

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

Abstract—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.

I. INTRODUCTION

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade cryptographic devices down to where they can be used in such commercial applications as remote cash dispensers and computer terminals. In turn, such applications create a need for new types of cryptographic systems which minimize the necessity of secure key distribution channels and supply the equivalent of a written signature. At the same time, theoretical developments in information theory and computer science show promise of providing provably secure cryptosystems, changing this ancient art into a science.

The development of computer controlled communication networks promises effortless and inexpensive contact between people or computers on opposite sides of the world, replacing most mail and many excursions with telecommunications. For many applications these contacts must be made secure against both eavesdropping and the injection of illegitimate messages. At present, however, the solution of security problems lags well behind other areas of communications technology. Contemporary cryptography is unable to meet the requirements, in that its use would impose such severe inconveniences on the system users, as to eliminate many of the benefits of teleprocessing.

Manuscript received June 3, 1976. This work was partially supported by the National Science Foundation under NSF Grant ENG 10173. Portions of this work were presented at the IEEE Information Theory Workshop, Lenox, MA, June 23–25, 1975 and the IEEE International Symposium on Information Theory in Ronneby, Sweden, June 21–24, 1976.

W. Diffie is with the Department of Electrical Engineering, Stanford University, Stanford, CA, and the Stanford Artificial Intelligence Laboratory, Stanford, CA 94305.

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

Section III proposes two approaches to transmitting keying information over public (i.e., insecure) channels without compromising the security of the system. In a *public key cryptosystem* enciphering and deciphering are governed by distinct keys, E and D , such that computing D from E is computationally infeasible (e.g., requiring 10^{100} instructions). The enciphering key E can thus be publicly disclosed without compromising the deciphering key D . Each user of the network can, therefore, place his enciphering key in a public directory. This enables any user of the system to send a message to any other user enciphered in such a way that only the intended receiver is able to decipher it. As such, a public key cryptosystem is a multiple access cipher. A private conversation can therefore be held between any two individuals regardless of whether they have ever communicated before. Each one sends messages to the other enciphered in the receiver's public enciphering key and deciphers the messages he receives using his own secret deciphering key.

We propose some techniques for developing public key cryptosystems, but the problem is still largely open.

Public key distribution systems offer a different approach to eliminating the need for a secure key distribution channel. In such a system, two users who wish to exchange a key communicate back and forth until they arrive at a key in common. A third party eavesdropping on this exchange must find it computationally infeasible to compute the key from the information overheard. A possible solution to the public key distribution problem is given in Section III, and Merkle [1] has a partial solution of a different form.

A second problem, amenable to cryptographic solution,

ness communications by teleprocessing systems is authentication. In current business, the validity of contracts is guaranteed by signatures. A signed contract serves as legal evidence of an agreement which the holder can present in court if necessary. The use of signatures, however, requires the transmission and storage of written contracts. In order to have a purely digital replacement for this paper instrument, each user must be able to produce a message whose authenticity can be checked by anyone, but which could not have been produced by anyone else, even the recipient. Since only one person can originate messages but many people can receive messages, this can be viewed as a broadcast cipher. Current electronic authentication techniques cannot meet this need.

Section IV discusses the problem of providing a true, digital, message dependent signature. For reasons brought out there, we refer to this as the one-way authentication problem. Some partial solutions are given, and it is shown how any public key cryptosystem can be transformed into a one-way authentication system.

Section V will consider the interrelation of various cryptographic problems and introduce the even more difficult problem of trap doors.

At the same time that communications and computation have given rise to new cryptographic problems, their offspring, information theory, and the theory of computation have begun to supply tools for the solution of important problems in classical cryptography.

The search for unbreakable codes is one of the oldest themes of cryptographic research, but until this century all proposed systems have ultimately been broken. In the nineteen twenties, however, the "one time pad" was invented, and shown to be unbreakable [2, pp. 398-400]. The theoretical basis underlying this and related systems was put on a firm foundation a quarter century later by information theory [3]. One time pads require extremely long keys and are therefore prohibitively expensive in most applications.

In contrast, the security of most cryptographic systems resides in the computational difficulty to the cryptanalyst of discovering the plaintext without knowledge of the key. This problem falls within the domains of computational complexity and analysis of algorithms, two recent disciplines which study the difficulty of solving computational problems. Using the results of these theories, it may be possible to extend proofs of security to more useful classes of systems in the foreseeable future. Section VI explores this possibility.

