massey, Fellow IEEE*

*Institute for Signal and Information Processing, Swiss Federal Institute of Technology,
CH-8092 Zürich, Switzerland*

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 Simmon'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 counter-claims 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 book.

* © 1991 IEEE. Reprinted, with permission, from *Contemporary Cryptology: The Science of Information Integrity*, G.J. Simmons, Editor. IEEE Press, Piscataway, NJ.

The present paper is an updated, expanded, and slightly revised version of our earlier paper [45], large sections of which appear virtually unchanged herein. The reader who is quite familiar with this earlier paper may wish to concentrate his attention on the new material that appears in this one. Reference numbers [45] and onwards denote references added to the earlier paper and their appearance will flag such a reader's attention to the substantially new segments of this paper.

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 earlier survey of cryptology [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.

B. Cryptology 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 eight-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 a 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 submit 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.

In the preceding paragraph, we abided by the long-accepted attribution of the dogmatic pronouncement, 'Gentlemen do not read each other's mail', to

H.L. Stimson in 1929. Kruh [46] has recently given a convincing historical argument suggesting that these famous words may in fact have been uttered by Stimson first in 1946 during his interviews with McGeorge Bundy, who was then preparing Stimson's authorized biography [47]. Kruh [46, p. 80] concludes: 'It thus seems highly likely that Stimson's 1946 remark accurately described his motivation for closing the Cipher Bureau in 1949. But whether he also said it then remains unknown.'

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 fifteen 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 research 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 fraught 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

$$(1) \quad y = x \oplus z$$

where x is the plaintext letter ($A = 0, B = 1, \dots, Z = 25$), 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 has 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 plaintext 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 [9, 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 cryptography*. 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 re-

quired amount of secret key that must be transferred securely to the intended receiver when any perfect cipher is used.

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 secret key, $Z = [Z_1, Z_2, \dots, Z_K]$, 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, $X = [X_1, X_2, \dots, X_M]$. The private random source (whose purpose will soon be evident) generates the private randomizer, $S = [S_1, S_2, \dots, S_J]$, and the public random source (whose purpose will be seen later) generates the *public randomizer*, $R = [R_1, R_2, \dots, R_T]$. The encrypter forms the cryptogram, $Y = [Y_1, Y_2, \dots, Y_N]$, as a function of X , R , S and Z . We write this encrypting transformation as

$$(2) \quad Y = E_{ZRS}(X)$$

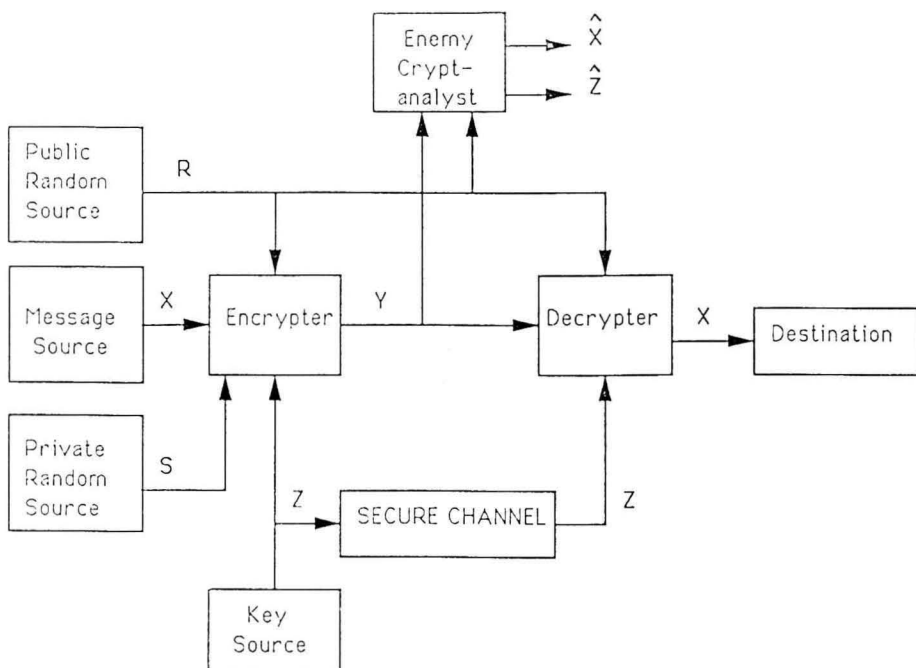


Fig. 1. Model of a secret-key cryptosystem.

to underscore the fact that it is useful to think of the cryptogram Y as a function of the plaintext X with the particular function being specified by the values of the secret key Z and of the randomizing sequences R and S . As fig. 1 implies, the decrypter must be able to invert this transformation without knowledge of the *private* randomizing sequence S . That is,

$$(3) \quad X = D_{ZR}(Y),$$

which expresses the fact that the plaintext X must be a function of the cryptogram Y where the particular function is determined only by the secret key Z and the public randomizer R . The enemy cryptanalyst observes the cryptogram Y and the public randomizer R but nothing else. He then forms his estimate \hat{X} of the plaintext X and/or his estimate \hat{Z} of the secret key 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 X , S , and, in particular, of 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 private and a public randomizer in our model.

Private randomization is an old cryptographic trick. 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 large 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, private randomization is a useful cryptographic tool. We will see later that the newer cryptographic trick of using a public randomizer can be even more powerful in enhancing the security of a cryptographic system. For these reasons and because their inclusion scarcely complicates Shannon’s theory of secrecy, we have included both types of randomizers in our fig. 1.

It is important to recognize that X , Z , R , and S are *random quantities*. The statistics of the plaintext X are of course determined by the message source, but the statistics of the secret key Z and of the randomizing sequences R and S are under the control of the cryptographer. As fig. 1 suggests, we shall always assume that the random quantities X , Z , R , and S are statistically independent.

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 X for the total of messages that will be enciphered before the secret key Z and the randomizers R and S are changed. Because the enemy crypt-

analyst observes only the cryptogram Y and the public randomizer R , it is appropriate, following Shannon [10], to define *perfect secrecy* to mean that the plaintext X is statistically independent of the pair Y and R , i.e., that

$$P_{X|YR}(x|y,r) = P_X(x)$$

holds for all x , y , and r . This is the same as saying that the enemy cryptanalyst can do no better estimating X with knowledge of Y and R than he could do in the absence of this knowledge, no matter how much time and computing power he has at his disposal. 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 $\{0, 1, \dots, L-1\}$, 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., $P(Z=z)=L^{-M}$ for all L^M possible values z of the secret key, and that the enciphering transformation is

$$(4) \quad Y_i = X_i \oplus Z_i, \quad i=1, 2, \dots, M$$

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 $P(Y=y|X=x)=L^{-M}$ for every possible particular y and x , no matter what the statistics of X may be. Thus X and 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 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.

We have learned recently from a reliable source that the Washington–Moscow hotline is no longer encrypted with a one-time pad, but that in its stead a conventional secret-key cipher that requires much less key is used. This change is apparently the result of increased confidence within the closed cryptographic community in the security of the secret-key ciphers at their disposal.

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, $H(X | Y)$ (which should be read as ‘the uncertainty about X given knowledge of Y ’) is the expectation of the negative logarithm of $P_{X|Y}(X | Y)$, i.e.,

$$H(X | Y) = \sum_{xy \in \text{supp}(P_{XY})} P_{XY}(x, y) (-\log P_{X|Y}(x | y))$$

where $\text{supp}(P_{XY})$ denotes the set of all x, y such that $P_{XY}(x, y) \neq 0$. (The reason that in information theory one takes an expectation by summing only over the *support* of the joint probability distribution of the random variables involved is that this permits one to deal with the expectation of functions such as $-\log P_{X|Y}(x | y)$ that can take on the values $-\infty$ or $+\infty$.) Uncertainties obey intuitively-pleasing rules, such as $H(X, Y) = H(X) + H(Y | X)$, 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

$$(5) \quad H(Y | X, Z, R, S) = 0$$

and

$$(6) \quad H(X | Y, R, Z) = 0$$

respectively, because for instance $H(Y | X, Z, R, S)$ is zero if and only if X, Z, R and S together uniquely determine Y . Shannon’s definition of perfect secrecy can then be written as

$$(7) \quad H(X | Y, R) = H(X)$$

since this equality holds if and only if X is statistically independent of the pair Y and R .

For any secret-key cryptosystem, one has

$$(8) \quad \left\{ \begin{array}{l} H(X | Y, R) \leq H(X, Z | Y, R) \\ \quad = H(Z | Y, R) + H(X | Y, R, Z) \\ \quad = H(Z | Y, R) \\ \quad \leq H(Z) \end{array} \right.$$

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

$$(9) \quad H(Z) \geq H(X).$$

Inequality (9) 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

$$(10) \quad H(Z) \leq \log(L_z^K) = K \log L_z$$

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

$$(11) \quad H(X) \leq M \log L_x$$

(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),

$$(12) \quad K \geq M.$$

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*

$$(13) \quad f(n) = H(Z \mid Y_1, Y_2, \dots, Y_n)$$

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 that $f(n) \approx 0$. 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, \dots, 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

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

where

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

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

$$(16) \quad u \approx \frac{K}{r}.$$

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 $H(X) = M \log L_x = N \log L_y$. 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 Y provide no information at all about X , whereas breaking the system demands that Y determines X essentially uniquely, i.e., that Y must provide the maximum possible information about X . If the secret key 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 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 non-ideal 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. THISISASIMPLFORMOFDATA COMPRESION.

Shannon's derivation of (14) assumed a cryptographic system without the two randomizers that we have included in our fig. 1. When a private randomizer R is included in the system, then $H(X)$ in (15) must be replaced by the joint uncertainty $H(X, R)$ 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.

Homophonic substitution, described in Section B above, is also a method for using a private randomizer to reduce the redundancy r in the cryptogram. Günther [48] quite recently suggested an ingenious variant of homophonic substitution in which the substitutes for a single plaintext letter are binary strings of varying length. Günther showed that it is possible to make the redundancy of the ciphertext *exactly* zero while at the same time making only a modest expansion in the number of binary digits needed to represent the plaintext. Jendahl, Kuhn and Massey [49] modified Günther's scheme to achieve the minimum possible expansion of the plaintext and showed that, on the average, less than four bits of a completely random binary private randomizer suffice to determine the homophonic string for replacing each plaintext letter (whether or not the plaintext alphabet is also binary). What keeps both of these schemes from achieving zero redundancy in practice (and hence from yielding unbreakable practical ciphers) is that both schemes require complete and exact knowledge of the plaintext statistics, something that is never available for real information sources. However, both schemes can make use of available partial knowledge of the plaintext statistics (such as knowledge of the statistics of single letters, pairs of letters, and triplets of letters) to reduce greatly the redundancy r of the cryptogram and hence to increase the unicity distance of an 'ordinary' cipher.

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.

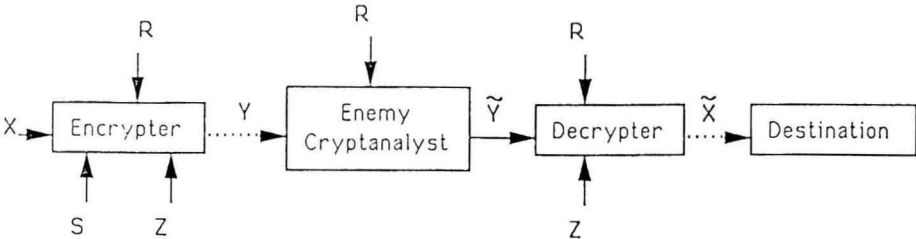


Fig. 2. Modifications to Fig. 1 for consideration of authenticity attacks.

The enemy cryptanalyst is now the one who originates the ‘fraudulent’ cryptogram \tilde{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{Y} as fraudulent and thus not be deceived into passing a fraudulent plaintext \tilde{X} to the destination. The authentic cryptogram Y is shown on a dotted input line to the enemy cryptanalyst in fig. 2 to suggest that the latter may have to form his fraudulent cryptogram \tilde{Y} without ever seeing the authentic cryptogram itself.

As did Shannon, Simmons assumes that the secret key Z will be used only one time, i.e., to form only one authentic cryptogram Y . But Simmons recognized that even in this case, three quite different attacks need to be distinguished. First, the enemy may of necessity form his fraudulent cryptogram \tilde{Y} without knowledge of the authentic cryptogram Y [the *impersonation attack*], indeed Y might not yet exist. The impersonation attack is said to succeed if the decrypter accepts \tilde{Y} as a valid cryptogram – even if it should turn out later that \tilde{Y} coincides with the valid cryptogram Y . The *probability of successful impersonation*, P_I , is defined as the enemy’s probability of success when he employs an optimum impersonation strategy. Second, the enemy cryptanalyst may be able to intercept the authentic cryptogram Y and replace it with his fraudulent cryptogram \tilde{Y} where $\tilde{Y} \neq Y$ [the *substitution attack*]. The substitution attack succeeds if the decrypter accepts \tilde{Y} as a valid cryptogram, and the *probability of successful substitution*, P_S , is defined as the probability of success when the enemy employs an optimum substitution strategy. And third, the enemy may be able to choose freely between an impersonation attack and a substitution attack [the *deception attack*]; the *probability of successful deception*, P_d , is then defined as the probability of success for an optimum deception strategy.

It may appear obvious that $P_d = \max(P_I, P_S)$. Simmons, however, used a game-theoretic authentication model, which was appropriate for the treaty-compliance-and-verification problems that he was considering and in which the cryptographer has the freedom to choose the key statistics to foil the type of attack that the enemy cryptanalyst may choose. In this case, one can only assert that $P_d \geq \max(P_I, P_S)$, since the best choice of key statistics for foiling a deception attack can differ from that for foiling an impersonation attack or for foiling a substitution attack. Our adoption of Kerckhoff’s assumption (see Section B above), however, forces us to assume that the key statistics are fixed once and for all by the cryptographer, independently of the attack used by the enemy cryptanalyst. In this case, which we assume hereafter, it is indeed true that $P_d = \max(P_I, P_S)$.

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 $\# \{Y\}$ denote the number of cryptograms y such that $P(Y=y) \neq 0$, and let $\# \{X\}$ and $\# \{Z\}$ be similarly defined as the number of plaintexts and cryptograms, respectively, with nonzero probability. It follows from (3) that, for every z , there must be at least $\# \{X\}$ different cryptograms y such that $P(Y=y | Z=z) \neq 0$. Hence, if the enemy cryptanalyst in an impersonation attack selects Y completely at random from the $\# \{Y\}$ cryptograms with

nonzero probability, his probability of success will be at least $\#\{X\}/\#\{Y\}$. Thus, P_I , the probability of success in an optimal impersonation attack satisfies

$$(17) \quad P_I \geq \#\{X\}/\#\{Y\}.$$

This shows that good protection against an impersonation attack demands that $\#\{Y\}$ be much greater than $\#\{X\}$, 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 $\#\{X\}$ valid cryptograms y for each key 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 $\#\{Y\} = \#\{X\}$ 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*, $\phi(y, z)$ be defined to be 1 if y is a valid cryptogram for the secret key z and to be 0, otherwise. Note that if $Z=z$, the decrypter will accept $\tilde{Y}=y$ as a valid cryptogram just when $\phi(y, z)=1$. The probability that a particular y is a valid cryptogram can be written

$$(18) \quad P(y \text{ valid}) = \sum_z \phi(y, z) P_Z(z),$$

which is just the total probability of the keys 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

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

In [14], Simmons derived the following fundamental lower bound on the probability of successful impersonation:

$$(20) \quad P_I \geq 2^{-I(Y;Z)},$$

which reveals the quite surprising fact that P_I can be made small *only if the cryptogram gives away much information about the secret key* – at least in principle, exploiting this information is another matter. One of the minor original contributions of our earlier paper [45] was a shortened proof of the bound (20) that allowed one to identify the necessary and sufficient conditions for equality. This simplification motivated Sgarro [50] to provide a still simpler proof of (2) based on properties of ‘informational divergence’. Johannesson and Sgarro then observed that the bound (20) could be strengthened and, in their paper [51] thereon, included an even simpler proof of (20) that was suggested to them by Körner and is based on the ‘log-sum inequality’ [52, p. 48]. This led in turn to our finding yet a new proof of (20) that we now present.

It is immediate from (19) that

$$(21) \quad P_I \geq \sum_y P_Y(y) P(y \text{ valid})$$

with equality if and only if $P(y \text{ valid})$ is constant for all y . Substituting (18) into (21) gives

$$(22) \quad P(y \text{ valid}) \geq \sum_{y,z} P_Y(y) P_Z(z) \phi(y, z).$$

But the pair y and z is in $\text{supp } P_{YZ}$ precisely when $\phi(y, z) = 1$ and $P_Z(z) \neq 0$. Thus, this last inequality can be written equivalently in terms of an expectation as

$$P(y \text{ valid}) \geq E \left[\frac{P_Y(y) P_Z(z)}{P_{YZ}(y, z)} \right]$$

as follows from the discussion of expectations in Section D above. This inequality is of course equivalent to

$$(23) \quad \log P(y \text{ valid}) \geq \log E \left[\frac{P_Y(y) P_Z(z)}{P_{YZ}(y, z)} \right].$$

Because the logarithm is a strictly concave function, Jensen's well-known inequality [15, pp. 151–152] can be applied to give

$$(24) \quad \log E \left[\frac{P_Y(y) P_Z(z)}{P_{YZ}(y, z)} \right] \geq E \left[\log \frac{P_Y(y) P_Z(z)}{P_{YZ}(y, z)} \right]$$

with equality if and only if $(P_Y(y) P_Z(z))/P_{YZ}(y, z)$ is constant for all pairs y and z in $\text{supp } P_{YZ}$. The final step in the derivation of (20) is to note that

$$(25) \quad E \left[\log \frac{P_Y(y) P_Z(z)}{P_{YZ}(y, z)} \right] = H(YZ) - H(Y) - H(Z) = -I(Y; Z).$$

Combining (23)–(25) gives

$$(26) \quad \log P_I \geq -I(Y; Z),$$

which is equivalent to (20). The necessary and sufficient conditions for equality in (20) are seen to be that both (i) $P(y \text{ valid})$ is constant for all y (or, equivalently, that *every impersonation strategy is optimum*) and (ii) that $(P_Y(y) P_Z(z))/P_{YZ}(y, z)$ is constant for all pairs y and z in $\text{supp } P_{YZ}$.

Johannesson and Sgarro obtained a first strengthening of Simmons' bound (20) by noting that although P_I does not depend on the statistics of the plaintext X (as can be seen from (18) and (19)), the mutual information $I(Y; Z)$ generally does depend on P_X . Thus

$$(27) \quad P_I \geq 2^{-\inf 1 I(Y; Z)}$$

where $\inf 1$ here denotes the *infimum* (or 'minimum') of $I(Y; Z)$ over all choices of P_X that leaves the authentication function $\phi(y, z)$ unchanged. They further strengthened this bound by noting that nothing in the derivation of (20) demands that the plaintext X and the key Z be statistically independent (although they always are in our model and in practice) and hence that

$$(28) \quad P_I \geq 2^{-\inf 2 I(Y; Z)}$$

where \inf_2 denotes the infimum of $I(Y;Z)$ over all choices of *conditional* probability distributions for the plaintext X given the key Z .

Because for our Kerckhoffian assumption we have that

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

it follows that (20) gives also *Simmons' lower bound on the probability of successful deception*, namely

$$(30) \quad P_d \geq 2^{-I(Y;Z)}$$

where conditions (i) and (ii) above are necessary, but no longer sufficient, conditions for equality.

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 $I(Y;Z)$ is large, i.e., only when the cryptogram provides the enemy cryptanalyst with much information about the key! The information that Y gives about Z is a measure of how much of the secret key is used to provide authenticity. [It might seem more appropriate to define 'perfect authenticity' to mean that equality holds when the stronger bounds $\inf_1 I(Y;Z)$ or $\inf_2 I(Y;Z)$ are used on the right of (30). However, it seems to us better to abide by Simmons' use of $I(Y;Z)$ and then to consider the case when $\inf_1 I(Y;Z)$ or $\inf_2 I(Y;Z)$ is less than $I(Y;Z)$ as indicating that the authenticity system is 'wasting' part of the information $I(Y;Z)$ that the cryptogram Y betrays about the key Z and thus does not deserve the appellation 'perfect'.]

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 $Y = [Y_1, Y_2]$ is a binary sequence of length 2, the key $Z = [Z_1, \dots, Z_K]$ is a completely-random binary sequence so that $P(Z=z) = 2^{-K}$ for all z , and all logarithms are taken to the base 2 so that $H(Z) = K$ bits.

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

$\begin{array}{c} x \\ \backslash \\ z \end{array}$	0	1
0	00	10
1	01	11

The meaning is that, for instance, $Y=[1,0]$ when $X=1$ and $Z=0$. The en-

ciphering rule is simply $Y = [X, Z]$, 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, $H(Z | Y) = 0$ so that $I(Y; Z) = 1$ 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 $Y = y$, 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 in which the private randomizer S is a binary random variable with $P(S=0) = \frac{1}{2}$.

		x	
		0	1
z	s		
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 $Y = y$ for any y , the two possible values of Z are equally likely so that $H(Z | Y) = 1$, and thus $I(Y; Z) = 0$. 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, $Y = [0, 0]$, 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 $-I(Y; Z)$ 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 s 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 $Y = y$ for any y , there are still two equally likely possibilities for Z so that $H(Z | Y) = 1$ and hence $I(Y; Z) = 1$ bit. But $P(y \text{ valid}) = \frac{1}{2}$ for all four cryptograms y and thus $P_I = \frac{1}{2}$. Moreover, given that he observes $Y = y$, 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 (non-trivial) *perfect authenticity*, no matter what the statistics of the plaintext X may be.

Example 4: Consider the encipherment system.

$\begin{array}{c} z_1 \\ z_2 \end{array} \backslash x$		x	
		0	1
0	0	00	11
0	1	01	10
1	0	10	01
1	1	11	00

Because $P(Y=y | X=x) = \frac{1}{4}$ for all x and y , the system provides *perfect secrecy*. By the now familiar arguments, $I(Y;Z) = H(X)$ and $P_I = \frac{1}{2}$, the corresponding best possible protection against impersonation when $H(X)=1$, i.e., when $P(X=0) = \frac{1}{2}$. But, upon observing $Y=y$, the enemy can always succeed in a substitution attack by choosing Y to be the complement of y . Thus $P_S = 1 = P_d$ and hence this system provides *no protection against deception* by substitution.

Example 5: Consider the encipherment system.

$\begin{array}{c} z_1 \\ z_2 \end{array} \backslash x$		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 $I(Y;Z) = H(X)$ bits. Moreover, $P(y \text{ valid}) = \frac{1}{2}$ for all y so that $P_I = \frac{1}{2}$. Upon observing that $Y=y$, say $y = [0,0]$, 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*.

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 $K \ll H(X)$] 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 known 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 deter-

mined 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 WKL BUT. 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 X , the cryptogram Y and the key Z are binary sequences with lengths $M=64$, $N=64$, and $K=56$, respectively. All 2^{64} possible values of 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 en-

ryption 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 controversy. 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]. Almost fifteen 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 Z , is used only to control a *running-key generator* (RKG) that emits a binary sequence, Z'_1, Z'_2, \dots, 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

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

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

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

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, \dots, 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-

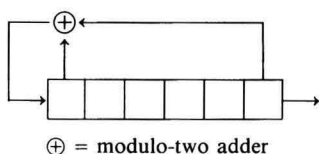


Fig. 3. A 'typical' linear-feedback shift-register.

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 'correlation-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.

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.

The previous words, which appeared in our earlier paper [45], have taken on a prophetic ring. At Eurocrypt'90, Maurer [53] presented a new cipher that exploits a very large public randomizer R , that is provably secure, and that is at least arguably on the verge of being practical. Perhaps the most interesting facet of Maurer's work was his introduction of a new information-theoretic approach to cryptography that allows one to overcome the 'bottleneck' of Shannon's inequality (9) for perfect secrecy. Maurer's trick was to introduce a *security event*, S , with the property that the cipher provides *perfect secrecy given that the event S occurs* [and even if $H(X) \gg H(Z)$] – but 'all bets are off' when S does not occur! For his 'strongly randomized' cipher, Maurer showed that the probability that S does *not* occur will be negligibly small unless the enemy cryptanalyst examines a substantial fraction of all the bits in the very large public randomizer R . The legitimate sender and receiver need examine only the very small portion of the public randomizer that is specified by the short secret key Z . The conclusion from Maurer's work is the (in retrospect obvious) fact that Shannon's bound (9) governs the needed key size only when one demands that his cipher provide *perfect secrecy with probability 1*.

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*

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

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 α ($1 \leq \alpha < p$) is an integer such that $\alpha, \alpha^2, \dots, \alpha^{p-1}$ are, in some order, equal to $1, 2, \dots, p-1$. 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 α is called a *primitive element* of the finite field $GF(p)$, and such α 's are known always to exist.) If $y=\alpha^x$, then it is natural to write

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

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 α 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 , $1 \leq x < p$, would take less than 2000 multiplications (modulo p).

If the discrete exponential function is indeed one-way, then for virtually all

integers y , $1 \leq y < p$, it must be computationally infeasible to compute $\log_x y$. 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 α 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

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

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

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

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 Z was kept secret en route to him, but also that the key 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 $1 \leq i < 6$ 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

$$(37) \quad \phi(pq) = (p - 1)(q - 1).$$

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 $x < n$

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

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

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

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

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

where x is a non-negative 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

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

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

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

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 $\gcd(e, \phi(n))$.

Note that the domain and range of the RSA trap-door one-way function coincide, both are the set of integers from 0 to $m-1$ inclusive. This means that the RSA function can be used to form digital signatures in the manner suggested by Diffie and Hellman. This digital signature capability is one of the most important and useful features of the RSA function.

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

$$(43) \quad de = \phi(n)Q + 1$$

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

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

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 (44) holds for all non-negative 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 computationally impossible to invert this function f_z when one knows only n and e , and also how it is 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 1000 decimal digits each (as RSA suggested thirteen 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 a computational effort that grows asymptotically in the same manner with p and n , respectively. Thus the RSA trap-door function (49) and the Diffie–Hellman function (33) have, as of today, about the same claim to be called ‘one-way.’ For given $n \approx p$, however, the Diffie–Hellman function appears more difficult to invert.

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^{1000} that are primes is about $(\ln 10^{1000})^{-1} \approx \frac{1}{230}$. 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

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

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

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

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 independent 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.

The technique just described leads to the formation of large randomly-chosen 'probable primes' and is the technique currently in widest use for finding the large primes needed with RSA. There is an alternative approach, however, that leads to *sure* primes that are 'probably randomly chosen.' It is not hard to 'grow' large primes with a probabilistic algorithm; the trick is to make the primes appear to be chosen according to a probability distribution that is as uniform as possible over some interval. Maurer [54] has recently given such an algorithm that is very fast (its running time is about the same as for $t = 4$ Fermat tests) and plausibly gives an almost uniform distribution for the selected primes. It would not be surprising should this or similar algorithms eventually replace prime-testing as the method of choice for finding the large primes needed in the RSA public-key cryptosystem or in the Diffie–Hellman public key-distribution system.

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

* The only exceptions are the rare Carmichael numbers that pass Fermat's test for every base b to which they are relatively prime and thus require a strengthened Fermat test for their quick detection as non-primes (which test also speeds up the detection of non-primeness of other numbers), see the paper by van Tilborg in these Proceedings, pp. 41.

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 than 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 trap-door 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 Cook's and Karp's theory of NP-completeness, cf. [34], will lead to provable 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 trap-door-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.

We are grateful to J. Dénes for calling our attention to the fact that the notion of 'one-wayness' is much older than we had suspected. W.S. Jevons, in his book [55] first published in 1873, wrote:

'There are many cases in which we can easily and infallibly do a certain thing but may have much trouble in undoing it. ... Given any two numbers, we may by a simple and infallible process obtain their product, but when a large number is given it is quite another matter to determine its factors. Can the reader say what two numbers multiplied together will produce the number 8 616 460 799? I think it is unlikely that anyone but myself will ever know; for they are *two large prime numbers* (emphasis added).'

Thirty years later, Lehmer [56] announced the 'two large prime numbers' to be 89 681 and 96 079, but added 'I think that the number has been resolved before, but I do not know by whom.' Such anecdotes as that just recounted here serve to feed the suspicions of those who innately mistrust public-key cryptography and who will continue to do so until a provably-secure public-key cipher is produced. But, as we have stressed above, the security of all known practical secret-key ciphers also rests upon conjectures. Neither the secret-key advocate nor the public-key advocate is in a good position to hurl stones at the other.

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 multi-party algorithm, i.e., a specified sequence of actions by which two or more parties cooperatively accomplish some task. Sending a secret message from one user to another in a large network by means of 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 $i = 1, 2, \dots, S$.
- 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, z_1 and z_2 ,

$$(47) \quad E_{z_2}(E_{z_1}(x)) = E_{z_1}(E_{z_2}(x))$$

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 $(x \oplus z_1) \oplus z_2 = (x \oplus z_2) \oplus z_1$, 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, Z_A and Z_B , respectively.
- 2) When user A wishes to send a secret message X to user B , he encrypts X with his own key Z_A and sends the resulting cryptogram $Y_1 = E_{Z_A}(X)$ on the open-but-tamperproof link to user B .
- 3) User B , upon receipt of Y_1 , treats Y_1 as plaintext and encrypts Y_1 with his own key Z_B . He sends the resulting cryptogram $Y_2 = E_{Z_B}(Y_1) = E_{Z_B}(E_{Z_A}(X))$ on the open-but-tamperproof link to user A .
- 4) User A , upon receipt of Y_2 , decrypts Y_2 with his own key Z_A . Because of the commutative property (47), this removes the former encryption by Z_A and results in $Y_3 = E_{Z_B}(X)$. User A then sends Y_3 over the open-but-tamperproof link to user B .
- 5) User B , upon receipt of Y_3 , decrypts Y_3 with his own key Z_B to obtain 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

$$(48) \quad \begin{cases} Y_1 = X \oplus Z_A \\ Y_2 = X \oplus Z_A \oplus Z_B \\ Y_3 = X \oplus Z_B. \end{cases}$$

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

$$Y_1 \oplus Y_2 \oplus Y_3 = X$$

where we have used the fact that two identical quantities sum to **0** 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 '1½ 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

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

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

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

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

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

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

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

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

where α 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

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

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

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

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], [58], [59] on cryptology that will give the reader an orderly development of the

subject. Two recent texts [43], [57] give a broad treatment of the number-theoretic concepts on which much of present-day public-key cryptology depends. The book by Rueppel [44] is a good source of information about stream ciphers.

REFERENCES

1. Diffie, W. and M.E. Hellman – Privacy and authentication: An introduction to cryptography, *Proc. IEEE*, vol. **67**, pp. 397–427, Mar. 1979.
2. Simmons, G.J. – Cryptology, in *Encyclopaedia Britannica*, ed. **16**. Chicago, IL: Encyclopaedia Britannica Inc., 1986, pp. 913–924B.
3. Kahn, D. – *The Codebreakers, The Story of Secret Writing*. New York, NY: MacMillan, 1967.
4. Kahn, D. – *The Codebreakers, The Story of Secret Writing*, abridged ed. New York, NY: New American Library, 1973.
5. Merkle, R.C. and M.E. Hellman – Hiding information and signatures in trapdoor knapsacks, *IEEE Trans. Informat. Theory*, vol. **IT-24**, pp. 525–530, Sept. 1978.
6. Shamir, A. – A polynomial-time algorithm for breaking the basic Merkle–Hellman cryptosystems, *IEEE Trans. Informat. Theory*, vol. **IT-30**, pp. 699–704, Sept. 1984.
7. Newman Jr., D.B. and R.L. Pickholtz – Cryptography in the private sector, *IEEE Commun. Mag.*, vol. **24**, pp. 7–10, Aug. 1986.
8. Vernan, G.S. – Cipher printing telegraph systems for secret wire and radio telegraphic communications, *J. Amer. Inst. Elec. Eng.*, vol. **55**, pp. 109–115, 1926.
9. Hodges, A. – *Alan Turing, The Enigma*. New York, NY: Simon and Schuster, 1983.
10. Shannon, C.E. – Communication theory of secrecy systems, *Bell Syst. Tech. J.*, vol. **28**, pp. 656–715, Oct. 1949.
11. Shannon, C.E. – A mathematical theory of communication, *Bell Syst. Tech. J.*, vol. **27**, pp. 379–423, 623–656, July and Oct. 1948.
12. Diffie, W. and M.E. Hellman – New directions in cryptography, *IEEE Trans. Informat. Theory*, vol. **IT-22**, pp. 644–654, Nov. 1976.
13. Merkle, R.C. – Secure communication over insecure channels, *Comm. ACM*, pp. 294–299, Apr. 1978.
14. Simmons, G.J. – 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. Feller, W. – *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. Morris, R. – The data encryption standard – retrospective and prospects, *IEEE Commun. Mag.*, vol. **16**, pp. 11–14, Nov. 1978.
18. Diffie, W. and M.E. Hellman – Exhaustive cryptanalysis of the NBS data encryption standard, *Computer*, vol. **10**, pp. 74–84, June 1977.
19. Hellman, M. – A cryptanalytic time-memory trade-off, *IEEE Trans. Informat. Theory*, vol. **IT-26**, pp. 401–406, July 1980.
20. Merkle, R.C. and M.E. Hellman – On the security of multiple encryption, *Comm. ACM*, vol. **24**, pp. 465–467, July 1981.
21. Massey, J.L. – Shift-register synthesis and BCH decoding, *IEEE Trans. Informat. Theory*, vol. **IT-15**, pp. 122–127, Jan. 1969.
22. Siegenthaler, T. – Correlation-immunity of nonlinear combining functions for cryptographic applications, *IEEE Trans. Informat. Theory*, vol. **IT-30**, pp. 776–780, Sept. 1984.
23. Massey, J.L. 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. Pohlig, S.C. 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. Rivest, R.L., 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. Knuth, D.E. – *The Art of Computer Programming*, Vol. 1, Fundamental Algorithms. Reading, MA: Addison-Wesley, 1973.
27. Rivest, R.L. – RSA chips (past/present/future), presented at Eurocrypt 84, Paris, France, Apr. 9–11, 1984.
28. Odlyzko, A.M. – 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. Pomerance, C. – 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. Hardy, G.H. and E.M. Wright – *An Introduction to the Theory of Numbers*, ed. 4. London, England: Oxford, 1960.
31. Rabin, M.O. – Probabilistic algorithm for primality testing, *J. Number Theory*, vol. 12, pp. 128–138, 1980.
32. Hardy, G.H. and E.M. Wright – 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. Williams, H.C. – 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. Garey, M.R. and D.S. Johnson – *Computers and Intractability, A Guide to the Theory of NP-Completeness*. New York, NY: Freeman, 1979.
35. Lempel, A. – Cryptology in transition, *Computing Survey*, vol. 11, pp. 285–303, Dec. 1979.
36. Shamir, A., R.L. Rivest and L. Adleman – Mental poker, in *Mathematical Gardener*, D.E. Klarner, Ed. New York, NY: Wadsworth, 1981, pp. 37–43.
37. Chaum, D. – Security without identification: Transaction systems to make big brother obsolete, *Comm. ACM*, vol. 28, pp. 1030–1044, Oct. 1985. See the last contribution of these Proceedings for an updated version.
38. Beker, H. and F. Piper – *Cipher Systems, The Protection of Communications*. New York, NY: Van Nostrand, 1982.
39. Davies, D.W. and W.L. Price – *Security for Computer Networks*. New York, NY: Wiley, 1984.
40. Denning, D.E.R. – *Cryptography and Data Security*. Reading, MA: Addison-Wesley, 1982.
41. Konheim, A.C. – *Cryptography, A Primer*. New York, NY: Wiley, 1981.
42. Meyer, C.H. and S.M. Matyas – *Cryptography: A New Dimension in Computer Data Security*. New York, NY: Wiley, 1982.
43. Kranakis, E. – *Primality and Cryptography*. New York, NY: Wiley, 1986.
44. Rueppel, R. – *Analysis and Design of Stream Ciphers*. New York, NY: Springer, 1986.
45. Massey, J.L. – An introduction to contemporary cryptology, *Proc. IEEE*, vol. 76, pp. 533–549, May 1988.
46. Kruh, L. – Stimson, the black chamber, and the gentlemen's mail quote, *Cryptologia*, vol. XII, p. 65–89, April 1988.
47. Stimson, H.L. and McG. Bundy – *On Active Service in Peace and War*. New York: Harper & Bros., 1947.
48. Günther, C.G. – A universal algorithm for homophonic coding, in *Advances in Cryptology – Eurocrypt'88*, Lecture Notes in Computer Science No. 330. New York and Heidelberg: Springer, 1988.
49. Jendahl, H.N., Y.J.B. Kuhn and J.L. Massey – An information-theoretic treatment of homophonic substitution, to appear in *Advances in Cryptology – Eurocrypt'89*, Lecture Notes in Computer Science. New York and Heidelberg: Springer, 1990.

50. Sgarro, A. – Informational divergence bounds for authentication codes, to appear in *Advances in Cryptology – Eurocrypt’89*, Lecture Notes in Computer Science. New York and Heidelberg: Springer, 1990.
51. Johannesson, R. and A. Sgarro – Strengthening Simmons’ bound on impersonation, submitted to *IEEE Trans. Informat. Theory*, 1990.
52. Csiszár, I. and J. Körner – *Information Theory: Coding Theorems for Discrete Memoryless Systems*. New York: Academic Press, 1981.
53. Maurer, U.M. – A provably-secure strongly-randomized cipher, to appear in *Advances in Cryptology – Eurocrypt’90*, Lecture Notes in Computer Science. New York and Heidelberg: Springer, 1990.
54. Maurer, U.M. – Fast generation of secure RSA-moduli with almost maximum diversity, to appear in *Advances in Cryptology – Eurocrypt’89*, Lecture Notes in Computer Science. New York and Heidelberg: Springer, 1990.
55. Jevons, W.S. – *The Principles of Science* (1st Ed. 1873, 2nd Ed. 1883). New York: Dover, 1958.
56. Lehmer, D.H. – A theorem in the theory of numbers, *Bull. Amer. Math. Soc.*, pp. 501–503, July 1907.
57. Koblitz, N. – *A Course in Number Theory and Cryptography*. New York: Springer, 1987.
58. Tilborg, H.C.A. van – *An Introduction to Cryptology*. Norwell, MA: Kluwer Academic, 1988.
59. Seberry, J. and J. Pieprzyk, *Cryptography: An Introduction to Computer Security*. Englewood Cliffs, NJ: Prentice Hall, 1988.

Public Key Cryptology and Fundamental Research; their Interaction

by Henk C.A. van Tilborg

*Dept. of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven,
the Netherlands*

ABSTRACT

The introduction of public key cryptosystems has led many researchers to look for mathematical transformations that are easy to perform, but very difficult to reverse except when extra information about that transformation is available.

The cryptosystems based on these transformations in turn have led to new research in the underlying mathematical theory.

1. INTRODUCTION

The reader is assumed to be familiar with the preceding paper “Contemporary cryptology: an introduction”, by J.L. Massey [this issue, pages 1–40]. In particular Chapter I “Preliminaries” and Chapter III “Public-key cryptology” are essential for the understanding of this manuscript.

Here, the following three systems will be discussed: the logarithm system, the RSA system and the knapsacksystem. The first two are explained in [7]. From the explanation of the systems it is clear which underlying mathematical theory has been used to create the cryptosystem. The emphasis in this manuscript will be to show that the (potential) use of such cryptosystems has led to a renewed interest in these theories, but, more interesting, also to entirely new questions for the researchers in these fields!

For further reading we refer the interested reader to [3], [11], [12], [14], [15], or the references given in [7].

2. THE LOGARITHM SYSTEM

Diffie and Hellman [2] describe a simple way to establish a secret key (to be used in a conventional cryptosystem) over a public channel. See [7] for the description.

They make use of the fact that it is quite easy [7] to solve e.g.

$$2^{78} \equiv c \text{ modulo } 83,$$

but very elaborate to solve

$$2^m \equiv 34 \text{ modulo } 83.$$

The first problem is that of **exponentiation**, the second is that of **taking logarithms**, since it can be rewritten as

$$m = \log_2 34.$$

The number 2 above is called the **base** of the logarithm.

In general, when the calculations are done modulo p , one needs at most $2 \log_2 p$ multiplications for a single exponentiation, while about \sqrt{p} of such operations are needed to find one logarithm. Writing q for $\log_2 p$, i.e. $p = 2^q$, one sees that the complexity of one exponentiation is $2q$, while this is $2^{q/2}$ for one logarithm; one grows linearly in q , the other exponentially!

In table 1 the discrepancy in growth between the complexity of a single exponentiation (so $2 \log_2 p$) and the standard way of taking a logarithm (so \sqrt{p}) is demonstrated. The last column will be explained later.

Two years later already, Pohlig and Hellman [10] showed that taking logarithms can be done very efficiently if $p-1$ happens to have only small prime factors. It is based on a theorem by Fermat.

Fermat, 1601-1665

If p is a prime number and $1 \leq a < p$, then

$$a^{p-1} \equiv 1 \text{ modulo } p$$

An extreme case of this method is given by $p = 257$ (with base $a = 3$ instead of 2). Indeed $p-1 = 2^8$, so $p-1$ has 2 as only prime factor.

Table 1. The complexity of exponentiation versus taking logarithms.

number of digits in p	computing an exponent operations	taking a logarithm		
		standard method		Adleman operations
		operations	memory	
2	12	10	10	201
10	66	10^5	10^5	$2.4 \cdot 10^7$
20	132	10^{10}	10^{10}	$3.4 \cdot 10^{11}$
50	332	10^{25}	10^{25}	$2.0 \cdot 10^{20}$
100	664	10^{50}	10^{50}	$5.5 \cdot 10^{30}$

Due to the choice of the primitive element $\alpha (= 3)$, not only $3^{256} \equiv 1$ modulo 257 (by Fermat), but also $3^{128} \equiv -1$ modulo 257 because apart from 1 only -1 will have 1 as its square.

Example

To solve $3^m \equiv 75$ modulo 257, write the unknown m in its binary representation:

$$m = m_0 + m_1 2 + m_2 4 + m_3 8 + \cdots + m_7 128,$$

where the m_i 's are zero or one.