Before proceeding to newer developments, we introduce terminology and define threat environments in the next section.

II. CONVENTIONAL CRYPTOGRAPHY

Cryptography is the study of "mathematical" systems for solving two kinds of security problems: privacy and authentication. A privacy system prevents the extraction of information by unauthorized parties from messages

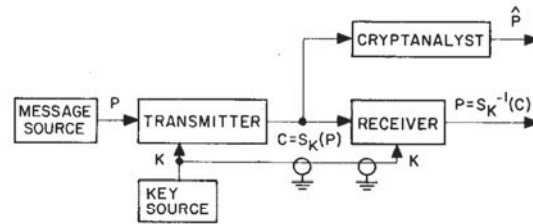


Fig. 1. Flow of information in conventional cryptographic system.

transmitted over a public channel, thus assuring the sender of a message that it is being read only by the intended recipient. An authentication system prevents the unauthorized injection of messages into a public channel, assuring the receiver of a message of the legitimacy of its sender.

A channel is considered public if its security is inadequate for the needs of its users. A channel such as a telephone line may therefore be considered private by some users and public by others. Any channel may be threatened with eavesdropping or injection or both, depending on its use. In telephone communication, the threat of injection is paramount, since the called party cannot determine which phone is calling. Eavesdropping, which requires the use of a wiretap, is technically more difficult and legally hazardous. In radio, by comparison, the situation is reversed. Eavesdropping is passive and involves no legal hazard, while injection exposes the illegitimate transmitter to discovery and prosecution.

Having divided our problems into those of privacy and authentication we will sometimes further subdivide authentication into message authentication, which is the problem defined above, and user authentication, in which the only task of the system is to verify that an individual is who he claims to be. For example, the identity of an individual who presents a credit card must be verified, but there is no message which he wishes to transmit. In spite of this apparent absence of a message in user authentication, the two problems are largely equivalent. In user authentication, there is an implicit message "I AM USER X," while message authentication is just verification of the identity of the party sending the message. Differences in the threat environments and other aspects of these two subproblems, however, sometimes make it convenient to distinguish between them.

Fig. 1 illustrates the flow of information in a conventional cryptographic system used for privacy of communications. There are three parties: a transmitter, a receiver, and an eavesdropper. The transmitter generates a plaintext or unenciphered message P to be communicated over an insecure channel to the legitimate receiver. In order to prevent the eavesdropper from learning P, the transmitter operates on P with an invertible transformation S_K to produce the ciphertext or cryptogram C = S_K(P). The key K is transmitted only to the legitimate receiver via a secure channel, indicated by a shielded path in Fig. 1. Since the legitimate receiver knows K, he can decipher C by operating with S_K^{-1} to obtain S_K^{-1}(C) = S_K^{-1}(S_K(P)) = P, the original plaintext message. The eavesdropper, however, does not know K and cannot decipher C.

be used to transmit P itself for reasons of capacity or delay. For example, the secure channel might be a weekly courier and the insecure channel a telephone line.

A *cryptographic system* is a single parameter family $\{S_K\}_{K \in \{K\}}$ of invertible transformations

$$S_K: \{P\} \rightarrow \{C\} \quad (1)$$

from a space $\{P\}$ of plaintext messages to a space $\{C\}$ of ciphertext messages. The parameter K is called the key and is selected from a finite set $\{K\}$ called the key space. If the message spaces $\{P\}$ and $\{C\}$ are equal, we will denote them both by $\{M\}$. When discussing individual cryptographic transformations S_K , we will sometimes omit mention of the system and merely refer to the transformation K .

The goal in designing the cryptosystem $\{S_K\}$ is to make the enciphering and deciphering operations inexpensive, but to ensure that any successful cryptanalytic operation is too complex to be economical. There are two approaches to this problem. A system which is secure due to the computational cost of cryptanalysis, but which would succumb to an attack with unlimited computation, is called *computationally secure*; while a system which can resist any cryptanalytic attack, no matter how much computation is allowed, is called *unconditionally secure*. Unconditionally secure systems are discussed in [3] and [4] and belong to that portion of information theory, called the Shannon theory, which is concerned with optimal performance obtainable with unlimited computation.