Now modulo 257:

$$\begin{aligned} 75^{128} \equiv 3^{m_{128}} &\equiv 3^{(m_0 + m_1 2 + \cdots + m_7 128)128} \\ &\equiv 3^{m_0 128} \equiv \begin{cases} +1 & \text{if } m_0 = 0, \\ -1 & \text{if } m_0 = 1. \end{cases} \end{aligned}$$

So we compute 75^{128} and find as result -1 . The conclusion is that $m_0 = 1$. To find m_1 we divide 3^m (i.e. 75) by 3^{m_0} (so by 3) and obtain 3^{m-m_0} , which is 25. Similarly to above

$$\begin{aligned} 25^{64} \equiv 3^{(m-m_0)64} &\equiv 3^{(m_1 2 + m_2 4 + \cdots + m_7 128)64} \\ &\equiv 3^{m_1 128} \equiv \begin{cases} +1 & \text{if } m_1 = 0, \\ -1 & \text{if } m_1 = 1. \end{cases} \end{aligned}$$

Computing 25^{64} modulo 257 gives -1 , so $m_1 = 1$. Continuing in this way one finds: $m_2 = 1$, $m_3 = 1$, $m_4 = 0$, $m_5 = 1$, $m_6 = 1$ and $m_7 = 0$. The conclusion is that $m = 1 + 2 + 4 + 8 + 32 + 64 = 111$ is the solution of $3^m \equiv 75$ modulo 257.

Note that in this method the factorisation of $p-1$ played an important rôle! If $p-1$ does have a large prime factor, in particular if $p-1$ is two times another prime, the Pohlig-Hellman method reduces to the \sqrt{p} method, mentioned earlier.

Of course the probability that $p-1$ factors into small primes is rather small. Much effort has been undertaken to find faster algorithms to determine a logarithm. However in the context of cryptology, other arguments play a rôle than in a regular mathematical context. For instance, if an enemy can decrypt ciphertexts with some nontrivial probability (but not always), the system is still considered useless. For the same reason, the maximum running time of an algorithm that breaks a cryptosystem is not the right measure of its usefulness, but the average running time is!

In [1] a method is described that finds the logarithm with average running time:

$$e^{c\sqrt{\ln p \ln \ln p}},$$

where c is a constant. In the last column of table 1 the growth of this function is illustrated for $c=2$. In view of this, it is recommended to take p at least 100 digits long.

3. THE RSA SYSTEM

The RSA-system is also explained in [7]. It is based on the difficulty of factoring large numbers.

In this context, two obvious questions are left for the researcher:

1. How to generate large prime numbers (to be used in RSA).
2. How to factor large numbers (to break RSA).

As explained in [7] an odd, 100-digit number that is randomly selected has about 1% chance of being prime. So, if a fast primality test is available, one expects to need it about 100 times to find a prime number from randomly selected, odd, 100-digit numbers.

Again there are two ways to proceed.

- **Deterministic** primality tests. They will find out if a candidate number is a prime or not.
- **Probabilistic** primality tests: The probability that a composite number does survive the test can be made very small.

Inspired by the potential applications of the RSA system, researchers looked for better tests of both kinds. Probabilistic primality tests are orders of magnitude faster than deterministic tests. So, if one really wants absolute certainty that a number is prime, one first applies a fast probabilistic primality test. In this way one saves the slower, deterministic test for the few promising candidates, which have passed the other test.

The mathematical problem with probabilistic primality tests is to say something about their effectiveness. In other words, one wants the probability that a composite number is not detected as such by one run of the test to be bounded from above by a certain number less than one, say b . The smaller b the better of course. The probability that k runs of the test (each with its own starting value) do not detect that a number is composite, while in fact it is, is at most b^k . By taking k sufficiently large, this probability can be made arbitrarily small.

A word of warning is in place here. If in the Rabin test, mentioned in [7], one wants to test the primality of the number q , one may think that it suffices to take (k times) a random number, say a with $1 \leq a < q$, and to check whether

$$a^{q-1} \equiv 1 \text{ modulo } q$$

(as it should be according to Fermat's Theorem when q is indeed a prime).

However there exist numbers q such that for **each** a with $\gcd(a, q) = 1$ the relation above holds and still they are not prime. (Here \gcd stands for greatest common divisor.) Such numbers q are called **Carmichael** numbers.

The smallest one is $561 = 3 \cdot 11 \cdot 17$. The actual test in Rabin's method computes $a^{(q-1)/2^i}$ modulo q for $i = 0, 1, 2, \dots$ provided that $a^{(q-1)/2^i}$ is an integer, until the value is not equal to 1. If the first value is not 1 modulo q or the first

value in the sequence different from 1 is not -1 modulo q , then q is composite. Otherwise q passes the test for a . Rabin proved that a non-prime q fails the test for at least $3/4$ of the possible values of a . So the value of b , mentioned above, is $3/4$ in the Rabin test.

Deterministic primality tests are much more complicated and involve deep results in number theory. In [5] the reader can find a clear explanation of the primality test by H. Cohen and H.W. Lenstra jr.

If the complexity of the deterministic algorithms further decreases in the future, they will certainly find more applications.

The world of the fast factorization algorithms is much more alive. For years one has been hearing of new records being broken. In 1978 the designers of the RSA system, Rivest, Shamir and Adleman, suggested a modulus n about 200 digits long. At that time, the fastest factorization algorithm [9] had complexity

$$e^{\sqrt{\ln n \ln \ln n}}.$$

For practical implementations it is convenient to have a modulus length equal to a power of a power of 2. Now $2^{512} \approx 10^{154}$, while $2^{1024} \approx 10^{308}$. Based on the complexity formula above, designers were convinced that 154 as modulus length was safe enough (giving $512 = 2^9$ as exponent of 2). Not any more! First, existing algorithms, in particular the so-called *quadratic sieve* method, were generalized and enhanced by researchers. Besides, in 1985, H.W. Lenstra jr. e.a. [6] introduced and improved the so called *elliptic curve factoring algorithm*. The latter method is better than the former when the number to be tested has a (relatively) small prime divisor. For the modulus n of the RSA system, this is not the case. The factorization of any composite 80, 90 and even 100 digit number is now possible.

Quite recently (in 1990) A.K. Lenstra, H.W. Lenstra jr., M.S. Manasse and J.M. Pollard have developed a new method, which is called the *number field sieve*. Right now it is only suitable to factor numbers of the form $r^e \pm s$, with r and s small, but researchers are trying to adapt the algorithm in such a way that it can factor arbitrary numbers of that length.

Numbers completely factored by them are:

number $r^e \pm s$	# digits	computing time in days
$3^{239} - 1$	107	16
$2^{373} + 1$	108	7
$7^{149} + 1$	122	19
$2^{457} + 1$	138	49

In the middle of 1990 Arjen Lenstra and Mark Manasse orchestrated (from Palo Alto) the factorization of $2^{512} + 1$, which is a **148** digits long number! They did not do this by themselves. In fact, after a long precomputation their computer communicated with hundreds of computers of colleagues all over the world. Each of these computers performed part of the necessary calculations (using the quiet hours).

This length of 148 digits is so close to the modulus length of 154 digits (= 512 bits), that one should not consider that modulus to be absolutely safe anymore. One may argue that the above algorithm only works for moduli of a very special form, but on the other hand this method is so new, that it is not clear at all at this moment to which extent it can be adapted to handle other moduli.

It seems wiser to use the 200 digit numbers proposed originally.

4. THE KNAPSACK SYSTEM

Merkle and Hellman [8] base their system on the knapsack problem, explained in the following example.

Example

Suppose one has a knapsack that can carry exactly 7750 grams. Can this knapsack be filled up to capacity by making the appropriate choice from the following items.

Items	Weight
butter	250
eggs	450
meat	1000
bread	2000
fruits	4500
drinks	9000

The answer is: **YES**. It is even easy to find the solution, because the weights 250, 450, 1000, 2000, 4500, 9000 form a **superincreasing sequence**, which means that each weight is greater than all the previous ones together! Indeed, one obviously should not take the drinks, because their weight exceeds the 7750 limit.

Next, one needs to take the fruits of weight 4500, because the remaining items have a total weight $250 + 450 + 1000 + 2000$ which is less than 4500 (by the super-increasing property), which in turn is less than or equal to 7750. The remaining weight to be filled is $7750 - 4500 = 3250$.

So, take the bread of weight 2000, because $250 + 450 + 1000 < 2000 \leq 3250$. The remaining weight is: $3250 - 2000 = 1250$.

Similarly, take the meat of weight 1000, because $250 + 450 < 1000 \leq 1250$. The remaining weight is: $1250 - 1000 = 250$.

Do not take the eggs of weight 450, because $450 > 250$.

Do take the butter of weight 250, because $0 < 250$. The remaining weight is $250 - 250 = 0$. This means that the knapsack is filled up to its capacity of 7750 by choosing fruits, bread, meat and butter (of weights 4500, 2000, 1000 and 250 respectively).

In general the knapsack problem is very difficult to solve as may be illustrated by the following example (of the same small length).

Example

Let the capacity of a knapsack be 101077 grams and let six items have their weight given by

Item	Weight
1	44434
2	19714
3	56639
4	31669
5	44927
6	36929

All these items have roughly the same weight!

The only way to solve the knapsack problem in general is essentially to try all possibilities: take the first item or not, take the second item or not, etc.

In this way, the computing time grows **exponentially** in the number of items! Merkle and Hellman based a cryptosystem on the above observations:

1. Start with superincreasing knapsack of sufficient length (say 100 instead of 6), but keep it secret.
2. Transform it into a “difficult” looking knapsack and make this public.
3. Others use this for encryption in a way that will be described below.
4. The legitimate user can transform it back to the “easy” superincreasing knapsack.

We shall illustrate their technique with the following example.

Example

The (future) receiver of a message, called Bob for convenience, chooses as superincreasing knapsack the numbers

22, 89, 345, 987, 4567, 45678.

Bob multiplies these numbers with 12345 and reduces the results modulo 56789. For instance $22 \times 12345 \equiv 271590 \equiv 44434 + 4 \times 56789 \equiv 44434$ modulo 56789. In this way Bob gets the “difficult” knapsack

44434, 19714, 56639, 31669, 44927, 36929

and makes it public.

If somebody else, say Ann, wants to send a message in secret to Bob, she rewrites the message as binary sequences of length 6. One such sequence is for instance 1, 1, 0, 0, 0, 1. The ones tell which terms in the knapsack have to be added.

So Ann computes the *sum* of

$$\begin{array}{rcccccc} 44434 & 19714 & 56639 & 31669 & 44927 & 36929 \\ \times & \times & \times & \times & \times & \times \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}$$

and sends the result, i.e. **101077**, to Bob.

A third party cannot recover 1, 1, 0, 0, 0, 1 from the publicly known knapsack and the sum 101077 (at least if the length of the knapsack was 100 in stead of 6).

Bob has the same problem, but knows that the difficult knapsack came from multiplying the easy knapsack with 12345 modulo 56789.

Bob also knows that

$$12345 \times 39750 \equiv 1 \text{ modulo } 56789.$$

So, if one multiplies a number (less than 56789) by 12345 modulo 56789 and the result is multiplied by 39750 modulo 56789, one gets the original number back!

So instead of solving

$$\begin{array}{rcccccc} 44434 & 19714 & 56639 & 31669 & 44927 & 36929 \\ \times & \times & \times & \times & \times & \times \\ ? & ? & ? & ? & ? & ? \end{array} = 101077.$$

Bob multiplies all these numbers by 39750 modulo 56789. In particular $39750 \times 101077 = 45789$ modulo 56789. So Bob has to solve the original “easy” knapsack problem (with the superincreasing sequence):

$$\begin{array}{rcccccc} 22 & 89 & 345 & 987 & 4567 & 45678 \\ \times & \times & \times & \times & \times & \times \\ ? & ? & ? & ? & ? & ? \end{array} = 45789$$

The solution is indeed 1, 1, 0, 0, 0, 1.

The knapsack cryptosystem is very easy to implement, because only a few additions, multiplications and divisions are involved in the calculations.

As a result of its cryptographic application many researchers looked again at the knapsack problem, but now with different eyes!

Very soon one realized that there are more mappings that transfer the “difficult” knapsack back into a superincreasing sequence. It should be noted that one does not have to find the original superincreasing sequence back for decryption. Anyone will do.

Making use of this observation, Shamir [13] in 1982 broke the simple knapsack system, explained above, but this attack left an iterated version intact.

In 1983 Lagarias and Odlyzko [4] came with a completely new idea: an algorithm that solves a non-negligible percentage of all knapsacks, not just the knapsacks constructed from superincreasing sequences! There is a condition that the knapsack should satisfy, but later it turned out that knapsacks that do not satisfy this condition are not safe for a different reason! The result by Lagarias and Odlyzko is quite surprising, because the general knapsack prob-

lem is known to belong to a class of difficult problems. It still is, but apparently not because almost all specific knapsacks are difficult to solve, but just some.

For the reasons explained above, the knapsack system is no longer interesting for cryptographical purposes.

5. CONCLUSION

Three public key cryptosystems have been discussed, each based on a mathematical operation that is easy to perform, but in general difficult to undo, except when additional information is available.

The potential value that these systems had for cryptographic applications gave an enormous impulse to research in the underlying mathematical theory.

In all cases this led to new and interesting results. Especially, completely new algorithms have been found that have a probabilistic running time or that succeed with some nontrivial probability.

REFERENCES

1. Adleman, L.M. – A subexponential algorithm for the discrete logarithm problem with applications to cryptography, Proc. 20-th Annual *IEEE Symp. on Found. of Comp. Science*, p. 55–60, 1979.
2. Diffie, W. and M.E. Hellman – New directions in cryptography, *IEEE Trans. Inf. Theory*, IT-22, p. 644–654, 1976.
3. Koblitz, N. – A Course in Number Theory and Cryptography, Graduate Texts in Mathematics 14, Springer-Verlag, New York, 1987.
4. Lagarias, J.C. and A.M. Odlyzko – Solving low density subset problems, Proc. 24-th Annual *IEEE Symp. on Found. of Comp. Science*, p. 1–10, 1983.
5. Lenstra, H.W. jr. – Fast prime number tests, *Nieuw Archief voor Wiskunde* (4), 1, p. 133–144, 1983.
6. Lenstra, H.W. jr. – Factoring integers with elliptic curves, Report 86-16, Dept. of Mathematics, University of Amsterdam, the Netherlands 1986.
7. Massey, J.L. – Contemporary Cryptology: An Introduction, Proc. Cryptography and Data Protection, Verh. Afd. Natuurkunde, Eerste Reeks, deel 37, p. 1–40.
8. Merkle, R.C. and M.E. Hellman – Hiding information and signatures in trapdoor knapsacks, *IEEE Trans. Inf. Theory*, IT-24, p. 525–530, 1978.
9. Morrison, M.A. and J. Brillhart – A method of factoring and the factorization of F_7 , *Math. Comp.* 29, p. 183–205, 1975.
10. Pohlig, S.C. and M.E. Hellman – An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, *IEEE Trans. Inf. Theory*, IT-24, p. 106–110, 1978.
11. Salomaa, A. – Public-key Cryptography, *EATCS Monographs on Theoretical Computer Science* 23, Springer-Verlag, Berlin, 1990.
12. Seberry, J. and J. Pieprzyk – Cryptography: an Introduction to Computer Security, *Advances in Computer Science Series*, Prentice Hall, New York, 1989.
13. Shamir, A. – A polynomial time algorithm for breaking the basic Merkle–Hellman cryptosystem, in Proc. 23-rd *IEEE Symp. Found. Computer Sci.*, p. 145–152, 1982.
14. Van Tilborg, H.C.A. – An Introduction to Cryptology, Kluwer Academic Publishers, Boston, 1988.
15. Welsh, D. – Codes and Cryptography, Oxford Science Publications, Clarendon Press, Oxford, 1988.

Protection of Medical Data

by J.H. van Bommel

Department of Medical Informatics, Erasmus University Rotterdam, the Netherlands

1. INTRODUCTION

There is hardly any area of health care where the computer is not yet visible: in primary care, pharmacies, clinical laboratories, outpatient clinics, departments in hospitals – the influence of computers is seen everywhere. One of the applications that are most frequently installed is the storage of patient-related data for easy and fast retrieval at any place and time for which a user is authorized.

Medical data play a key role for diagnosis, therapy, prognosis, prevention, research, and management. The transport of such data is nowadays also supported by electronic means (*EDI*, electronic data interchange). Modern health care is no longer feasible without computers for data storage and retrieval, electronic transportation, processing, and interpretation.

Databases

One of the problems we are confronted with today, is whether this electronic storage, transmission, and processing of medical data should be further stimulated or whether it should be limited because of potential disadvantages and negative aspects, such as possible misuse of the data. Such considerations do, incidentally, equally apply to data storage in computers of the police, the civil service, or for fiscal purposes. In all areas it should be carefully considered whether the advantages of electronic data storage are in balance with present and future drawbacks. In any case, when using database systems for medical

purposes we should realize that the developments in information technology will proceed at an increasingly faster pace.

Relationships

Computers may assist in assessing the relationships between medical data and diseases. There exists a wide variety of medical data (pictures, biological signals, alphanumeric data); some of these data are dynamic and time-dependent, but others are permanent or stationary and remain with us for the rest of our life, such as gender, blood group, allergies and our genetic profile. For example, the relationship between genetic data and many diseases or handicaps is gradually becoming more evident because of progress in the field of molecular biology and genetics. Genetic data are not only of interest for individuals, but possibly for their relatives as well (see, e.g. [1,2]).

Different requirements are simultaneously fulfilled with the structured storage of medical data:

- the patient or client requires medical advice;
- the care provider wishes to control the process of medical care;
- the researcher wants to obtain more insight;
- insurance companies need to calculate risks;
- the government is interested in long-term planning and the setting of priorities;
- employers want to use data for planning and to be safeguarded against financial claims.

Since stationary medical data (read, e.g.: genetic data) are of increasing importance and have large impacts on privacy and data protection, we will pay special attention to this category. First of all, we will deal with the specific nature of medical data, including those of genetic origin. Subsequently we will deal with the purposes of computer storage of medical data. In doing so, we will consider some consequences for data protection.

2. MEDICAL DATA

Humans have the ability to discern new situations and are capable of new and creative tasks. The computer does not possess these faculties, but has complementary properties that supplement human shortcomings such as a finite memory, and slow and inaccurate data processing, e.g., calculating. However, the computer causes a twofold reduction of reality: computer processing should be highly structured and formalized, and computer-stored data can only be expressed in symbolic form, i.e., characters and numbers which are subsequently coded in binary form. Computers do not know how to handle singular events or individual persons, or issues that cannot be quantified or coded. Computers also have no capability for feelings, concepts, or intentions. All this has major consequences for the structuring of processes in medicine and the storage of medical data.

Use of Medical Data

The acquisition and processing of data for the solution of some technical problem may be complicated, but is in principle feasible. Technical processes can often be modelled and described in structural terms, and the data to be derived from such processes can often be expressed quantitatively. In several other areas, such as medicine, this is different. For instance, processes that deal with health and disease can only be partly described in a formal manner, and far from all data can be expressed in quantitative terms. The patient's disease is generally highly individual and unique, and in many cases the treating physician does not only rely on 'hard' facts such as, e.g., laboratory analysis or results from organ function analysis, but also on 'on soft' data from the patient history. When treating the patient there frequently is no fully formalized strategy; diagnostic findings and therapeutic possibilities have to be carefully balanced with the prognosis, life expectancy, possible risks, patient reaction, acceptance by relatives, etc.

In using medical data for patient care we have to consider one further aspect: very often there is no one-to-one relationship between medical data and diseases. This is partly due to the limited formal description of medical processes and quantifiability of medical data, but is also caused by the large variability of all biological processes and the fact that all required data are often not fully available. These limitations lead to inaccuracies and uncertainties in the diagnosis and even to inevitable errors, denoted as false positive (FP, the disease is positively concluded but in reality not present) and false negative errors (FN, the disease is present, but not diagnosed). Errors of these types may also be caused by incomplete knowledge or improper use of diagnostic methods.

Types of Medical Data

Medical data can be categorized into different types, but this treatise will focus only on classification into permanent and variable data. The first category is perhaps the most privacy-prone. Use of these two types of data is generally also different: variable medical data (to which belong alpha-numeric data, biological signals, and pictures) are primarily used for the diagnosis and treatment of 'transient' diseases, whereas permanent, e.g., genetic data are often strongly related to one's life, possibly far in the future, to the prognosis. The last category is, as remarked earlier, also of interest for one's next of kin: parents, children, brothers, sisters. For instance, if someone in a family appears to have a genetically determined disease, then different relatives may also be carriers of the disease c.q. of the abnormal gene, either in a dominant or a recessive form.

Because of its importance, the circle of people interested in genetic data is at least as large as that for variable medical data. Genetic data do not change or age; they are of interest for people during their entire lifetime. Knowing one's own genetic profile and risks could possibly lead to a certain lifestyle; it

can also result in unbearable psychological suffering so that not knowing is sometimes a preferable option.

Data in Computers

What makes storage of medical data in computers so special? This is related to the nature of the computer; the fact that processes have to be formally structured and data have to be coded or quantitatively described. This indicates both the power and the limitation of the computer. All data have to be described in symbolic form, but in real life, including medical care, not all observations can be expressed in such form (think of patient feelings and expectations). The medical record as expressed in a computer, therefore, represents only a limited view of the patient's disease. If someone other than the treating physician uses only computer-stored patient data, he would not always be able to obtain a complete and reliable picture of the patient's complaints and observations. However, for many reasons, this often also holds for the written medical record. Both the patient and the physician have to be protected against an improper, let alone illegal, use of computer-stored medical data.

There is another very important difference between written and computer-stored medical data. Essentially, written data only serve direct patient care whereas computerized data can furthermore be used for epidemiology, medical research, the evaluation of medical care, or education. Computers enable us to use medical data to investigate, for instance, the relations between symptoms and diseases, or the effect of different therapies on patient outcome.

3. USERS OF MEDICAL DATA

Different groups of users are interested in medical data. Since especially the permanent medical data are particularly privacy-prone, we will investigate which people are interested in such data (e.g., [3,4]). Consecutively we will deal with the following groups of interested users: the patient or client and their relatives (parents, children, brothers, sisters), the treating physician (general practitioner (GP), specialist), the medical researcher, insurance companies (pension funds, health or life insurance companies), employers, and the government (e.g., the Ministries of Health or Justice).

The Patient or Client

Now that there is an exponentially increasing amount of medical data in computers it is of utmost importance for patients and physicians to have such data stored as reliably and as completely as possible. A complicating factor is that data from the same patient are often stored in many different databases for different purposes: general practice, clinical use, occupational medicine, usage by insurance companies, etc. These different views on the same patient may cause conflicts of interest and have impact on the patient's privacy.

It is not always desirable, or even permissible, that all care providers have access to these different databases, even when they intend to use the data solely

for the patient under treatment. For example, the patient will not appreciate when a radiologist has access to or is informed of the fact that once in the past he consulted a psychiatrist, or when a medical examiner has knowledge of his entire medical record. Even the patient himself may not always want full knowledge when, on the basis of such data, the possibility of getting some future disease might be predicted. A conflict of interest of a different type may arise when, in contrast, a relative is highly interested in such data or, the reverse, when some relative is informed about hereditary diseases that someone else in the family prefers not to know.

The patient should be able to understand as fully as possible the implications of a request from a physician to allow storage of his medical data – either permanent or variable – in a computer. He should maintain the right to freely decide his response and, based on some expert advice, to know what are the consequences of such a request [5]. It should also be made clear to the patient what the implications of such a request would entail for his relatives. It is a complex ethical and legal problem whether the patient has the right to prohibit this physician from informing his relatives in the case that his medical data might have major consequences for his next of kin. Protection of the privacy of an individual may, in some circumstances, hamper the interest of others. This consideration could even be extrapolated to society as a whole.

From the foregoing it becomes evident that, especially genetic data, are of wider interest than for the individual alone. In fact this aspect has not just become apparent due to the recent progress in molecular biology and human genetics. For many generations it was already known that problems such as diabetes, cardiac diseases or certain allergies were associated with certain families [6] and that genetic properties, once acquired, are transmitted to the offspring. But nowadays, with computer storage of medical data, a much wider field for proper use, as well as abuse, has been disclosed. For that reason, all members of the same family are interested in the protection of genetic data of one member against unlawful use. When someone puts his genetic data at the disposal of an insurance company this may also have consequences for his relatives.

The Treating Physician

It is not yet clear whether the treating physician – and especially the GP – should literally be the key person to open medical data to third persons, preferably in close agreement with the patient. Apart from that it is important that someone other than the patient himself stands up for his interest and guards access to the different medical databases in which the data are stored. In case the GP has knowledge of the risk someone is running based on his or her medical or genetic data, the GP may be confronted with the dilemma of whether to inform the person concerned, especially when the patient has not requested such information. It is even more difficult to decide whether that person's relatives should be informed. This decision becomes harder as the level of risk increases and the consequences become more far-reaching. Both the

passing on and withholding of information has legal and moral consequences for the physician, which requires that his mode of conduct should be regulated.

The Scientific Researcher

Both medical care and medical research make use of the same collected patient data. Such research is inconceivable without access to computer-stored medical data that should, preferably, be made anonymous. At the same time, however, such data should be properly documented, for instance with related diagnoses and it should be possible to collect such data over time so that trends and relationships can be investigated. The 'pooling' of data concerning rare diseases can only be accomplished if data are stored in large databases, in order to obtain further knowledge about the prognosis of diseases. This implies that it should in principle be possible to trace certain patients, which could prove to be in sheer contrast to the protection of privacy.

Also, early recognition of changes in the disease profiles of the entire population can only take place when data are collected in large databases and are analyzed by epidemiologists. If this had happened at the time of the so-called Softenon case, in principle this disease would have been discovered one year earlier. No wonder that computers are today the necessary tools for medical researchers.

Insurance Companies

Pension funds, health or life insurance companies, and sickness funds are, understandably, highly interested in all data that concern the person or patient for which they have to calculate the future risks. It is understandable that insurance companies might decide to increase premiums, or even refuse to cover some risk, if on the basis of the medical data the risk appears to be too high and/or the patient has an unfavorable life expectancy. On the other hand, the patient will do his utmost to obtain an insurance that is as beneficial as possible if only he, and not the examining physician, has knowledge of his future risks, e.g., perhaps based on his knowledge of hereditary diseases within his family.

If insurance companies would calculate the risk factor also taking into consideration genetic data, this could imply high expenses for certain groups of the population, or even their total exclusion from health insurance. In the latter cases legislators should provide the rules how to handle these circumstances so that access to genetic data should not imply the end to solidarity in health care, where healthy people carry part of the burden for the less fortunate ones [3,7]. Therefore, if genetic data would become accessible to insurance companies, there is a real danger for a new type of undesirable discrimination, i.e., against those people or families who have a genetic profile with an increased genetic risk [4,6,8].

To preclude such developments in society, it should be stipulated by law that certain medical data are not accessible to others than the physician who has the full confidence of the patient, and that the insurance premiums should not be related to genetic risks.

The Employer

Following in the wake of the insurance companies, employers are also interested in the medical data of their (future) employees. Here, however, matters are more complicated, since some types of work entail a higher risk for people with specific medical conditions (e.g., the combination of chemicals or dust with certain allergies or lung diseases), which could be detrimental to the health of the employee. This could lead to earlier onset of disease or unfitness for work resulting in financial disadvantage for the employer. But also third persons could become involved in such risks, for example airline pilots who have an established risk for cardiac diseases. The remark on solidarity is of relevance in such cases as well, but it should be realized that nobody should be challenged to undertake a higher risk if it can be avoided by choosing some other profession or employer. Also here, the physician examining for fitness for some type of work should be obliged to follow legal regulations regarding his access to medical data. The discussions around HIV and genetic data are illustrative for the problems in this respect.

The Government

Societies and governments are nowadays confronted with steadily increasing costs of health care, which amount to 9% (The Netherlands) or even over 12% (U.S.A.) of the GNP. For that reason there is a tendency to decrease the number of patient-days in hospitals or nursing homes, in some instances leading to the closure of entire hospitals, and to stimulate primary care and home care. Foremost, governments prefer to stimulate prevention and health care planning and to determine priorities in health care. For those reasons governmental authorities are highly interested in the prevalence and prevention of genetically determined diseases. It should be realized that there is a long distance, but also a gradual transition between interest for reasons of prevention and measures based on eugenic intentions.

4. PRIVACY

Privacy implies several different issues. For instance, it means the right to be left alone, but it also signifies that everyone is entitled to decide for himself how, when and to what degree others may dispose of his (medical) data. In many countries, this right has been described in the law; in The Netherlands, this has even been laid down in the Constitution [9], in Europe it has been anchored in the Treaty for the Protection of Human Rights and Fundamental Freedom [10].

The Dutch Constitution specifies that all persons have 'the right that their privacy shall be respected' (article 1), also 'related to the recording and the provision of personal data' (article 2). 'The Law regulates the rights of persons regarding the cognizance of data that have been recorded about them and their usage, together with the correction of such data' (article 3). It may be evident from this that the privacy of a patient also concerns his bodily integrity, which is described in article 11.

Essentially, the privacy of the patient is guaranteed by the professional secrecy of the physician, which is at the same time a right of the patient. A patient should be able to transfer all his medical information to his physician, without fearing that the physician will pass these data to third parties without the patient's approval. This professional secrecy has been regulated in several articles of the Law in The Netherlands.

If the physician wants to fulfill his responsibility to guard patient data, he should take measures that data are well protected. This entails measures against loss, theft, or damage; against (unintended) abuse and/or false interpretations; also against intended use or misuse. The latter also includes that medical data would be unjustly used for purposes other than for which they were collected, without the provider of the data knowing this. To accomplish this, proper scientific (syntactic and semantic), technical (e.g., spatial safeguarding against damage or fire), software (such as passwords, auditing trails, confined functionality, encryption), and hardware measures (e.g., back-ups and double installation of essential parts such as disks) are required.

Because of the fact that modern health care provision very often requires teamwork instead of care by a single physician, and as a consequence of information technology, the individual physician is no longer capable to personally guarantee the patient's privacy. For that reason, after the regulation of the professional secrecy, modern society has also legally laid down the right to privacy. This means that for all automated registrations of personal data written regulations are required, to be supervised by a Privacy Committee. These regulations should contain descriptions of the purpose of the registration, the disposal of data to third parties, the right of all persons concerned to inspection, alteration and destruction of their data. In principle, these regulations do not concern anonymous data.

The sensitivity of medical data to privacy is very much dependent on the context in which they are used. For instance, psychiatric data are often indicated as being highly sensitive to privacy. Nevertheless, there are data on many other diseases that could damage someone's career and that are also privacy-prone. The mere fact that it is known that someone once had a medical consultation in a psychiatric clinic or was hospitalized in a cardiological department, might influence decisions of employers or insurance companies. Similar consequences apply to the use of certain drugs, documented in someone's medical record.

Regretfully, the present privacy regulations in The Netherlands offer few guarantees that take into account the special character of genetic data that are of interest for more people other than the single person about whom they were recorded; perhaps being relevant for different generations of families. For that reason, in all systems in which genetic data are to be stored, the purpose of the data collection should be properly described; outside this scope it should be forbidden that such data are used. The same applies to coupling of different databases. Preferably, the data should be stored anonymously and the key to the data should be in the hands of a physician who is fully trusted by the patient, e.g., his general practitioner. All data that are stored in any such system

have to be maximally reliable and objective; one should also be very careful in storing subjective data or personal opinions. All databases should be subjected to a periodic auditing; the Privacy Committee should take care of the observance of all such requirements. In these committees patient and consumer organizations should also be represented.

A physician who has (perhaps accidentally) knowledge of the risk of his patient or client, for instance based on this genetic data, now faces the difficult legal and ethical problem whether, in some circumstances, he should inform the patient – or even his relatives. Most experts in such matters have the opinion that the physician should only transfer information if the patient requests so and is able to carry the burden of knowing. In some circumstances people prefer to live further without having knowledge of their future destiny. In other circumstances, however, people may decide to be fully informed, for instance when they consider to marry and/or to have children. True prevention may ultimately imply the renouncement of offspring.

REFERENCES

1. Harris R. – Genetic counselling and the new genetic. TIG 1988; 4: 52–6.
2. Loppe M. – The limits of genetic inquiry. Hastings Center Report 1987; 17: 5–10.
3. Holtzman N.A. – Public interest in genetics and genetics in the public interest. Am J Med Genet 1980; 5: 383–9.
4. Special issue. Am J Med Genet 1987; 26.
5. Rowley P.F. – Genetic discrimination: rights and responsibilities of tester and testee. Am J Hum Gen 1988; 43: 105–6.
6. Lappé M. – Ethical issues in genetic screening for susceptibility to chronic lung disease. J Occup Med 1988; 30: 493–501.
7. Kenen R.H., Schmidt R.M. – Stigmatization of carrier status: social implications of heterozygote genetic screening programs. Am J Publ Health 1978; 68: 1116–20.
8. Holtzman N.A. – Recombinant DNA technology, genetic tests and public police. Am J Hum Gen 1987; 42: 624–32.
9. Constitution of the Kingdom of The Netherlands.
10. Treaty for the Protection of Human Rights and Fundamental Freedom. Rome; 1954, art. 8.

Application of Encryption Techniques for Security Purposes in Financial Systems

by T.W.M. Jongmans

De Nederlandsche Bank N.V., Postbus 98, 1000 AB Amsterdam, the Netherlands

1. INTRODUCTION

Today's theme is 'how do I protect my data'. This theme is above all interesting for banks because, contrary to other enterprises, banks do not deal in goods whose amount and location are recorded in a stock accounting system. Sure, banks do operate an accounting system, but the data contained in it do not concern stored banknotes; rather, *the data themselves are money*. The balance on your bank account *is* money, which you can spend in a variety of ways or must repay, depending on whether you are in the black or in the red.

Allow me to give some statistics on different volumes of payments:

- (a) Each year Dutch households pay some 90 billion guilders *in cash*. For enterprises the figure is 45 billion. However, *giro transfers* involve some 2,500 billion guilders a year.*
- (b) Annually, the banks' clearing house (BGC) processes about *one billion* giro transfers.
- (c) Annually, in the Financial Accounting System of De Nederlandsche Bank some 10,000 *billion guilders* is transferred between financial institutions.

It will be clear that for banks the question 'how do I protect my data' is exceptionally important, since it is equivalent to the question 'how do I protect my money'.

In terms of substance, too, these questions are far from trivial. This is due primarily to the emergence of automated networks, so that money is handled

* These figures, established by the Scientific Research and Econometrics department of De Nederlandsche Bank, are necessarily not more than indications. However, they do give an impression of the order of magnitude of real payments, excluding transfers between financial institutions.

by a widening variety of processes in ever-greater volume and at ever-higher speeds.

Hence, in a sense, the numbers that are money, the subject discussed by David Chaum (these Proceedings pp. ??-??), are already occurring in day-to-day practice, albeit in a more prosaic form than in his protocols. What I intend to do here is to give an impression of security practice with regard to financial systems (first the classical methods, followed by the modern ones). I shall do so from the perspective of the Nederlandsche Bank, and especially in the light of the Bank's main tasks, as they have been formulated in section 9 of the Bank Act:

- regulating the value of the guilder;
- facilitating transfers and external payments;
- supervising the solvency and liquidity of the banking system.

Each of these three principal tasks is in some way related to the reliability and security of the payments system and of the automated financial processes used in this system.

2. THE CLASSICAL BASIS OF SECURITY

The classical aim of security in automation is to ensure uninterrupted processing by safeguarding the proper operation of the automated systems and controlling the risks ensuing from the use of such systems.

Specific goals are:

- reliability (completeness and correctness) of the data produced and stored;
- controllability of processing;
- continuity of services.

It is generally known that measures are necessary in *classical* financial systems and are more or less (but not quite) sufficient in order to achieve these security aims. In brief:

- (a) Structured systems development, focusing explicitly on processing checks, segregation of functions, audit trail, etc.
- (b) Segregation between development, testing and operation in order to ensure software integrity and stability.
- (c) Controlled access to software and data in order to permit only authorized actions by authorized persons in their proper interrelationship. This applies to the financial systems themselves, but also, more in general, to *all* the software and data present in the computer in order to ensure *total* system integrity.
- (d) Operating procedures.

Financial systems must be operated in a controlled fashion in the same way as industrial processes. The systems must be stable, robust and easy to operate (when operators and systems managers have to run their systems

in a state of agitation, this means they are not fully in control of the situation. This must always be avoided, a computer room must be a dull place where no time-critical or otherwise critical decisions need to be taken unless they have been adequately tested and rehearsed).

(e) Backup, recovery, disaster recovery.

Procedures must have been prepared, tested and rehearsed for emergency situations, ranging from malfunctions to calamities, in order to permit data recovery and continuation of services, if necessary at a reduced functional level, until the problem has been resolved.

(f) Physical security.

All computer systems which permit physical manipulation of the hardware, software or variables are weak. As we have seen, a computer contains money and, hence, *also* requires a protected environment in a physical sense in the same way as banknotes are stored in vaults.

Tens of thousands of pages have been written in the form of checklists, monographs and audit manuals about the purpose, design, implementation and cost of combinations of the above measures. On 20 September 1988 the Nederlandsche Bank added a modest 10 pages in the form of its memorandum on the reliability and continuity of EDP at banks.

The significance of the memorandum is mainly to be found in the fact that it requires the boards of the individual banks to pay attention to security issues. In this way the central bank, acting in its capacity as supervisor of the financial institutions, *stressed* that inadequate security of automated information systems may cause a bank's solvency and liquidity, and hence indirectly the function of the banking system in society, to be impaired.

The developments which have taken place in payments in the two years since the memorandum was published suggest that in the future the classical security techniques will no longer be adequate, so that more powerful procedures will be needed.

3. THE MODERN SECURITY TECHNIQUES

I shall now discuss the more advanced security techniques, which are notably applied within the framework of the new developments in payments.

First, an overview of some major forms of payments:

- Card-oriented
 - credit cards
 - cash dispensers
 - point of sale terminals
- Paper-based
 - personal sector giro payments (transfers, cheques, inpayment transfers)
- Message-oriented
 - business sector giro payments (magnetic tape, diskette, direct debits)
 - electronic payments (personal and business sectors) (home banking, telebanking)

- interbank payments (high-speed circuit of the banks' clearing house, SWIFT network, Financial Accounting system of the Nederlandsche Bank)
- settlement (Financial Accounting System)

The crucial questions in the case of card-oriented systems are: is the card *genuine*, is it used by the *rightful* owner, and is the transaction *within the applicable limit*?

Security has been known to be infringed in each of these three areas. Counterfeiting of cards is countered by using high-tech cards, with holograms, made with sophisticated printing techniques and employing esoteric physics. The card is combined with a PIN code in such a way that the only manner to gain access to financial services is through the combination of card and code. The PIN code is a cryptogram of the data contained in the magnetic stripe.

The most difficult problem is to enforce transaction limits. Strictly speaking, it is not necessary for cash dispensers and point of sale terminals to be connected on-line with a central computer, since it would be sufficient if the transaction data were transmitted once a day; however, in such off-line situations, it is difficult to prevent overdrafts. Hence, cash dispensers and point of sale terminals must have an on-line connection, leading to a considerable increase in the costs of data communication.

At present an experiment is being conducted at Woerden to solve the problem in an entirely new way, by using a smart card. Contrary to magnetic stripe cards, smart cards cannot really be counterfeited and are able to enforce transaction limits without any connection with a central computer. Although the experiment has not yet been completed, it is already evident that the smart cards technology is very demanding and that the cards still require some improvement. The consequent effect on costs is not yet clear.

Compared with the new method, paper-based payments (using cheques and transfer forms) are somewhat obsolete and uninteresting. Nonetheless, they give the banks serious cause for concern. Processing (sorting, data entry, filing) is expensive, and in the disastrous period round about 1986 the combined banks incurred losses of up to 100 million guilders per annum as a result of fraud with cheques. The profits of some banks suffered considerably as a result of these losses. Since that time, new procedures have been successful in reducing fraud. Through their structure of charges, the banks seek to influence consumer preferences towards more efficient and lower-cost payment methods.

By the way, I have already used a fair proportion of the time allotted to me and I have not even yet used the word encryption. Let us, therefore, turn to the message-oriented payment methods; I shall use the word message in a broad sense to denote magnetic tape and diskettes as well.

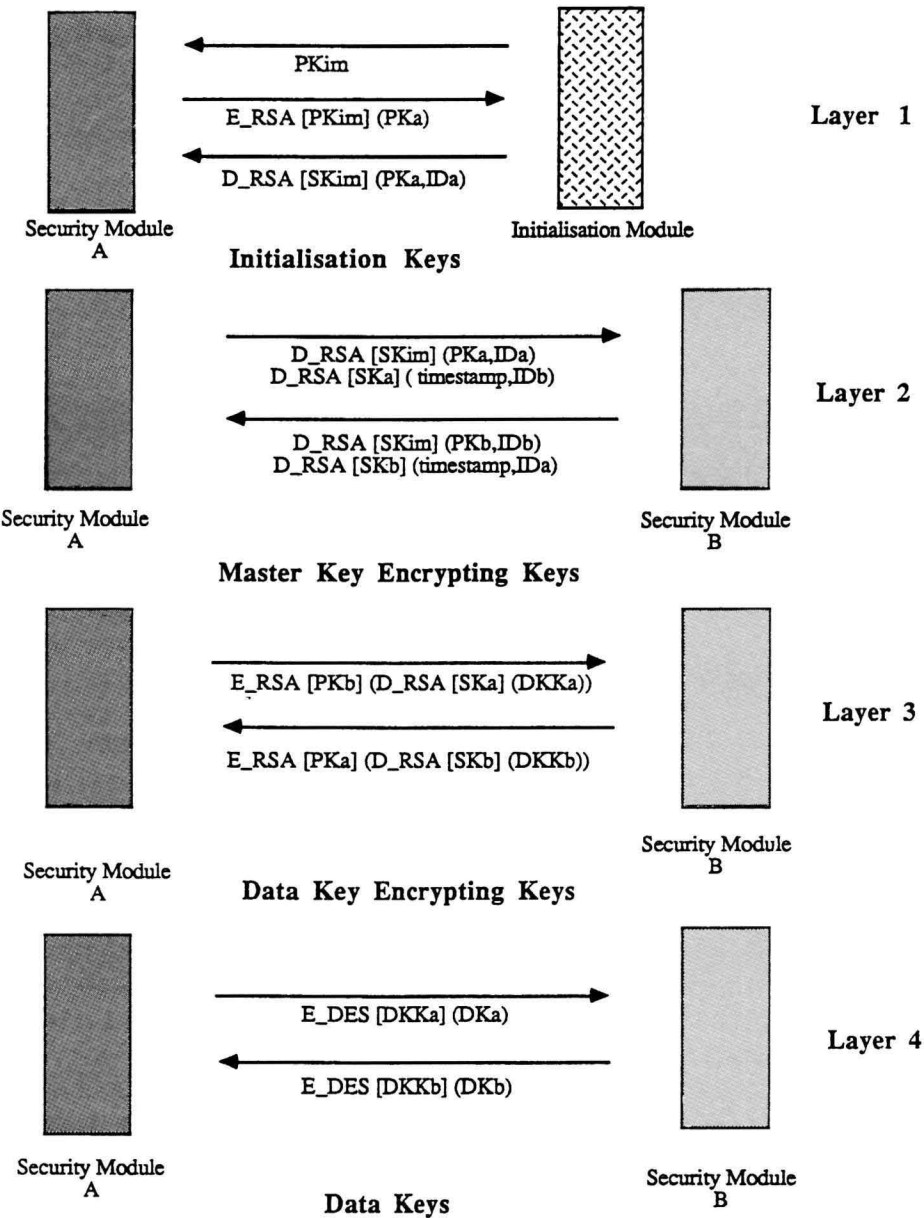
In message-oriented payments, the crucial security questions are: has the integrity of the message been preserved (is the message unchanged or unmutated), is it authentic (has it actually been sent by the person indicated as the sender) and has it been authorized (is it a valid instruction)?