Unconditional security results from the existence of multiple meaningful solutions to a cryptogram. For example, the simple substitution cryptogram XMD resulting from English text can represent the plaintext messages: now, and, the, etc. A computationally secure cryptogram, in contrast, contains sufficient information to uniquely determine the plaintext and the key. Its security resides solely in the cost of computing them.

The only unconditionally secure system in common use is the *one time pad*, in which the plaintext is combined with a randomly chosen key of the same length. While such a system is provably secure, the large amount of key required makes it impractical for most applications. Except as otherwise noted, this paper deals with computationally secure systems since these are more generally applicable. When we talk about the need to develop provably secure cryptosystems we exclude those, such as the one time pad, which are unwieldy to use. Rather, we have in mind systems using only a few hundred bits of key and implementable in either a small amount of digital hardware or a few hundred lines of software.

We will call a task *computationally infeasible* if its cost as measured by either the amount of memory used or the runtime is finite but impossibly large.

Much as error correcting codes are divided into convolutional and block codes, cryptographic systems can be divided into two broad classes: *stream ciphers* and *block ciphers*. Stream ciphers process the plaintext in small

bits of the plaintext. Block ciphers act in a purely combinatorial fashion on large blocks of text, in such a way that a small change in the input block produces a major change in the resulting output. This paper deals primarily with block ciphers, because this *error propagation* property is valuable in many authentication applications.

In an authentication system, cryptography is used to guarantee the authenticity of the message to the receiver. Not only must a meddler be prevented from injecting totally new, authentic looking messages into a channel, but he must be prevented from creating apparently authentic messages by combining, or merely repeating, old messages which he has copied in the past. A cryptographic system intended to guarantee privacy will not, in general, prevent this latter form of mischief.

To guarantee the authenticity of a message, information is added which is a function not only of the message and a secret key, but of the date and time as well; for example, by attaching the date and time to each message and encrypting the entire sequence. This assures that only someone who possesses the key can generate a message which, when decrypted, will contain the proper date and time. Care must be taken, however, to use a system in which small changes in the ciphertext result in large changes in the deciphered plaintext. This intentional error propagation ensures that if the deliberate injection of noise on the channel changes a message such as "erase file 7" into a different message such as "erase file 8," it will also corrupt the authentication information. The message will then be rejected as inauthentic.

The first step in assessing the adequacy of cryptographic systems is to classify the threats to which they are to be subjected. The following threats may occur to cryptographic systems employed for either privacy or authentication.

A *ciphertext only attack* is a cryptanalytic attack in which the cryptanalyst possesses only ciphertext.

A *known plaintext attack* is a cryptanalytic attack in which the cryptanalyst possesses a substantial quantity of corresponding plaintext and ciphertext.

A *chosen plaintext attack* is a cryptanalytic attack in which the cryptanalyst can submit an unlimited number of plaintext messages of his own choosing and examine the resulting cryptograms.

In all cases it is assumed that the opponent knows the general system $\{S_K\}$ in use since this information can be obtained by studying a cryptographic device. While many users of cryptography attempt to keep their equipment secret, many commercial applications require not only that the general system be public but that it be standard.

A ciphertext only attack occurs frequently in practice. The cryptanalyst uses only knowledge of the statistical properties of the language in use (e.g., in English, the letter e occurs 13 percent of the time) and knowledge of certain "probable" words (e.g., a letter probably begins "Dear Sir:"). It is the weakest threat to which a system can be

A system which is secure against a known plaintext attack frees its users from the need to keep their past messages secret, or to paraphrase them prior to declassification. This is an unreasonable burden to place on the system's users, particularly in commercial situations where product announcements or press releases may be sent in encrypted form for later public disclosure. Similar situations in diplomatic correspondence have led to the cracking of many supposedly secure systems. While a known plaintext attack is not always possible, its occurrence is frequent enough that a system which cannot resist it is not considered secure.

A chosen plaintext attack is difficult to achieve in practice, but can be approximated. For example, submitting a proposal to a competitor may result in his enciphering it for transmission to his headquarters. A cipher which is secure against a chosen plaintext attack thus frees its users from concern over whether their opponents can plant messages in their system.