The integrity of the message, which concerns what security experts in banking usually call layer 1 security, is checked by means of a hash count relating to the contents of the message. A hash count contains information *about* a message which very probably changes when the contents of the message are changed. A constant is not a hash count. The sum total of amounts or the number of items in a message are poor hash counts; cryptographic one-way functions are excellent hash counts. One of these has been developed by IBM, termed MDC (modification detection code). Another has been developed by an interbank working party under the auspices of the banks' clearing house (BGC); it is known as the BGC hash. In the Dutch banking system the BGC hash is the standard. It has also been proposed for standardization within the ISO. Writing the hash count on a piece of paper, signing it and adding it to the message ensures not only the integrity but also the authenticity of the message. Adding the signature of an authorized signatory furthermore proves the authorization or validity of the transfer order. Unfortunately, this is possible only for magnetic tapes, diskettes and similar media but not for data transmitted by means of data communication.

Hence, there is a need for suitable layer 2 security, that is, the authentication of the message. MAC (ISO 8730, ANSI) is a standard for a message authentication code. Other methods have been developed as well, such as the BGC layer 2 authentication, the French system Etabacc5 and the German Deutsche Bank system. The principal theoretical issue concerns the use of symmetrical or asymmetrical algorithms (in practice the choice is between DES and RSA). (For RSA, see Massey's contribution to these Proceedings, p. 27.) The principal advantage of DES is the fact that high-speed hardware and software are available; a few megabytes per second are easy to achieve. The disadvantage is that, because of its symmetrical nature, DES is sensitive to insider attacks: secret keys must be managed and transported. In the case of RSA, this drawback carries less weight: only the user's own key is secret but it need never be transported. The major drawback of RSA is that implementations are very slow: a few kilobytes per second is considered a very high speed in this context. If operations are conducted at that rate, how can the banks' clearing house ever process its three million transactions a day? This problem has led to the development of hybrid systems, which employ DES for bulk transactions and RSA for the upper layers of key management. By the way, development may be too big a word: hybrid systems have been conceived, but I for one have never seen one implemented in either hardware or software.

I would like to discuss with you the design of a hybrid system in which I myself was closely involved: the security concept for the National Payments circuit, known as the NBC. The NBC is a project to achieve uniform direct interbank payments. After many years of preparation, the project was started in 1985 and it is being realized in stages and on a limited scale. In the period 1987-1989 an interbank working party designed a security system for the future data communication within the NBC. The chart shows the key management protocol developed by the working party. The bottom layer of key management

is the exchange of data keys, for instance daily (layer 4 in the Chart); it is implemented in conformity with the ISO 8732 point to point protocol. The exchange of the symmetrical encryption keys (layer 3) is effected weekly or monthly and takes place under an asymmetrical algorithm with authentication. The public keys are exchanged in the forms of a certificate in layer 2. This certificate is prepared by a central institution in layer 1.



Exchange of keys between Security Modules and Initialisation Module

It took only a few months to prepare the concept and to have it approved by the banks. However, the search for suitable implementations in hardware and software is still in progress. Even at this moment I cannot say that the search has been completed in the sense that the concept has been successfully implemented. NBC is at present secured by other means. Indirectly, a success has been scored in that, within the ISO context, a closely parallel concept is now a candidate for standardization. At present, it has the status of a committee draft (ISO/TC68/SC2 Committee Draft 11166).

I would like to use the last part of this address to discuss the practical aspects of encryption.

4. PRACTICAL ASPECTS OF CRYPTOGRAPHY IN FINANCIAL SYSTEMS

Cryptography is one of the most powerful security methods and has very specific areas of application. For some security requirements it is the only feasible solution. On the other hand, it is a complex and difficult subject, with which just a few experts are conversant.

I should like to discuss what I think are the two key problems of applied cryptography in banking.

First: Finding a cryptographic solution to a security problem is virtually always possible; the difficulty is to define precisely the security problem itself. Yet this is essential in order to know exactly

- what is secured, and, at least as important,
- what is not secured.

Anyway, from the viewpoints of both sound banking and competent investment, it is necessary to know exactly the security performance of measures taken and whether the measures do *indeed* satisfy the relevant requirements. The Vernam Cipher (cf. the first paper of these Proceedings, p. 5) is a case in point. Suppose that a transfer message has the following layout: account number of the ultimate payee 64 bits and the amount 64 bits. Someone who changes bit number 66 of such an encrypted message can be pretty sure that he substantially raises the amount. The security *performance* of the Vernam Cipher is confidentiality, which is not the same thing as authenticity and integrity. The lesson is that even a very powerful security tool may not be appropriate to your application.

Let us take another example. One might well wonder what the *legal* status is of certain measures, such as non-repudiation. Non-repudiation is the situation where the sender of a message cannot deny having sent the message. It can be achieved by means of an authentic signature on a piece of paper. However, will the Courts also accept digital signatures in the case of data transmission? I do not know. Within the NBC, we did consider whether complicated legal non-repudiation procedures were actually necessary. We decided that this was not the case, because that is not the way banks go about things with their fellow banks; most disputes never reach the courtroom. This is perhaps best illustrated

by the manner in which foreign exchange dealers work. They transact their business by telephone, somewhat like this: 'Do you have 10 million dollars for me?' 'Sure, at 1.6720.' 'Okay, it's a deal!' This two-second dialogue between dealers entails the same obligations as a duly signed legal document. (By the way, the dialogue is taped; if one of the parties has made a mistake, the tape is used to clear the matter; however, the verbal agreement will always stand!)

Identifying and solving a security problem in respect of a system must be based on an analysis of the system made by the owner of that system. This must not be done by an encryption expert who has no authentic affinity with the system. The typical automation situation, in which the one who *experiences* the problem (the security problem owner), is not the same as the one who *solves* it, is very evident in the area of security, and even more so when cryptography is used.

Second: practically no-one (including in any case many cryptographers) is really aware of the immense difficulty of operating encryption techniques *in practice*. This is due not so much to encryption itself as to the limited availability and flexibility of encryption hardware and software; another problem is the lack of common ground between encryption experts and systems designers.

To conclude this address, I shall give an overview, based on my own personal experience, of the numerous vexing practical problems and constraints, which cannot be avoided:

- The cost problem (good encryption hardware and software can cost up to 5,000 guilders per work station and that may be more than the cost of the station itself; this is acceptable in special cases only).
- The labour-intensity of key management (initializing a smart card requires that the card is inserted in a card reader, packed and sent, something that will easily take a minute or so. Doing this for 100,000 smart cards takes a lot of time).
- Global secrets, which are present at various locations, must be avoided. Such secrets require tamper-resistant hardware, which is expensive; moreover, if the secret leaks out, the system must be stopped. Also, situations must be avoided where institutions must know each other's secrets.
- Details of security measures must be kept secret, whereas a knowledge of the system must be present within a stable group of operational staff. These are conflicting requirements.
- Dependence on a single supplier must be avoided. However, this is practically impossible in the case of hardware and difficult in the case of software (due to incompatible implementations).
- Strict security measures are hard to reconcile with sound backup, recovery and disaster recovery procedures. Furthermore, they complicate operation (remember, operators in a state of agitation are a sure sign of serious problems), repairs, and especially maintenance and change-over to new releases.
- Security always detracts from the user-friendliness of a system, because of:

- more management effort, such as the issue, monitoring and withdrawal of access rights and authorizations (a typical problem is that, once issued, an encryption key certificate cannot be withdrawn);
- stricter procedures (segregation of functions, limitation of functions, frequent log-on);
- psychological and ergonomic factors; care must be taken to ensure that the end user, who experiences the burden (never the fruits) of security, is intuitively able to grasp and accept the significance of the measures taken and to appreciate that he has an *inherent* interest in adequate security. If that is not achieved, all security efforts are in vain;
- lack of uniformity: users are sometimes confronted with a multitude of devices, diskettes, bank cards, PIN codes, log-on/log-off procedures, etc., if more than one bank is involved; this may degrade the commercial viability of automated banking services.

CONCLUSION

I hope that, through this overview of the potential applications of cryptography in the financial area, I have succeeded in showing that cryptography is not just interesting from a theoretical viewpoint, but that it also holds out fascinating prospects in practice.

REFERENCES

- Boeschoten W.C. and M.M.G. Fase – The way we pay with money, *Journal of Business & Economic Statistics*, July 1989, Vol. 7, No. 3, 319–326.
- Boeschoten W.C. and M.M.G. Fase – Het bankbiljet van f 1.000,-, *Economisch Statistische Berichten* 73, no 3658 (June 1, 1988), 523–527.
- Boeschoten W.C. and M.M.G. Fase – *Betalingsverkeer en Officieuze Economie in Nederland, 1965–1982*, Kluwer, Deventer 1984.
- The Memorandum on reliability and continuity of electronic data processing in financial institutions was published in the fourth quarterly report of 1988 of De Nederlandsche Bank.

Computing and Security; a Task for the Lawyer?*

by H. Franken

*Juridisch Studiecentrum 'Hugo de Groot', R.U. Leiden, Postbus 9520, 2300 RA Leiden,
the Netherlands*

INFORMATION MEANS POWER

Knowledge means power. So most of the hacker stories are success stories because of the respect people have for the lonely, young, intelligent, pale student who is too sharp for a multinational. Some high-school boys, working with cheap home computers, broke into the datacentre of the Chase Manhattan Bank by telephone. They came in using the free clients line and they tried various entry-codes. Once in the system, they changed all the entry-codes so that the bank could not reach its own data any more.

Another story concerns a prisoner in the USA, who was working on his resocialization-plan. He took a course in computing and succeeded in gaining access to the prison records. So while sitting in his cell, he changed the date of his release and was a free man two months earlier than he was supposed to be released.

In Holland hackers got into the central computer of the National Post Office where all the lists of secret telephone-numbers of ministers and police authorities are stored. Other hackers have manipulated bank computers getting away with millions of dollars. But it has also happened that schoolboys have manipulated radiation data in a hospital computer system. Such interference can be fatal, and then we are talking about attempted murder.

* Parts of this article are quoted from the report of the Netherlands Committee on Computer Crime, which I had the honour to chair. (SDU the Hague 1987).

ABUSE OF INFORMATION

Knowledge means power. Information means power. It is clear that information can also be used to hurt other people. Such abuse of information may happen by

- interruption of information systems
- violation of secrets
- manipulation of data.

We must all of us realise that nowadays information technology presents a new challenge to those who intend to abuse information or data. It is a challenge to hackers and to people who want to disclose and publish confidential documents. In addition modern information technology is a very powerful tool in the commission of well known crimes such as fraud, forgery and swindle.

DARK NUMBER

It is difficult to estimate the extension of the abuse of data in our information society. There are several reasons for this statement: First of all, how does one go about discovering computer abuse? It is a big problem. All the cases we know about are the result of a clear mistake by the offender and not of initiatives of the police authorities.

In the second place, it is not possible to categorise a great deal of the abuse of data within the existing types of offences recognized by the law. Thirdly, there is a problem in ascertaining any increase in the rate of abuse due to the lack of cooperation by victims with the police authorities. Many victims will not inform the judicial authorities for several reasons:

- a. They fear that this will damage their company's reputation. Clients may get the impression that the company is not safe.
- b. The victims are afraid of the way the judicial authorities behave. Policemen are not yet accustomed to investigating computer crimes and they may cause damage to the business. Shutting down the information system, or seizing tapes may cause chaos in the administration or production process.
- c. Another reason for not informing the authorities is that the victim often prefers to obtain reparation from the offender for the damage caused rather than to suffer the loss and put the offender in jail.

INCREASE OF ABUSE

Although we are confronted with a lack of research results about the dark number of abuses of modern information technology, we cannot avoid the conclusion that there has been an increase in the number and variety of these abuses. The more sophisticated the technology becomes... the more sophisticated the crimes become. In recent years we have seen an increase in the daring of the abuser, who has little fear of being discovered. And what's more: an important increase in the total losses has taken place.

It is also clear that the range and type of victims has already become very wide. All sorts of businesses, governmental departments and private individuals have been affected.

It is also interesting to note that we can find future offenders in groups of criminals cooperating with whizz kids and computer professionals. And: A recent Swedish study reports that the number of female offenders is almost the same as that of male offenders. That is really unusual in criminology! – a result of emancipation!

Looking to the future we can be sure that the abuse of computers will increase not only because of the continuity of the trends I have already mentioned, but also because of a favourable trend in technological developments. The hardware and software are designed to be user friendly.

This notion of 'user-friendliness' is an important selling point, yet we must be aware that it also contributes to the decline of computer illiteracy. User-friendly interfaces enable a broader section of the public to use modern information technology which also means that criminally minded people will have access to it, too. Furthermore, we know organizations and institutions are becoming more and more dependent on electronic data processing, and that in turn means an increasing vulnerability in the way all our organizations – our social structures – function.

COUNTERMEASURES

What can we do against these abuses of information or data? There are several ways in which to protect ourselves against the interruption of systems, breach of confidence and manipulation of data.

First, preventive measures can be taken to reduce the risk of damage. These measures concern not only defects in hardware and software, and take into account the faults of owners and personnel, but also affect unforeseen risks, such as accidents and misuse by the owner's own personnel as well as by outsiders. We can divide the preventive measures for risk avoidance into four basic categories.

They concern:

- physical measures (as a safe place in the building);
- organizational measures (such as separation of functions between designers, operators and controllers);
- logical measures (such as protection built into the program itself – encryption modules, for example);
- legal measures.

The last category refers particularly to the transfer of risks to third parties. It may sound rather cynical but this is a task for the private lawyer. In all stages of the contact between sellers and buyers, and during the whole period in which a person uses computersystems, he has to be aware of risks he can put on the

shoulders of another person or company. This can be achieved through negotiations with one's partners in business or with one's employees, for example by inserting competition clauses, or by buying security from insurance companies or specialized agents such as escrow firms. I mention this because we must be aware that the legal solution of the problems of risk should be solved in the first resort by measures initiated by the private persons themselves. My main subject in this article, however, concerns the possible measures a government can take. But these measures will not have sufficient effect when the citizen or company does not act at a private law level also.

GOVERNMENTAL ACTIONS

Now let us look at the task of the government. Government has a powerful tool, called legislation. But here there's an inherent conflict. On the one hand we acknowledge the fundamental principle of the free flow of information, on the other hand society demands protection of data. The European Convention on Human Rights guarantees the freedom to receive and to gather information. It also provides for legal rules which must be made to protect data concerning health, reputation, privacy and the rights of others. Several countries have already passed legislation on this subject, while others are still at the discussion stage. Here finally we see that the Organisation for Economic Cooperation and Development has declared in late 1986 that in almost all member countries, governments and courts are confronted with a new kind of criminality. This criminality shows the same characteristics in all countries and therefore similar measures are required to avoid the creation of computer crime havens and to protect countries from becoming victims of such criminality of foreign origin.

For these reasons national legislation should be made to combat the following acts:

- a. The input, alteration, erasure and/or suppression of computer data and/or computer programs made wilfully with the intent to commit an illegal transfer of funds or of another thing of value;
- b. The input, alteration, erasure and/or suppression of computer data and/or computer programs made wilfully with the intent to commit a forgery;
- c. The input, alteration, erasure, and/or suppression of computer data and/or computer programs, or other interference with computer systems, made wilfully with the intent to hinder the functioning of a computer and/or telecommunication system;
- d. The infringement of the exclusive right of the owner of a protected computer program with the intent to exploit the program commercially and put it on the market;
- e. The access to or interception of a computer and/or telecommunication system made knowingly and without the authorization of the person responsible for the system, either (i) by infringement of security measures or (ii) for other dishonest or harmful intentions.

The *OECD* view is that the criminal law should cover these types of conduct. The question is: can we combat this behaviour with the criminal law? The answer is no: the present provisions of the criminal law are no longer sufficient. We have several arguments for this statement.

1. In the statutes of many countries the offence of forgery is formulated in terms of counterfeiting a document. Forgery concerns falsely changing a message that is embodied in a durable way. But how does one treat the falsification of a computer program which is not embodied on a material carrier?

2. In respect of most fraudulent acts the act of a human being is essential. But realize: when a manipulated smart card is inserted in the petrol pump: who has delivered the gas? The machine has been manipulated and not a person. As a consequence this does not constitute fraud under the legislation of the most European countries. The legal definitions concern acts by a human victim and not by a 'machine victim'.

3. A rather prosaic argument. The punitive measures of the present statutes are too low to deter computer crime. In the Netherlands a swindler can get a maximum of 3 years in prison. A thief or a forger can get at most 4 years of detention. Another rule provides that one third of the punishment may be remitted for good behaviour in jail. So when his profits are several million guilders and the costs are only two or three years in prison then the overall profit will be a real incentive, certainly when it is gained tax free!

4. Another argument for the adaptation of the present criminal law: there are new types of behaviour which deserve penal sanctions that are not provided for in the present regulations: for example hacking, tapping of data communication, and data manipulation – such as the case of the prisoner who freed himself 2 months prematurely.

5. The next point has been the subject of debate between lawyers for several years. Our present statutes concern the property and loss of material 'goods' and especially formulated rights. The latter are protected by copyright or patent law. However 'goods' normally correspond to material objects. It is necessary to consider whether information or rather, data may be deemed a 'good' within the meaning of the law.

In the event of this question being answered in the affirmative, then data would be accorded protection on the basis of provisions in force regarding theft, embezzlement, criminal damage and the like. My opinion is *that data cannot be deemed a 'good'* and that consequently definitions of offences in which the term 'good' occurs have no bearing on data. As this conclusion may appear to be in conflict with some recent legal judgements in Belgium and the Netherlands, I shall give you the reasons for reaching this opinion.

ARE DATA TO BE CONSIDERED GOODS?

On October 1983 one of the Dutch Courts of Appeal decided a case of a man who copied software from a data carrier belonging to his employer onto a data

carrier belonging to him. He subsequently resigned and started up his own business in the same field as his employer. He made use in his business of the software which he had copied which, he admitted, saved him six months of research and development.

The Court of Appeal held that there was sufficient evidence in law that the accused had made unlawful copies of certain computer data which did not belong to him. The Court assumed that these data could be deemed to constitute a good within the meaning of the law, and that they were therefore capable of being embezzled.

The reasoning which led the Court to reject the defence argument that this case did not involve goods as defined by the law was the same as that used by the Dutch Supreme Court in 1921 in the ruling known as the Electricity Judgement. This judgement held that *electricity* is a good because:

- it cannot be denied that electricity has a certain independent existence;
- this energy can be transported and accumulated;
- this energy represents a certain value to the person who generates it, on the one hand because it takes money and effort to obtain it, and on the other hand because this person is in a position either to use this energy for his own purposes or to transmit it to others in return for remuneration.

The Court of Appeal adopted this line of argument in the case of computer data: these are also available, transferable and reproducible and possess an economic value, so that they, like electricity, can be deemed a good.

This development would appear to constitute a further step in the evolution of the concept of a 'good' from a purely physical, tangible object to include intangible things, from the material to the immaterial, from property object to property value. The question remains, however, as to whether this tendency is to be applauded. I feel that this stretches the meaning of 'good' too far, in such a way that it also embraces things which differ too greatly from material objects and on account of this ought to be dealt with in a different manner.

It cannot be denied that both data and material goods are transferable, reproducible, available and sometimes possess economic value. However, there are obvious differences. Goods (which also includes electricity) are the product of physical labour, while data are the product of mental effort: data, after all, often reflect or embody knowledge.

In addition to this, goods are *unique*: ownership or possession of these goods implies that others are denied the ownership or possession; data, on the other hand, are *multiple*: possession of them does not stop others also having possession of the same data. The act of copying does not deprive the legal 'owner' of any of his power – he continues to possess the data. It is not so much that he loses possession of the data, but he loses the *exclusive* possession of these data. It is my opinion that it would be going too far to extend the concept of 'goods' to include 'data'. Data, which are also taken to include software, are primarily intellectual products, to which other forms of protection should apply

apart from those which protect material objects. This conforms with the traditions, such as copyright and patent laws.

HOW TO CREATE NEW REGULATIONS?

New regulations are needed, but how can they be created? It does not fit with the status of a democratic society to let the work be done by the judge, who interprets by way of analogy the existing rules. No, this is a task for the legislator. But the legislator has to bear in mind that he can't use normal or traditional legal terms and concepts such as forgery, fraud, document, good. The legislator must also be aware that he can't use technical terms because they will become obsolete in a few months. There are many and rapid changes of concepts in the field of technology. The legislator has to look for new standards of behaviour. This is a potential field of research for the lawyer. I think these standards can be discovered through feedback mechanisms in which the interests which can be harmed are discovered. There are three types of interest which are at stake: availability, integrity and exclusivity.

The first interest concerns the *availability* of the means of storage, processing and transfer of data and of these data themselves (including software). The importance of uninterrupted access to these means and data increases in proportion to the degree of dependence of a society on these media and data.

The availability of means and data may be jeopardized by deliberate acts of malevolence such as sabotage, damage, destruction or removal of media or data, or the obstruction or interruption of data communications.

In order to achieve correct results and to be able to take the right decisions using data in computerized systems it is extremely important that these systems operate properly and that the data and programs are correct and complete. This is what is meant by the *integrity* of the systems and the data they contain.

If this integrity is undermined the result may be the disruption of production processes, the failure of the security systems of electricity generators or traffic control systems, or the payment of incorrect amounts in salaries or benefits, or any such potentially large-scale and costly malfunction.

Integrity may be damaged too by the falsification of data and software involving alterations, addition or removal of certain elements.

Having looked at the concepts availability and integrity, there is a third element. This involves the interest which companies, organizations and individuals can have in according data an *exclusive character*, for example because they do not wish unauthorized people to have access to secret or confidential information or because they wish to have exclusive control over how media and data are used and by whom.

In the first place the unauthorised possession, reproduction and dissemination of secret or confidential data may be considered prejudicial.

In addition to this there may be an interest in imposing restrictions on the use of particular data or media which are not in themselves secret or confidential. As they are the fruits of investment, it is understandable that there may

be resistance to the idea of third parties making use of the resultant products, for example by copying them or marketing them commercially without paying for them.

NEW OFFENCES

On the basis of the notions of availability, integrity and exclusivity the legislator can start his work. I had the honour to chair a committee charged with drafting a statute on this subject. In our draft we formulated new offences such as:

- disturbing data communication
- tapping data communication
- data manipulating
- computer trespass
- special rules for cheque cards

Apart from these rules for the behaviour of the citizen we need new areas of competence for the public prosecutor and the judge. We need the competence

- to receive and to gather information; that is to be able to search in a computer system.

- to decode a program; i.e. to oblige the system operator to give access, to enable the judge to oblige the operator to remove an encryption in order that the search may be conducted.
- to tap data communication for police purposes. At present the law permits the tapping of telephone conversations, but not the tapping of the communication of data in other ways. It is urgent that this lacuna is filled.

A THRIFTY HOUSEWIFE

A further statement must be made. It is an important point of judicial policy that a legislator has to behave like a thrifty housewife when it comes to making criminal law. This part of the law must be reserved for the last stage of control of the citizen, because it gives potentially wide reaching powers to the state. Through economical use of the criminal law my committee recommended that the unauthorized use of information technology equipment should not be penalized. An example of this behaviour is where employees carry out private work on their employers' computers.

Unauthorized use is not a criminal offence in many countries except in the case of 'joy riding'. 'Joy riding' can be distinguished from 'joy computing' in that it occurs in the public domain. In contrast, unauthorized use of computer media will generally occur in the non-public domain, so that it can be dealt with by means of internal disciplinary procedures. Where unauthorized use is carried out by an outside agent, this implies that unauthorized entry has been obtained. In that case we can't speak any more of the private character of the use of the equipment. When there are people from outside who try to get access without being authorized, we can speak in legal terms of computer trespass by analogy with the criminal offence of trespass of a dwelling, room or property. In defin-

ing computer trespass as a punishable offence we look upon the obtaining of unlawful access to computerized data-processing systems or those parts which are protected against intrusion. In my view it should only be a punishable offence if some form of security or protection against invasion is violated in order to secure unlawful access. Intrusion can be said to occur where a person obtains access without the consent of the authorized owner or user – this will can be demonstrated by words ('entry prohibited') or by deeds. In my opinion words alone are not sufficient. Words, such as a text on the screen stating that entry is prohibited for unauthorized persons, do indeed show an unambiguous desire on the part of the authorized controller, but do not exclude the possibility of entry by accident. This danger is far less great when a higher threshold is created, consisting of particular security measures to combat unlawful entry. This introduces a further restriction. The door must not only be closed, as it were, but also locked. The point at which the security measure is applied is then regarded as the border between the 'private domain' and the area that is open to the public. The background to this proposal is the belief that criminalization is necessary because 'computer trespass' as such is improper. The criminalization of such activity also creates an obstacle to harmful acts which might follow intrusion into computer systems (e.g. altering or erasing data, reading or copying confidential data). Subject to the necessary restrictions, the criminalization of computer trespass offers indirect protection to data in data processing systems.

THE CIVIL LAW

It is not only the criminal law that should be handled as a tool for impeding computer crime.

As well as the government, all branches of the private sector have an interest in a smoothly-running system of data management. The stipulation of rules pertaining to legal persons can provide a major stimulus to the creation of barriers to combat carelessness in protecting data flows. It is possible to imagine such a statutory regulation which could be introduced into the Civil Code. This might take a number of forms:

- (a) as part of the annual auditing of the company accounts, the registered accountant would have to provide an assessment of the security of the computerized data processing systems used by the company;
- (b) an expert (AC accountant or EDP auditor) would have to assess the reliability and continuity of the computerized data processing;
- (c) a statement regarding the reliability and continuity of the computerized data processing would be included by the directors in the company's annual report; this statement would be assessed by the accountant.

In my opinion, the first variant (a security audit) is not to be recommended, at least at present. The accountants' declaration on the annual accounts is concerned with the question of whether these provide a faithful representation of the assets and the results of the legal person.

In this regard, automatic data processing systems are only examined in so far as this serves the aim of the audit. Under this variant much more would be expected of the accountant, namely an assessment of all the automated systems in the company. Irrespective of the cost factor, it is unlikely that the accountant would be able to provide an unconditional assessment of this sort.

At the second variant – a management letter by an expert – the problem arises, as to who can be considered competent as an AC accountant or an EDP auditor to provide the required assessment. In the absence of any regulation of the training of such specialists, it is impossible to indicate a group of people who can exercise this competence in such a way that the public can and should rely on their pronouncements.

We are left with the third variant. This involves the directors of the company incorporating a statement in the annual report as to the reliability and continuity of the company's data processing system. This would explicitly indicate that responsibility for the scope and quality of security rests with the directors. They would have first to indicate in writing the requirements which security in the company in question has to fulfil. Finally, the accountant can publicly state whether this declaration by the directors is or is not a true reflection of the facts by comparing it with a set of rules.

CONCLUSIONS

In conclusion, we can say: the rapid developments in information technology and telecommunications create uncertainty among those involved; it is unclear what *is* and what is *not* allowed. The law can serve a function in delimiting the border between what is permitted and what is not permitted. Such signposts clarify the situation. They can also help to create an awareness of the norms among those who come into contact with computerized data processing and data transfer, whether as system managers or as potential offenders. But a lawyer in these times has to be a modest man. Because the final conclusion to be drawn is that for various reasons the law should be invoked sparingly. If too great a weight is attached to the criminal law it becomes something of a 'paper tiger' – with plenty of pretensions but little scope for genuine enforcement. It is better to be less ambitious and to concentrate on what are seen as vital interests. This in itself is a reason for guarding against 'norm-inflation'.

Security without Identification: Card Computers to make Big Brother Obsolete

David Chaum

*Centre for Mathematics and Computer Science, Kruislaan 413, 1098 SJ Amsterdam,
the Netherlands*

*You may soon use a personal 'card computer'
to handle all your payments and other transactions;
it can protect your security and privacy in new ways,
while benefitting organizations and society at large.*

Computerization is robbing individuals of the ability to monitor and control the ways information about them is used. Already, public and private sector organizations acquire extensive personal information and exchange it amongst themselves. Individuals have no way of knowing if this information is inaccurate, outdated, or otherwise inappropriate, and may only find out when they are accused falsely or denied access to services. New and more serious dangers derive from computerized pattern recognition techniques: even a small group using these and tapping into data gathered in everyday consumer transactions could secretly conduct mass surveillance, inferring individuals' lifestyles, activities, and associations. The automation of payment and other consumer transactions is expanding these dangers to an unprecedented extent.

Organizations, on the other hand, are attracted to the efficiency and cost-cutting opportunities of such automation. Moreover, they too are vulnerable, as when cash, checks, consumer credit, insurance, or social services are abused by individuals. The obvious solution for organizations is to computerize in ways that use more pervasive and interlinked records, perhaps in combination with national identity cards or even fingerprints. But the resulting potential for misuse of data would have a chilling effect on individuals. Nevertheless, this is essentially the approach of the electronic payment and other automated systems now being tried. Although these systems will require massive investment and years to complete, their underlying architecture is already quietly being decided and their institutional momentum is growing.

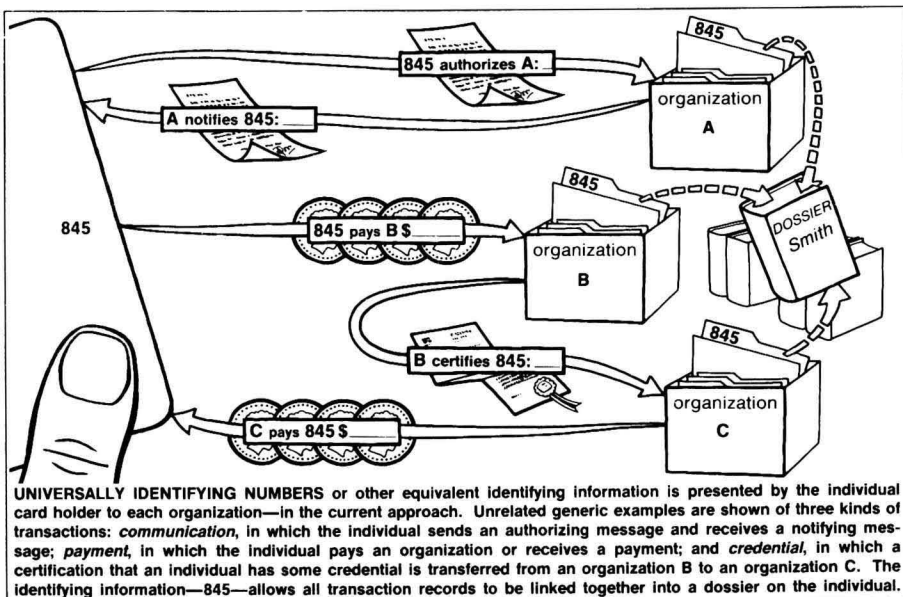
This momentum is driving us toward a seemingly irreconcilable conflict, be-

tween organizations' need for security and the benefits of automation on one side, and individuals' need for ensured privacy and other protections on the other. But this conflict may be avoided by early adoption of a fundamentally different approach to automating transaction systems. This new approach is mutually advantageous: it actually increases organizations' benefits from automating, including improved security, while it frees individuals from the surveillance potential of data linking and other dangers of unchecked record keeping. Its more advanced techniques offer not only wider use at reduced cost, but also greater consumer convenience and protection. In the long run, it holds promise for enhancing economic freedom, the democratic process, and informational rights.

The New Approach and How it Differs

Three major differences define the new approach. The first is in the use of identifying information. Currently, many Western countries require citizens to carry documents bearing universal identification numbers. Drivers' licenses are being upgraded to perform a similar function in the United States, and efforts toward machine-readable national identity documents are expanding internationally. Meanwhile, organizations routinely use such essentially identifying data as name, birthday, and birthplace or name and address to match or link their records with those of other organizations.

Under the new approach, an individual uses a different account number or 'digital pseudonym' with each organization. No other identifying information is used. A casual purchase at a shop, for example, might be made under a one-time-use pseudonym; for a series of transactions comprising an ongoing relationship, like a bank account, a single pseudonym would be used repeatedly.



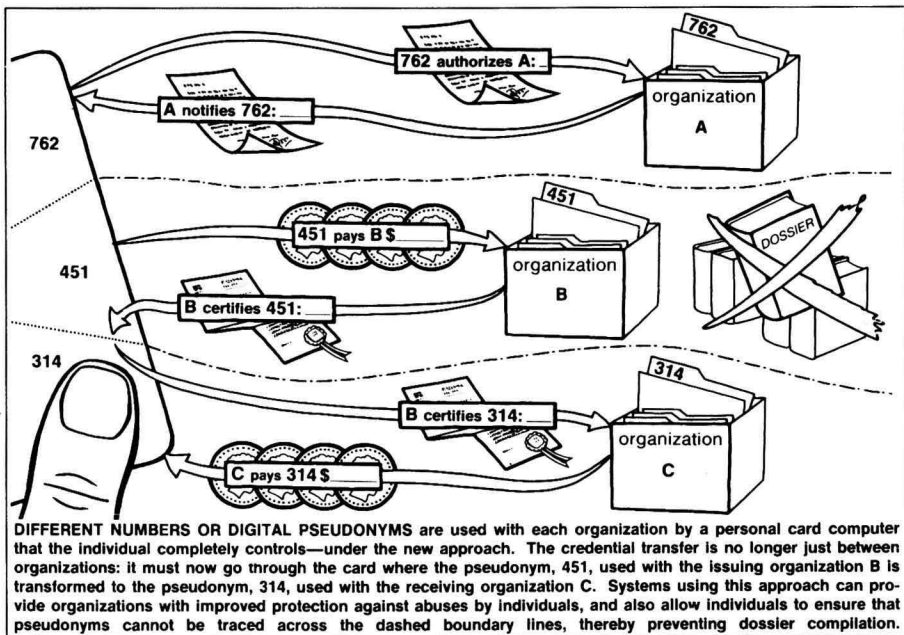
Because of the input individuals have into the process by which the pseudonyms are created, they are ensured that their pseudonyms cannot be linked. This input also yields them the exclusive ability to use, and authenticate ownership of, their pseudonyms. Organizations too can protect themselves through their participation in forming the pseudonyms; among other safeguards, they can limit individuals to one pseudonym per organization and ensure that individuals are held accountable for abuses created under any of their pseudonyms.

A second difference is in whose mechanism is used to conduct transactions. Today, individuals hold a variety of 'tokens' issued to them by organizations. These range from traditional paper documents to plastic cards with magnetic or optical stripes or even embedded microcomputers. Such tokens are usually owned by the issuing organization and contain information that the individual holder can neither decipher nor modify. With the spread of automatic teller and point-of-sale terminals, individuals are being asked to perform more transactions directly using computer-controlled equipment. These terminals, and even the microcomputers in some current tokens, are physically tamper-resistant and contain secret numeric keys that securely code their communication with central computers. Individuals derive little direct benefit from these security provisions, however: in using such a transaction mechanism, they must take on faith the information it displays to them while revealing their own secrets to it.

With the new approach, an individual conducts transactions using a personal 'card computer'. This might resemble a credit-card-sized calculator and include a character display, a keyboard, and a short-range communication capability (like that of a television remote control). Such computers can be bought or even constructed, just like any other personal computer; they need have no secrets from, or structures unmodifiable by, their owners. They can also be as simple to use as automatic teller machines. During a purchase at a shop, for example, equipment at the point of sale transmits a description of the goods and cost to the card, which displays this information to its owner. The card owner allows the transaction simply by entering a secret authorizing number on the card's keyboard. This same number is used by the owner to allow each transaction; without it, a lost or stolen card computer would be of very little use. A lost card's full capabilities, however, could be readily installed in a replacement, using backup data saved in a secure, encoded form at home or elsewhere.

The third defining difference is in the kind of security provided. Current systems emphasize the one-sided security of organizations attempting to protect themselves from individuals, while the new approach allows all parties to protect their own interests. It relies both on individuals' card computers withholding secret keys from organizations and on organizations' computers devising other secret keys that are withheld from individuals. During transactions, the parties use these keys to form specially coded confirmations of transaction details, the exchange of which yields evidence sufficient to resolve errors and disputes.

The systems presented here for the new approach depend on currently used codes to secure organizations against abuses by individuals. Since these codes



are 'cryptographic', they can be broken, in principle, by trying enough guessed keys. Such guessing, however, is infeasible because of the enormous number of possible keys. In short, no proofs of security are known for these cryptographic codes, but nor are any feasible attacks. By contrast, the security card computers provide for individuals against the linking of their pseudonyms is 'unconditional': simple mathematical proofs can show that, with appropriate use of the systems, even collusion of all organizations and tapping of all communication lines could not yield enough information to link the pseudonyms – regardless of how clever the attack or how much computation it uses.

In summary, if large scale automated systems for consumer transactions are actually to be built, the new approach offers a far more attractive way to structure them. Its specific advantages to individuals, organizations, and society at large will be argued further in the final section. The intervening three sections expand on its desirability and practicality for a comprehensive set of transaction types: communication, payments, and credentials.

Payment systems now being piloted for widespread use with the current approach include tamper-resistant card computers issued by banks and electronic connections between banks and retailers. The same basic mechanisms, however, could be designed to carry out payment transactions under the new approach. This in turn would allow new approach credential transactions to come naturally and gradually into use, with their applicability and benefits growing as computer and telecommunications infrastructures mature. The communication system proposed here would only begin to be practical with the advent of large-scale consumer electronic mail and would allow home use of the payment and credential systems. It is here presented first, however, since it most clearly

illustrates some concepts central to the latter more immediately applicable systems.

COMMUNICATION TRANSACTIONS

As more messages travel in electromagnetic and digital form, it becomes easier to learn about individuals from their communication. Exposure of message content is one obvious danger, but this is already addressed by well-known coding techniques. A more subtle and difficult problem with current communication systems, however, is the exposure of 'tracing information'. An important kind of tracing information today is individuals' addresses, which organizations often require and which they commonly sell as mailing lists. The trend is toward greater use of such information. Comprehensive computerized data on who calls whom and when, for instance, are increasingly being collected and maintained by telephone companies. Electronic mail systems, some new telephone systems, and the proposed integrated services networks automatically deliver tracing information with each message. When such information is available on a mass basis, the pattern of each individual's relationships is laid bare. Furthermore, tracing information can be used as an identifier to link together all the records on an individual that are held by organizations with whom that individual communicates. So long as communication systems allow system providers, organizations, or eavesdroppers to obtain tracing information, they are unsuitable for the new approach and, moreover, are a growing threat to individuals' ability to determine how information about themselves is used.

The other side of the issue is that current systems offer organizations and society at large inadequate protection against individuals who forge messages or falsely claim not to have sent or received messages. With paper communication, handwritten signatures are easily forged well enough to pass routine checking against signature samples, and they cannot be verified with certainty, even by expert witnesses. Also, paper receipts for delivery are too costly for most transactions, are often based solely on handwritten signatures, and usually do not indicate message content. As computerized systems come into wider use, moreover, the potential for abuse by individuals will increase. Solving these problems under the current approach might be attempted in several obvious ways: by providing recipients with the sender's address, by installing tamper-resistant identity-card readers or the like at every entry point to the communication system, and by keeping records of all messages to allow certification of delivery. But these security measures are all based on tracing information and thus are in fundamental conflict with individuals' ability to monitor and control information about themselves.

Both sets of problems are solved under the new approach. The nature of the solution is such that: individuals are able to send or receive messages without releasing any tracing information; receivers can show that messages were in fact sent to them, despite denial by the senders; senders can show that messages were in fact received, despite denial by the receivers; and message content is kept

confidential. To make messages untraceable, a person's electronic mail computer conceals, in an unconditionally secure way, which messages it sends and receives. To prevent denial by a sender, each sender cryptographically codes messages in a way that each receiver can check, but that prevents anyone from being able to imitate the sender's coded 'signature'. These two concepts – untraceability and coded signatures – will recur intertwined in the payment and credential transaction types and are presented in separate subsections below.

Unconditional Untraceability

It is easy, in principle, to prevent a message sent by an organization from being traced to its individual recipient. The organization simply broadcasts all its messages to all individuals, and each individual's electronic mail computer then scans the broadcasts for messages addressed to any of its owner's pseudonyms. Thus only the individual's computer knows which of the broadcast messages its owner obtains.

Preventing a message sent to an organization from being traced back to its individual sender, however, requires some novel techniques; since any physical transmission can, in principle, be traced to its source. The concept of these techniques is illustrated by a hypothetical situation. Suppose two of your friends invite you to dine at a restaurant. After dinner, the waiter comes to your table and mentions that one of the three of you has already paid for the dinner – but he does not say which one. If you paid, your friends want to know (since they invited you), but if one of them paid, they do not want you to be able to learn which one of them it was.

The problem is solved at the table in the following simple way: Your friends flip a coin behind a menu so that they can see the outcome, but you cannot. It is agreed that each of them will say the outcome aloud, but that if one of them paid, that one will say the opposite of the actual outcome. The uninteresting case is when they both say heads or both say tails: then everyone knows that you paid. If one of them says heads and the other says tails, however, then you know that one of them paid – but you have absolutely no information as to which one. You do know that the one you observed say tails paid if the coin toss was heads, and that the other one paid if the coin toss was tails. But since heads and tails tosses are equally likely, you learn nothing from your two friends' utterances about which one of them paid.