For the purpose of certifying systems as secure, it is appropriate to consider the more formidable cryptanalytic threats as these not only give more realistic models of the working environment of a cryptographic system, but make the assessment of the system's strength easier. Many systems which are difficult to analyze using a ciphertext only attack can be ruled out immediately under known plaintext or chosen plaintext attacks.

As is clear from these definitions, cryptanalysis is a system identification problem. The known plaintext and chosen plaintext attacks correspond to passive and active system identification problems, respectively. Unlike many subjects in which system identification is considered, such as automatic fault diagnosis, the goal in cryptography is to build systems which are difficult, rather than easy, to identify.

The chosen plaintext attack is often called an IFF attack, terminology which descends from its origin in the development of cryptographic "identification friend or foe" systems after World War II. An IFF system enables military radars to distinguish between friendly and enemy planes automatically. The radar sends a time-varying challenge to the airplane which receives the challenge, encrypts it under the appropriate key, and sends it back to the radar. By comparing this response with a correctly encrypted version of the challenge, the radar can recognize a friendly aircraft. While the aircraft are over enemy territory, enemy cryptanalysts can send challenges and examine the encrypted responses in an attempt to determine the authentication key in use, thus mounting a chosen plaintext attack on the system. In practice, this threat is countered by restricting the form of the challenges, which need not be unpredictable, but only nonrepeating.

There are other threats to authentication systems which cannot be treated by conventional cryptography, and which require recourse to the new ideas and techniques introduced in this paper. The *threat of compromise of the receiver's authentication data* is motivated by the situa-

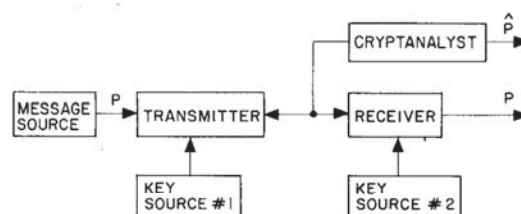
system itself. The receiver's password tables and other authentication data are then more vulnerable to theft than those of the transmitter (an individual user). As shown later, some techniques for protecting against this threat also protect against the *threat of dispute*. That is, a message may be sent but later repudiated by either the transmitter or the receiver. Or, it may be alleged by either party that a message was sent when in fact none was. Unforgeable digital signatures and receipts are needed. For example, a dishonest stockbroker might try to cover up unauthorized buying and selling for personal gain by forging orders from clients, or a client might disclaim an order actually authorized by him but which he later sees will cause a loss. We will introduce concepts which allow the receiver to verify the authenticity of a message, but prevent him from generating apparently authentic messages, thereby protecting against both the threat of compromise of the receiver's authentication data and the threat of dispute.

III. PUBLIC KEY CRYPTOGRAPHY

As shown in Fig. 1, cryptography has been a derivative security measure. Once a secure channel exists along which keys can be transmitted, the security can be extended to other channels of higher bandwidth or smaller delay by encrypting the messages sent on them. The effect has been to limit the use of cryptography to communications among people who have made prior preparation for cryptographic security.

In order to develop large, secure, telecommunications systems, this must be changed. A large number of users n results in an even larger number, $(n^2 - n)/2$ potential pairs who may wish to communicate privately from all others. It is unrealistic to assume either that a pair of users with no prior acquaintance will be able to wait for a key to be sent by some secure physical means, or that keys for all $(n^2 - n)/2$ pairs can be arranged in advance. In another paper [5], the authors have considered a conservative approach requiring no new development in cryptography itself, but this involves diminished security, inconvenience, and restriction of the network to a starlike configuration with respect to initial connection protocol.

We propose that it is possible to develop systems of the type shown in Fig. 2, in which two parties communicating solely over a public channel and using only publicly known techniques can create a secure connection. We examine two approaches to this problem, called public key cryptosys-



tems and public key distribution systems, respectively. The first are more powerful, lending themselves to the solution of the authentication problems treated in the next section, while the second are much closer to realization.