The system described allows the friend who paid to send you an *unconditionally untraceable* message; even though you know who says what, you cannot trace the 'I paid' message, no matter how clever or prolonged your analysis.

This hypothetical system can be generalized and made practical (as detailed in reference [1]). One such generalization uses additional coins to allow more potential senders at the table, while preventing tracing even by collusion. Another breaks long messages into a sequence of parts, each of which is dealt with in a separate round of coin tosses and utterances. In practical communication systems, each participant's electronic mail computer would share secret numeric keys with other mail computers (just as hosts shared coin tosses behind

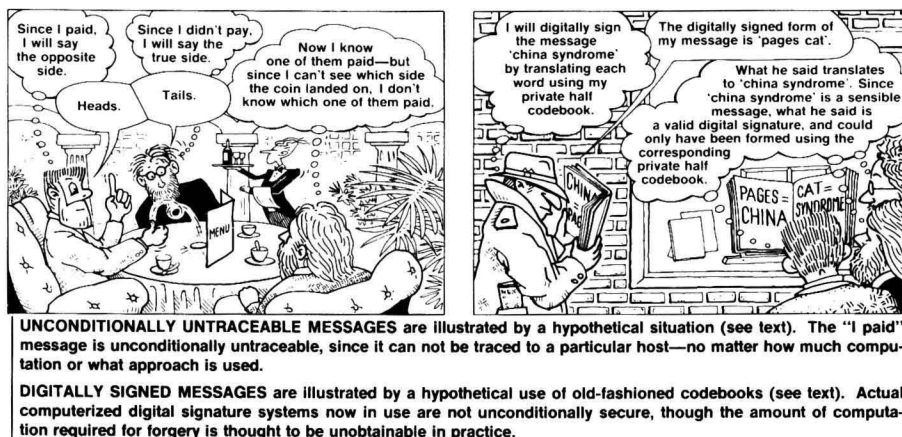
their menus). Each mail computer then uses these keys to produce transformed sequences of digits (like a sequence of outcomes uttered at the table), which it sends through the mail network. The network combines all these transmissions to recover the original messages, which it broadcasts back to the mail computers (just as messages were audible and understandable to everyone at the table).

Digital Signatures

Now consider the problem of preventing senders from later disavowing messages they have sent. The solution is based on the concept of 'digital signatures', which was first proposed by Diffie and Hellman [4]. To see how this concept works, imagine an old-fashioned codebook that is divided into two halves, like an English-French and French-English dictionary, except that only English words are used. Thus, if you look up an English word in the front half of the codebook, you find the corresponding (but usually semantically unrelated) English code word; if you then look this code word up in the back half, you find your original English word. Such codebooks are constructed by pairing off words at random: in the front half of the book, the pairs are ordered by their first words, and in the back half by their second words. For instance, if under 'spy' the front half shows 'why,' then under 'why' the back half shows 'spy'.

If you construct such a codebook, you can use it in your communication with an organization. You keep the front half as your *private key*, and you give the back half to the organization as your *digital pseudonym* with that organization. Before sending a message to the organization, you translate each word of the message into code using your private key; this encoded form of the message is called a *digital signature*. When the organization receives the digital signature from you, it translates it back to the original English message using your digital pseudonym.

The immensely useful property of such digital signatures is their resistance to forgery. No one – not even the organization that has your digital pseudonym



– can easily forge a digital signature of yours. Such forgery would entail creating something that your digital pseudonym decodes to a sensible English message. In the codebook analogy, of course, forgery merely requires searching through (or completely re-sorting by second words) the half of the book that is your digital pseudonym. With actual digital-signature cryptographic techniques currently in use, however, forgery is thought to require so much computation as to be infeasible even for the fastest computers working for millions of years. If an organization cannot forge a digital signature of yours, then it cannot successfully claim that you sent it a message that you in fact did not send. A third-party arbiter would decide in favor of an organization only if the organization could show a digital signature that yields the disputed message when translated with your digital pseudonym. But, because forgery is infeasible, the organization could obtain such a digital signature only if you had ‘signed’ (i.e., encoded) the disputed message using your private key.

An organization could create its own private key and corresponding digital pseudonym (its own ‘codebook’); it would keep the private key (the front half) to itself, while widely disseminating the corresponding digital pseudonym (the back half). It would then use this private key to transform messages into digital signatures before sending them to individuals. The organization, unlike an individual, would create only a single private key and corresponding digital pseudonym, which it would use for all digital signatures it sends. Thus, anyone receiving a signed message from the organization would decode it using the organization’s single, publicly disseminated digital pseudonym (commonly called a ‘public key’). These signatures would allow individuals to convince the organization, or anyone else if necessary, that the message had in fact been sent by the organization. In the payment and credential systems introduced in the following sections, such digital signatures formed by organizations play an important role.

Digital Signatures in Practice

Actual digital signatures are realized using numbers, and can be adapted to keep message content confidential and to certify delivery.

Practical, computerized digital-signature techniques work just as in the codebook analogy above, except that everything is done with twohundred-digit numbers. Each private key, and each digital pseudonym, is represented as one such number (rather than as a half codebook); each unsigned message and each signature is also represented as such a number (rather than as a string of English words). A standard, publicly available mathematical procedure lets anyone use a private key to form a corresponding digital signature from a message; a similar procedure allows anyone to recover the original message using the matching digital pseudonym (just as the simple procedure for looking up words in either half of the codebook can be public, so long as the private key is not). Another public mathematical procedure allows anyone to create a private key and corresponding digital pseudonym from a random starting point (just as the two halves of a codebook could be generated from a random pairing of words).

Messages are kept confidential during transmission by using digital pseudonyms and private keys in a different way: before transmitting a message, the sender first signs it and then encodes the result using the digital pseudonym of the intended recipient. Thus, the signed message can be recovered only by decoding the transmission using the intended recipient's private key.

A cartoon illustrating the steps of a quantum algorithm. Two customers are seated at a table with a menu. The first customer's thought bubble shows $[1] \leftarrow k + 1$. The second customer's thought bubble shows $[2] \leftarrow k$. The waiter's thought bubble shows $[1] \oplus [2] - 1$.

DIGITAL SIGNATURES WITH NUMBERS use special arithmetic systems, in which raising a number to a power \bar{x} scrambles it, and raising to a corresponding power x unscrambles it: $(m^{\bar{x}})^x = m$. (The power \bar{x} acts as the private half codebook, and the other power x acts as the corresponding half.) First the message is encoded as a one-hundred-redigit number, and then the digits are repeated to form a two-hundred-digit number m with this special repeated-halves property. Next the signer raises the special number to a private power \bar{x} and makes the result known to others in transmission [1]. Someone receiving this digitally-signed message merely raises it to the corresponding digital-pseudonym power x and checks that the result has the special repeated-halves property. If it does, then the recipient knows that the message was signed by the holder of the corresponding private power.

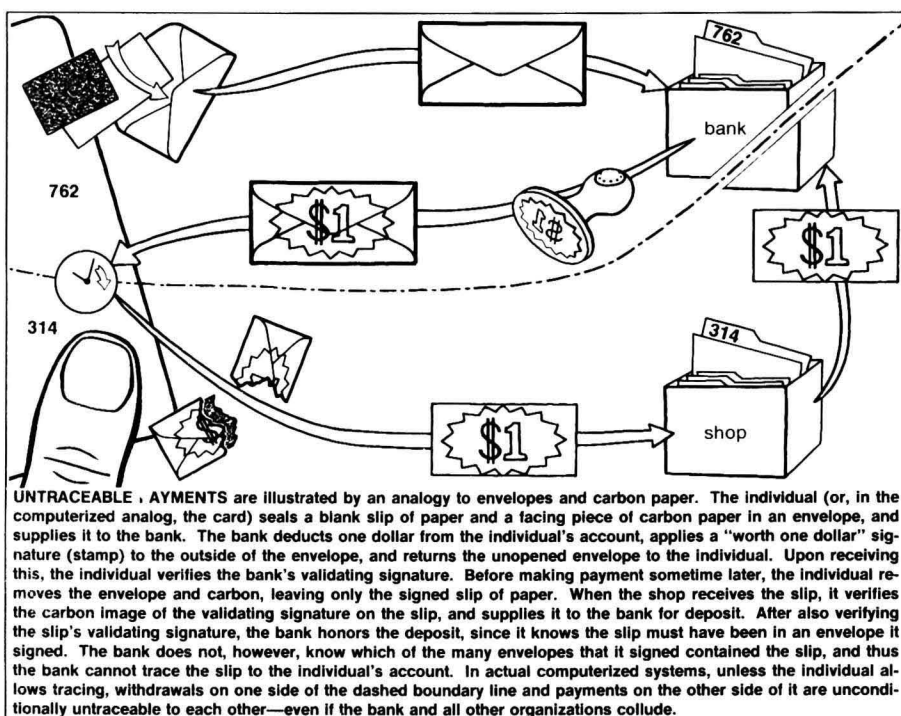
The computerization of payments is giving payment system providers and others easy access to extensive and revealing information about individuals through payments made for purchases from shops, subscriptions, donations, travel, entertainment, professional services, and so on. Today, many paper records of when, how much, from whom, and to whom payment was made are translated into electronic form. The trend is toward capturing this payment data electronically, right at the point of sale. This facilitates the electronic capture of the potentially more revealing details of what was purchased. Moreover, computerization is extending the data capture potential of payment systems in other ways. One is through emerging informational services like pay television and videotex; another is through new systems that directly connect central billing computers to things like electric-utility meters and automobile-identification sensors buried in toll roads. Just as, in communication systems, tracing information links all of an individual's records with organizations, payment data containing an account identifier links all of an individual's relationships involving payments.

From the other perspective, it is widely held that uncollectible payments made by consumers, such as credit card misuse and checks drawn against insufficient funds, cost society billions of dollars a year. Paper banknotes are vulnerable to counterfeiting and theft, and their lack of auditability makes them convenient for illicit payments such as bribes, extortion, and black-market purchases. Limiting all these abuses while automating seems to call for highly pervasive and interlinked systems that capture and retain account identifiers as well as other payment data – which is in clear conflict with the interests of individuals.

The nature of the new approach's solution to these problems ensures that organizations, even colluding with the payment system provider who maintains the accounts, cannot trace the flow of money between accounts. But the system provider does know the balance of each account, and if funds were to be transferred between accounts instantaneously, the simultaneous but opposite changes in balance would make tracing easy. Such tracing is prevented because funds are withdrawn, held, and paid as multidenominational notes, in some ways like 'unmarked bills'. These notes are unlike paper banknotes, however, in that individuals, but not organizations, can allow transfers to be traced and audited whenever needed; this makes the notes unusable if stolen, and unattractive for many kinds of illicit payments. The fully computerized systems introduced here offer practical yet highly secure replacements for most current and proposed consumer payment systems (as detailed in [2]).

Blind Signatures for Untraceable Payments

The new-approach payment systems are based on an extension of digital signatures, called *blind signatures*. This concept is illustrated by an analogy to carbon-paper-lined envelopes. If you seal a slip of paper inside such an



envelope and a signature mark is later made on the outside, then when you open the envelope, the slip will bear the signature mark's carbon image.

Consider how you might use such an envelope to make a payment. Suppose that a bank has a special signature mark that it guarantees to be worth one dollar, in the sense that the bank will pay one dollar for any piece of paper with that mark on it. You take a plain slip of paper sealed in a carbon-lined envelope to the bank and ask to withdraw one dollar from your account. In response, the bank deducts one dollar from your account, makes the signature mark on the outside of your envelope, and returns it to you. You verify that your sealed envelope has been returned with the proper signature mark on it. Later, when you remove the slip from the envelope, it bears the carbon image of the bank's signature mark. You can then buy something for one dollar from a shop, using the signed slip to make payment. The shop verifies the carbon image of the bank's signature on the slip before accepting it.

Now consider the position of the bank when the slip is received for deposit from the shop. The bank verifies the signature on the slip submitted for deposit, just as the shop did, and adds a dollar to the shop's account. Because the signature verified, the bank knows that the slip must have been in an envelope that it signed. But naturally the bank uses exactly the same signature mark to sign many such envelopes each day for all of its account holders, and since all slips were 'blinded' by envelopes during signing, the bank cannot know which envelope the slip was in. Therefore it cannot learn from which account the

funds were withdrawn. More generally, the bank cannot determine which withdrawal corresponds to which deposit – the payments are untraceable.

In actual computerized systems, both slips and envelopes are replaced by numbers, the bank's signature mark becomes a digital blind signature, and payments are unconditionally untraceable (as described later in this section). The protocols for transacting withdrawals and payments would of course be carried out automatically by the card computer; its owner would merely have to allow each transaction by entering the secret authorizing number.

Extending the Envelope Analogy

Using *note numbers* provides protections similar to those offered by check numbers today. Since the bank is unable to see into the envelopes, nothing is revealed to the bank by a randomly chosen note number written on the slip before it is signed. (Alternatively, the slip's unique, random paperfiber pattern could represent the note number.) Stolen notes should not be accepted by the bank once the individual who withdrew the funds reports their note numbers. When given these numbers, the bank can also attest to the accounts to which funds have been deposited. Such traceability at the payer's initiative would discourage the use of these systems in bribery, extortion, black market purchases, and other illicit payments: recipients of such payments risk having their accounts traced if they deposit the notes, and being apprehended or just discovering that the notes are worthless if they try to spend them.

A variation prevents organizations (even colluding with banks) from tracing the accounts of individuals to whom they pay such things as wages, settlements, refunds, and rebates. The individual places a slip in an envelope as before and gives it to the paying organization, which then supplies this blinded slip to the bank. The bank, without knowing which individual is involved, signs the envelope and charges the paying organization's account. Signed but still blinded, the slip is returned by the organization to the individual, who verifies the signature, and later removes the envelope and deposits the slip with the bank.

Other extensions to the basic concept offer replacements for today's payment systems attractive to both financial institutions and consumers. Regional clearing and signing centers would handle most of the work and responsibility for banks on a wholesale basis, while the banks could offer their own customized services. Different signatures would be used for different denominations. An adaption allows routine transactions to be consummated in a way not requiring immediate or online interaction with a bank. Further variations permit the payment system to be used just as credit and debit cards are used today, with interest charges for credit and interest earnings on unspent debitcard balances.

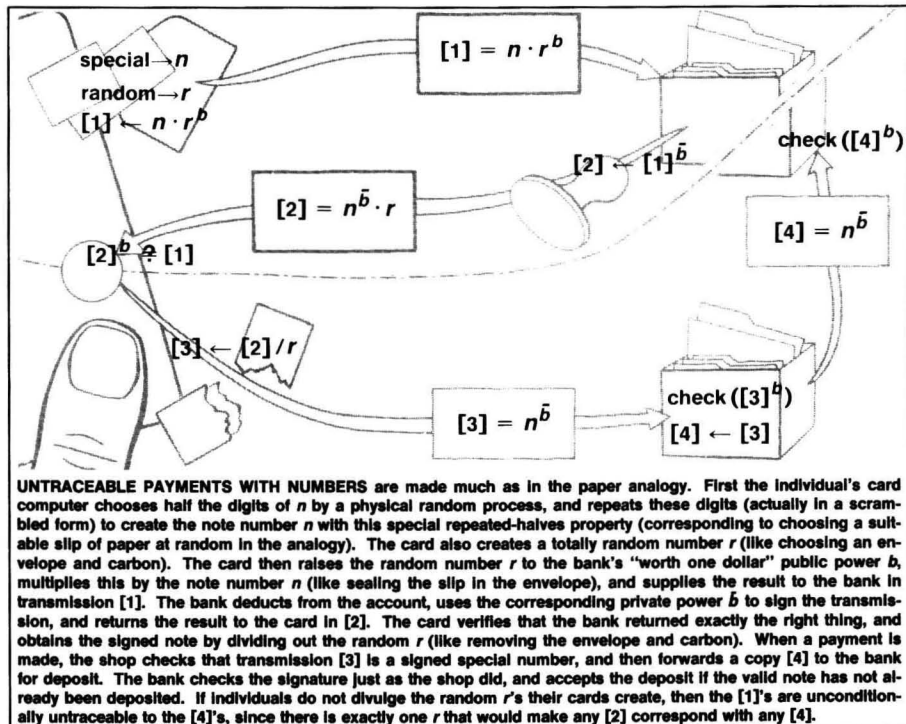
Leaving the Analogy

Actual payment systems would work very much along the lines of the envelope analogy, except that they use no paper, only numbers. A note number is first created by a true random process within the individual's card computer (used like the random number or fiber pattern on the slip of paper). Next, the

card computer transforms the note number into a numeric note that is the equivalent of the message: 'This is note number: 59..2' (used like the slip of paper itself). The card computer then blinds this numeric note by combining it with a second random number (like the payer choosing an envelope at random and placing the slip in it). During withdrawal, the bank uses the private key of the desired denomination to form a digital signature on the blinded numeric note (like the signature mark made on the envelope). When the signed but still blinded note is returned, the card computer is able to unblind it by a process that removes the random blinding number from the digital signature while leaving the signature on the note (like the payer removing the envelope). Both the organization receiving payment and the bank use the bank's digital pseudonym to decode the signature; if the result is an appropriate message, this verifies the note's digital signature.

A conceivable danger for the bank is that the same numeric note might be deposited more than once. To prevent this, a list of note numbers accepted for deposit is maintained and only note numbers not already on the list are accepted and recorded. The cost of maintaining such a list can be far less per transaction than the transaction cost of current payment systems, since expiration dates built into note numbers allow old numbers to be deleted from the list.

Another conceivable danger is that the bank's digital signature could be forged, which would allow counterfeiting. The security against this kind of threat is based on the underlying digital-signature cryptographic technique, which is



currently being proposed as an international standard and is already used by banks and even by nuclear agencies. The odds of someone guessing a valid, signed numeric note, or of any two independently chosen note numbers being the same in the foreseeable future, are less than 1 in 10 to the 75th power.

The numeric notes are unconditionally untraceable: the bank cannot learn anything from the numbers about the correspondence between withdrawals and deposits. In the hypothetical restaurant situation, both outcomes of each coin toss were equally likely, which meant that every correspondence between senders and messages was equally likely. Similarly, because all suitable numbers are equally likely to be used for the independent blinding of each note, all correspondences between withdrawals and deposits are equally likely.

CREDENTIAL TRANSACTIONS

In their relationships with many organizations, there are legitimate needs for individuals to show credentials. The term 'credentials' is used here to mean statements concerning an individual that are issued by organizations, and are in general shown to other organizations. In the past, credentials primarily took the form of certificates like passports, driver's licenses, and membership cards. Before computerization, such certificates provided individuals with substantial control over access to their credentials, though the certificates also often revealed unnecessary and identifying information like address, birthdate, and various numbers. Today, such identifying information is being used to link records on certificate holders; it even allows them to be 'blacklisted' or denied services because of reports from organizations that may be erroneous, obsolete, or otherwise inappropriate for the decision at hand. Where no substantiating certificate is required to be shown, as with application or tax forms, much similarly unnecessary or overly detailed information is demanded, presumably to allow confirmation. But confirmation itself can link further information and lead back to inappropriate records. The control over credential information that certificates once provided to individuals is thus being circumvented and rendered illusory by computerization.

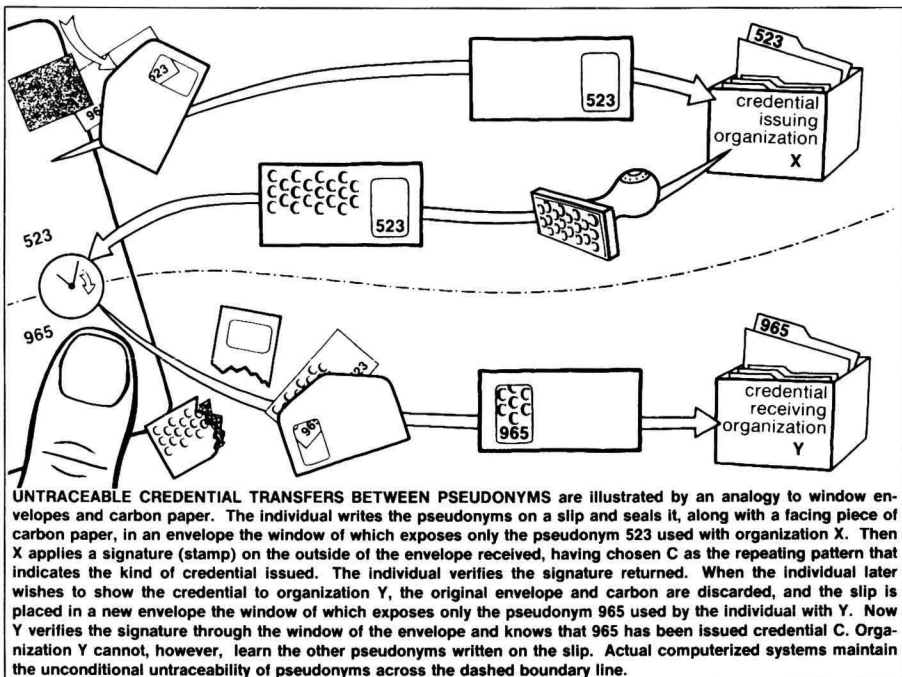
The countervailing problem is that credentials are subject to widespread abuse by individuals, who can easily modify or copy many kinds of paper and plastic certificates with today's technology. This is one reason why certificates are in effect being reduced to the role of providing identifying information, and organizations are maintaining the credentials themselves. To check on unsubstantiated credential information, organizations are also rapidly deploying so-called matching techniques, whereby they use identifying information to link and share records on individuals. Many organizations may also need the ability to blacklist individuals or to determine whether they are already blacklisted. As the number of such organizations grows, certificates or even matching techniques become impractical, hence the creation of large centralized databases on individuals. The use of multiple complete identities by sophisticated criminals is a related problem. As with communication and payments, the obvious

countermeasures under the current approach – widespread use of highly secure identity documents linked to centrally maintained credentials – are in direct conflict with individuals' ability to determine how information about themselves is used.

With the new approach's solution, an individual can transform a specially coded credential issued under one pseudonym into a similarly coded form of the same credential, which can be shown under the individual's other pseudonyms. Since these coded credentials are maintained and shown only by individuals, they return control similar to that formerly provided by certificates; and since they are convenient to use, they obviate the need for unsubstantiated credentials and for matching. Individuals can also tailor the coded form they show to ensure that only appropriate information is revealed or used to make particular decisions, and can ensure that obsolete information becomes unlinkable to current pseudonyms. Abuses of credentials by individuals, such as forgery and improper modification or sharing, are prevented by the cryptographic coding and the protocols for its use. Since each person is able to have at most one pseudonym with any organization requiring such protection, multiple complete identities are also prevented. Moreover, accountability for abuses perpetrated under any of an individual's pseudonyms can still be assured, without the need for centralized databases.

The Basic Credential System

The essential concept is again illustrated by analogy to carbon-lined envelopes, only this time the envelopes have windows. First, you make up



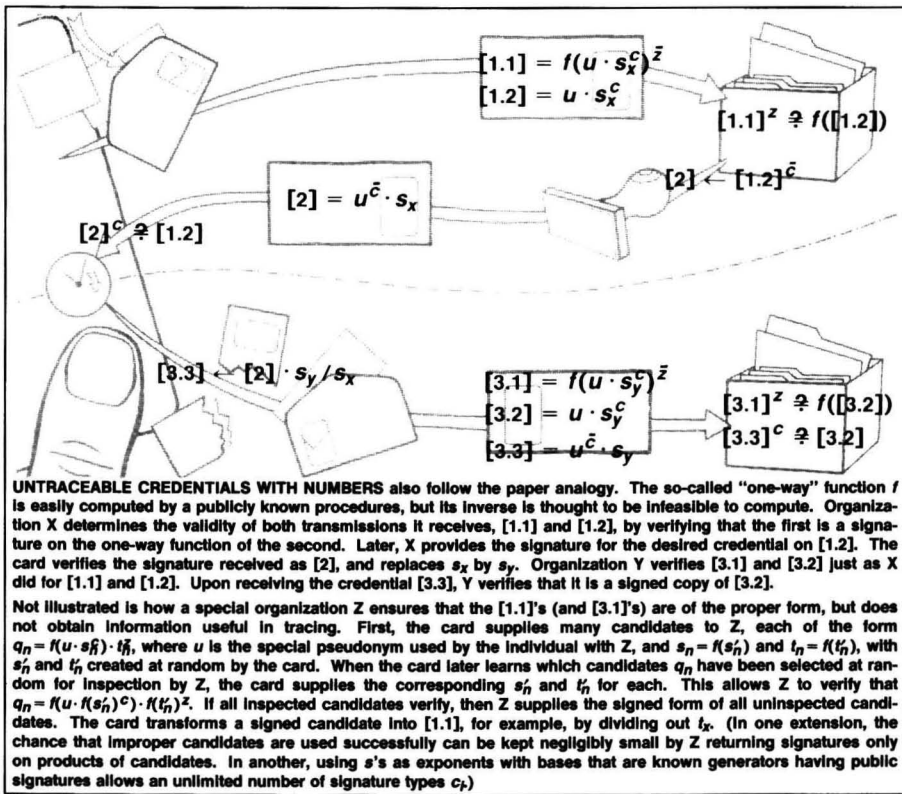
numeric pseudonyms at random and write them on a plain slip of paper. When you want to get a credential from an organization, you put the slip in a carbon-lined envelope with a window exposing only the pseudonym you use with that organization. Upon getting the envelope from you, the organization makes a special signature mark in a repeating pattern across the outside of it, and the carbon lining transfers the pattern to the slip. This signature pattern is the *credential*; the type of pattern corresponds to the kind of credential the issuing organization decides to give you, according to the pseudonym they see through the window. When you get the envelope back from the issuing organization, you verify the credential signature pattern. Before showing the credential to another organization, you place the slip in a different envelope with a window position that exposes only the pseudonym you use with that organization, along with some of the adjacent credential signature pattern. The receiving organization can verify, through the window, the pseudonym you use with it as well as the signature pattern. In this way, you can obtain and show a variety of credentials.

An organization can ensure that no individual is able to transact with it under more than one pseudonym. One way an individual could attempt to use more than a single pseudonym with an organization is to use different pseudonyms on the same slip of paper. This is prevented by a standard division of the slip into positional zones, such that each zone is assigned to a particular organization; an envelope is accepted by an organization only if the window position exposes that organization's zone, bearing a single indelibly written pseudonym. A second way of attempting to use more than one pseudonym per organization is to use more than one slip. This is prevented by the establishment of an agency that issues a single 'is-a-person' credential signature to each individual. Other organizations accept only envelopes with this signature recognizable through the window. The agency ensures that it issues no more than one signature per person by taking, say, a thumbprint and checking that the print is not already on file before giving the signature. This collection of prints poses little danger to individuals, however, since the prints cannot be linked to anything.

The pseudonyms used by individuals are untraceable, in the sense that envelopes give no clue, apart from the signatures shown, about the other randomly chosen pseudonyms they contain. Actual systems based on card computers would provide unconditional untraceability using digital blind signatures on numbers (as detailed in [3]).

Revealing Only Necessary Information

You need not show all your credentials to every organization; you can restrict what you show to only what is necessary. Because of the way the credential signature patterns repeat across the slips, a recognizable part of each signature pattern appears adjacent to each pseudonym. To prevent certain credentials from being seen, though, you could simply black out parts of an envelope's window when showing it to an organization. But more flexible restrictions are possible using your card computer. It serves as the single database of all your



credentials – and you alone control which queries from organizations it answers.

A typical such query might be: ‘Does the owner of pseudonym 72..4 have credentials sufficient to meet the requirement: ...?’ Your card can issue a convincing affirmative response only when it does in fact have credential signatures satisfying the requirement. But the card ensures – unconditionally – that organizations cannot learn any more about your credentials from its responses than the affirmations themselves. You might use it to convince an organization that your age, income, and education, for instance, meet their entry requirements in at least one way, without revealing any more than just that fact. Or, when a survey requires credentials for substantiating responses, using a different pseudonym for each response ensures that no more is revealed than the total number of each type of response.

Actual queries and responses can be realized as follows: an organization encodes a new credential into the query message itself, in such a way that the credential can be decoded using any one of several qualifying combinations of other credentials as the key. If any qualifying combination is held, then this new credential can be decoded and shown to the organization as the response. It can also be retained for later use, which additionally permits the gradual replacement of older and more detailed credentials by more appropriate summary ones. When such query messages are made public so that everyone can use

them, they provide for public and verifiable rules for decisions about individuals.

Some Uses of Credentials

The new approach supports most varieties of credentials used today. Some of these, like educational degrees, are lifelong, while others, like student cards, are valid only for prescribed periods. Still others, like membership cards, usually have long-term validity, but their certificates typically expire at the end of each year, thereby allowing their issuers to effectively revoke the credential by withholding new certificates.

A less common but still used kind of credential allows organizations in effect to blacklist individuals, without maintaining a central list of identities. Suppose, for example, that credentials are issued for filing tax forms, so that each adult citizen should get such a credential every year. Organizations might routinely modify their queries to include the requirement that adult citizens have filed tax forms for the last year. This would *blacklist* those who had not complied by barring them from relationships with organizations.

In actual widespread use, where many organizations may occasionally need to blacklist some individuals, such a mechanism is neither practical nor desirable: queries would have to demand vast numbers of credentials, while individuals would be unable to protect themselves against being blacklisted by organizations even with which they have had no contact.

Authorized Blacklisting Without Lists

These problems of wider use can be solved by techniques that require an organization to obtain, directly from an individual, the authorization to blacklist that individual for a specified reason. Organizations would insist on such authorizations as are appropriate before establishing or extending relationships.

The way these techniques work is illustrated by applying the envelope analogy to buying goods on credit. A special row of zones is reserved on each slip for this purpose. You provide the shop where you make the credit purchase with an envelope that has (in addition to any window you may ordinarily use with that shop) a window exposing one of these reserved zones. The shop first broadcasts the numeric pseudonym it sees indelibly written in that reserved zone, so that when no other organization objects, the shop is assured exclusive use of that zone.

When you later pay the shop, it gives you a *resolution* credential signature mark; unlike the credential signature marks previously described, it is made only on the single zone to which it applies. If some of the reserved zones remain unused, you can show them to a 'voiding' agency that obtains exclusive use of these unneeded zones in the same way as do shops, and then issues a resolution signature mark on each.

Only when you repay by deadline all due loans can you obtain resolution signature marks on each zone of the reserved row. Then you can demonstrate

that you are not blacklisted, without revealing more, just by showing that all of your reserved zones have their resolution signatures. You do this by presenting an envelope that has a slit-shaped window positioned over the reserved row. It exposes only a narrow band of each reserved zone's resolution credential signature, while concealing the pseudonym-bearing parts of the zones that were shown separately to lenders or the voiding agency. In actual systems, card computers would obtain and show digital signatures for this purpose as part of their general management of the reserved row.

Preventing Use of Untimely Information

The mechanisms of the new approach can both guarantee individuals time to review credential information before it is required, and unconditionally ensure them the ability to shed such information once it is outdated.

If individuals can expect to receive their resolution credentials some 'cooling-off' interval before they are needed, instead of at the last minute, then there may be time to resolve errors or disputes before any unnecessary consequences occur. Organizations may not wish to increase the maximum delay before blacklisting takes effect, but some cooling-off interval can always be provided without doing so. For example, when a different resolution credential is valid for each calendar month and organizations provide them just before the beginning of the month, then the maximum delay before blacklisting takes effect is one month and there is no cooling-off interval. But this same maximum delay can be maintained while providing cooling-off intervals half a month long: twice a month, organizations issue credentials that expire a month after their issue date, so that a credential remains valid for a half-month interval following the scheduled issue of its successor.

If individuals change pseudonyms periodically, they cannot be linked to obsolete information. The initial information associated with new pseudonyms would be provided through the transfer of credentials from previous pseudonyms. The changeovers could be staggered to allow time for completion of pending business.

There are additional benefits to changing pseudonyms beyond the weeding-out of obsolete information. For one thing, the periodic reduction to essentials prevents organizations from gradually accumulating information that might ultimately be used to link pseudonyms. Moreover, for individuals to be able to transfer all the initial information for a period, they must know each organization's information demands, they must know where each piece of information comes from, and they must consent to each such transfer. Information linkable by each organization is thus known to and agreed on by individuals – that is, individuals can monitor and control it.

MICRO- AND MACROCOMPARISONS

Advantages to Individuals

As the public becomes more aware of the extent and possibilities of emerging

information technology, there should be a growing demand for the kinds of systems described here. Compared to the current approach, individuals stand to gain increased convenience and reliability; improved protection against abuses by other individuals and by organizations; monitorability and control; and full access to transaction systems.

Increased convenience derives from the freedom of individuals to obtain their card computers from any source, to use whatever hardware or software they choose, and to interface with communication systems wherever they please. This permits card computers to be adapted to the requirements of sophisticated, naive, and handicapped users alike. The systems need be no more complicated to use than under the current approach; people might choose never to actually see their pseudonyms or to be concerned with other implementation details.

The individual is ensured reliable system access by a numeric key with which the card computer encodes backup copies of its contents, and which allows a replacement card to recover these contents. Since this key should be 40 or more digits in size, it might be impractical for its owner to remember. Known techniques allow the key to be divided into parts, each of which can be given to a different trustee. This provides certain subsets of the trustees with the ability to recover the key, while insufficient subsets would be unable to learn anything about it. Still other subsets, given parts of the owner's secret authorizing number, would be able to take over the owner's affairs when needed. These provisions are an example of how an individual's power to designate proxies, a power now enjoyed by organizations, is ensured.

Abuse of a lost or stolen card computer by another individual would be very difficult without the owner's secret authorizing number, as asserted earlier. This is because the card would require the authorizing number, which might typically be about six digits long, before allowing transactions. A reasonably tamper-resistant device within the card computer could: read fingerprints or the like to prevent use by anyone but the card owner; accept a special authorizing number that the owner could use in case of duress to trigger a prearranged protective strategy; and permit only the current owner to reset the card for a new owner, to prevent its use as a replacement by a thief. Even if sophisticated criminals were to extract the card's information content, and the owner were not to cancel in time using backup data, a great many guesses at the authorizing number might have to be tried with organizations before the actual number could be determined. This would make such attacks very likely to be detected and to fail.

The new approach protects individuals unconditionally from abuses by organizations, such as the false attribution of messages, and from organizations blacklisting without advance warning. Moreover, individuals are provided with secure relationships without ever having to sacrifice the protection of their pseudonyms by revealing linking information – but they can always do so if they choose. While it is relatively easy for individuals to provide convincing evidence only of their role in particular transactions, it is even possible for them to provide evidence that they were not involved in certain other transactions.

For example, in communication transactions, individuals could show that their physical entry to the system was not used to send a particular message; in payment transactions, they could show that a payment did not involve their account; and in credential transactions, they could show that a pseudonym was not among the set obtainable under their thumbprint.

The primary way that individuals gain monitorability and control is through their ability to prevent linking. Some linking of separate relationships might occur if, for instance, a consumer actually wanted to be recognized, or as part of an investigation or other exceptional situation. But the linking of some relationships does not, in general, allow others to be linked, and the regular changing of pseudonyms allows linkings to be shed over time. In addition, the scope of an individual's separate, unlinkable relationships need not depend on the legal or administrative structure of the organizations involved; an individual might use the same pseudonym with different organizations or, when allowed, different pseudonyms with the same organization. Naturally, the scope of relationships, along with such things as the level of detail in credentials and the frequency of pseudonym changeover, must be adjusted to provide the desired degree of protection against inference by statistical or pattern recognition techniques. Such protections would likely create a widespread expectation of control over information; thus, as similar expectations have done in the past, it might also engender commensurate legal safeguards.

Individuals would have the same access to systems as organizations, in addition to enjoying the same protections; such parity is precluded under the current approach in efforts to protect the security of organizations. A new-approach payment, for example, could be made between two friends using their card computers. A small business would even be able to handle all customer transactions, using only a card computer.

Advantages to Organizations

Organizations have an interest in cultivating the goodwill of individuals. But they gain further direct benefits from the advantages to individuals described earlier, since in making their own transactions, they have many of the same concerns as individuals. Moreover, the new approach offers them reductions in cost; reductions in the quantity and sensitivity of necessary data; and improved security against detectable, undetectable, and extrasystemic abuses.

The systems described here would be less costly for organizations than comparable systems based on the logical extension of the current approach. This is primarily because the latter requires widely trusted, tamper-resistant devices at all points of entry to transaction systems. Such a requirement implies substantial initial agreement, outlay, and commitment to design, and can be expected to result in technology that is outdated when systems come into widespread use. Furthermore, the tamper resistance techniques currently contemplated require significant compromise in security, even at high cost. The new-approach system provider need not supply user organizations with tamper-resistant terminal equipment for each entry point, any more than it need supply

card computers to individuals. Thus, user organizations can supply their own terminal equipment wherever they please and take advantage of the latest technology. Although these cards and terminals make more sophisticated use of cryptographic techniques than does equipment envisioned under the current approach, this difference between the two is just a fraction of a chip in the technologies of the near future.

The new approach reduces the sensitivity and the quantity of consumer data in the hands of organizations; by the same token, it reduces their exposure to incidents that might incur legal liability or hurt their public images. Reductions in data could also streamline operations, and the increased appropriateness of the remaining data could provide a better basis for decision making. As electronic mail replaces paper mail, individuals' computers may routinely reject unsolicited commercial messages and instead seek out only desired information. Thus, data for targeting such messages might become superfluous even under the current approach. The new approach's protections, however, may compensate by making individuals less reluctant to provide information for surveys and the like.

Under either approach, if an automated transaction system detects sufficiently serious abuse or default by an individual, the best it can do is to lock that individual out. This is because the individual can always step outside such a system's controls by 'going underground'. The new-approach systems can lock individuals out, but can also have a cooling-off interval built in to allow matters to be resolved before lockout is needed. The approach also reduces the need for such measures, however, since its mechanisms allow organizations or society at large the flexibility to set policy that establishes a desired balance between prior restraint, as in the basic payment system, and accountability after the fact, as with credit or other authorized blacklisting functions.

Undetectable abuse by individuals acting alone seems to be precluded by the systems of the new approach. But no transaction system is able to detect an individual who obtains something through legitimate use of the system and then transfers it to another person by some means outside the system. Transferring the ability to use a communication system to others is an instance of the proxy power already mentioned, which could be inhibited under the current approach. In the context of the payment system, such transfers can be treated as illicit payments, which are deterred by the use of note numbers. The credential system directly prevents the transfer of credentials from the pseudonyms of one person to those of another. Currently, 'in-person' proxy is prevented by certificates bearing photos. Such photo tokens could still be used with the new approach, if and when needed; but they might include only a photo, an indication of the kind of credential, and possibly a digital pseudonym.

Meanwhile, it is too easy to step outside current transaction systems by using coin phones, sending anonymous letters, dealing in cash, and using false credentials. Significantly improved security, particularly against more sophisticated abuse, can only be obtained with comprehensive automated systems. But such systems under the current approach may meet with broad-

based resistance from individuals – especially once they become aware of the alternatives posed by the new approach.

Implications for the Future

Large-scale automated transaction systems are imminent. As the initial choice for their architecture gathers economic and social momentum, it becomes increasingly difficult to reverse. Whichever approach prevails, it will likely have a profound and enduring impact on economic freedom, democracy, and our informational rights.

Restrictions on economic freedom may be furthered under the current approach. Markets are often manipulable by parties with special access to information about other participants' transactions. Information service providers and other major interests, for example, could retain control over various information and media distribution channels while synergistically consolidating their position with sophisticated marketing techniques that rely on gathering far-reaching information about consumers. Computerization has already allowed these and other organizations to grow to unprecedented size and influence; if continued along current lines, such domination might be increased. But the computerization of information gathering and dissemination need not lead to centralization: integrating the payment system presented here with communication systems can give individuals and small organizations equal and unrestricted access to information distribution channels. Moreover, when information about the transactions of individuals and organizations is partitioned into separate, unlinkable relationships, the trend toward large-scale gathering of such information, with its potential for manipulation and domination of markets, can be reversed.

Attempts to computerize under the current approach threaten democracy as well. They are, as mentioned, likely to engender widespread opposition; the resulting stalemate would yield security mechanisms incapable of providing adequate prior restraint, thus requiring heavy surveillance, based on record linking, for security. This surveillance might significantly chill individual participation and expression in group and public life. The inadequate security and the accumulation of personally identifiable records, moreover, pose national vulnerabilities. Additionally, the same sophisticated data acquisition and analysis techniques used in marketing are being applied to manipulating public opinion and elections as well. The opportunity exists, however, not only to reverse all these trends, by providing acceptable security without increased surveillance, but also to strengthen democracy. Voting, polling, and surveys, for example, could be conveniently conducted via the new systems; respondents could show relevant credentials pseudonymously, and centralized coordination would not be needed.

The new approach provides a practical basis for two new informational human rights that is unobtainable under the current approach. One is the right of individuals to parity with organizations in transaction system use. This is established in practice by individuals' parity in protecting themselves against

abuses, resolving disputes, conferring proxy, and offering services. The other is the right of individuals to disclose only the minimum information necessary: in accessing information sources and distribution channels, in transactions with organizations, and – more fundamentally – in all the interactions that comprise an individual's informational life.

Advances in information technology have always been accompanied by major changes in society: the transition from tribal to larger hierarchical forms, for example, was accompanied by written language, and printing technology helped to foster the emergence of large-scale democracies. Coupling computers with telecommunications creates what has been called the ultimate medium – it is certainly a big step up from paper. One might then ask: To what forms of society could this new technology lead? The two approaches appear to hold quite different answers.

Acknowledgements

The author is pleased to thank Jan-Hendrik Evertse, Wiebren de Jonge, and Ronald L. Rivest for discussions during the early development of some of the ideas herein presented, as well as everyone who showed interest in and commented on this work.

REFERENCES

- Chaum, D. – The dining cryptographers problem: Unconditional sender and recipient untraceability. Available from the author.
- Chaum, D. – Privacy protected payments: Unconditional payer and/or payee untraceability. Available from the author.
- Chaum, D. – Showing credentials without identification: Transferring signatures between unconditionally unlinkable pseudonyms. Available from the author.
- Diffie, W. and M.E. Hellman. – New directions in cryptography. *IEEE Trans. Inf. Theory*, IT-22, 644–654. (November 1976)
- Rivest, R., A. Shamir and L. Adleman. – A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21, 2, 120–126. (February 1978).