A *public key cryptosystem* is a pair of families $\{E_K\}_{K \in \{K\}}$ and $\{D_K\}_{K \in \{K\}}$ of algorithms representing invertible transformations,

$$E_K: \{M\} \rightarrow \{M\} \quad (2)$$

$$D_K: \{M\} \rightarrow \{M\} \quad (3)$$

on a finite message space $\{M\}$, such that

- 1) for every $K \in \{K\}$, E_K is the inverse of D_K ,
- 2) for every $K \in \{K\}$ and $M \in \{M\}$, the algorithms E_K and D_K are easy to compute,
- 3) for almost every $K \in \{K\}$, each easily computed algorithm equivalent to D_K is computationally infeasible to derive from E_K ,
- 4) for every $K \in \{K\}$, it is feasible to compute inverse pairs E_K and D_K from K .

Because of the third property, a user's enciphering key E_K can be made public without compromising the security of his secret deciphering key D_K . The cryptographic system is therefore split into two parts, a family of enciphering transformations and a family of deciphering transformations in such a way that, given a member of one family, it is infeasible to find the corresponding member of the other.

The fourth property guarantees that there is a feasible way of computing corresponding pairs of inverse transformations when no constraint is placed on what either the enciphering or deciphering transformation is to be. In practice, the cryptoequipment must contain a true random number generator (e.g., a noisy diode) for generating K , together with an algorithm for generating the $E_K - D_K$ pair from its outputs.

Given a system of this kind, the problem of key distribution is vastly simplified. Each user generates a pair of inverse transformations, E and D , at his terminal. The deciphering transformation D must be kept secret, but need never be communicated on any channel. The enciphering key E can be made public by placing it in a public directory along with the user's name and address. Anyone can then encrypt messages and send them to the user, but no one else can decipher messages intended for him. Public key cryptosystems can thus be regarded as *multiple access ciphers*.

It is crucial that the public file of enciphering keys be protected from unauthorized modification. This task is made easier by the public nature of the file. Read protection is unnecessary and, since the file is modified infrequently, elaborate write protection mechanisms can be economically employed.

A suggestive, although unfortunately useless, example of a public key cryptosystem is to encipher the plaintext,

equals $E\mathbf{m}$. Letting $D = E^{-1}$ we have $\mathbf{m} = D\mathbf{c}$. Thus, both enciphering and deciphering require about n^2 operations. Calculation of D from E , however, involves a matrix inversion which is a harder problem. And it is at least conceptually simpler to obtain an arbitrary pair of inverse matrices than it is to invert a given matrix. Start with the identity matrix I and do elementary row and column operations to obtain an arbitrary invertible matrix E . Then starting with I do the inverses of these same elementary operations in reverse order to obtain $D = E^{-1}$. The sequence of elementary operations could be easily determined from a random bit string.

Unfortunately, matrix inversion takes only about n^3 operations. The ratio of "cryptanalytic" time (i.e., computing D from E) to enciphering or deciphering time is thus at most n , and enormous block sizes would be required to obtain ratios of 10^6 or greater. Also, it does not appear that knowledge of the elementary operations used to obtain E from I greatly reduces the time for computing D . And, since there is no round-off error in binary arithmetic, numerical stability is unimportant in the matrix inversion. In spite of its lack of practical utility, this matrix example is still useful for clarifying the relationships necessary in a public key cryptosystem.

A more practical approach to finding a pair of easily computed inverse algorithms E and D ; such that D is hard to infer from E , makes use of the difficulty of analyzing programs in low level languages. Anyone who has tried to determine what operation is accomplished by someone else's machine language program knows that E itself (i.e., what E does) can be hard to infer from an algorithm for E . If the program were to be made purposefully confusing through addition of unneeded variables and statements, then determining an inverse algorithm could be made very difficult. Of course, E must be complicated enough to prevent its identification from input-output pairs.

Essentially what is required is a one-way compiler: one which takes an easily understood program written in a high level language and translates it into an incomprehensible program in some machine language. The compiler is one-way because it must be feasible to do the compilation, but infeasible to reverse the process. Since efficiency in size of program and run time are not crucial in this application, such compilers may be possible if the structure of the machine language can be optimized to assist in the confusion.

Merkle [1] has independently studied the problem of distributing keys over an insecure channel. His approach is different from that of the public key cryptosystems suggested above, and will be termed a *public key distribution system*. The goal is for two users, A and B , to securely exchange a key over an insecure channel. This key is then used by both users in a normal cryptosystem for both enciphering and deciphering. Merkle has a solution whose cryptanalytic cost grows as n^2 where n is the cost to the legitimate users. Unfortunately the cost to the legiti-

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.