Novel Applications of Cryptography in Digital Communications

Jim K. Omura

IN 1976, DIFFIE AND HELLMAN [1] STARTED AN explosion of open research in cryptology when they introduced the notion of public-key cryptography [2]. Today there are many novel applications of public-key cryptography. Most of these are based on digital signatures. This article first presents a tutorial on digital signatures, then discusses several of these applications including those in key management for conventional encryption equipment, electronic mail and data interchange, access control and audit trails, software verification and virus detection, counterfeit-proof currency, nuclear test ban treaty detectors, and challenge response systems such as in aircraft identification of friend or foe.

Digital Signatures

With the rapid expansion of computer applications and digital communication networks, we would expect to have achieved by now the paperless, electronic office that has long been predicted. In fact, today there seems to be more delivery and storage of paper than ever as we continue to rely on filing cabinets to store papers and the mail system, which typically involves *signed* documents *sealed* in an envelope.

In general, hand written signatures are easy to counterfeit and difficult to verify. It is relatively easy with today's technology to take a person's signature, make a photocopy of it, and attach it to a contract that this person did not sign. At the same time, it is difficult to verify a hand written signature without a copy of a known good signature.

To achieve the paperless electronic office we need to replace hand written signatures with digital signatures, and the envelopes with encryption. Such digital signatures would be used to sign purchase orders, applications, time sheets, contracts, and electronic messages of all kinds. The desirable properties of such signatures are:

- Signatures must be difficult to counterfeit.
- Signatures must be easy to verify.

A digital signature consists of a string of symbols. If a person's digital signature were always the same for each message, then one could easily counterfeit it by simply copying the string of symbols. Thus, signatures must be different for each use. This can be achieved by making each digital signature be a function of the message that it is signing, together with a time stamp. To be unique to each signer and counterfeit proof, each digital signature must also depend on some secret number that is unique to the signer. Thus, in general, a counterfeit-proof digital signature must depend on the message and a unique secret number of the signer. The verifier of the signature, however, should not need to know any secret number.

Public-key techniques are the only known means of creating digital signatures with these properties. The two best known forms are due to Rivest, Shamir, and Adleman (RSA) [3] and El Gamal [4]. El Gamal added a signature feature to the basic public-key technique of Diffie and Hellman, the inventors of public-key cryptography. We will now describe the basic concepts of public-key cryptography and digital signatures based on public-key techniques.

Public-Key Cryptography

C. E. Shannon, in his comprehensive 1949 article, "Communication Theory of Secrecy Systems," [5] made the 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 the solution of some problem known to be laborious." The security of all public-key cryptographic techniques are based on well-known, hard-to-solve problems. The Diffie-Hellman [1] key exchange system and the El Gamal [4] signatures are based on the difficult problem of computing discrete logarithms in finite fields, while the RSA [3] public-key scheme is based on the difficulty of finding the prime factors of large integers.

In this article we skip the technical details and present instead a tutorial on the basic ideas in public-key cryptography. Our emphasis is on the various novel applications of this new cryptography.

Any conventional cryptographic algorithm employs one key, which can be used for both locking (encryption) and unlocking (decryption). Knowing this one key allows anyone to encrypt or decrypt a message. In conventional cryptographic systems, the transmitter and intended receiver of encrypted data must share the same cryptographic key. This key must be kept secret and shared only with those users who are allowed access to the encrypted data.

Public-key cryptographic algorithms employ two keys, one for locking (encryption) and another for unlocking (decryption). Although one key is uniquely related to the other in theory, the important point here is that as a practical matter, the two keys of a public-key algorithm are "almost" independent of each other, in the sense that knowing one key does not reveal the other. That is, it is computationally infeasible to find one key when given the other. With public-key algorithms it is possible to encrypt a message without being able to decrypt the

0163-6804/90/0005-0035 $01.00 © 1990 IEEE

encrypted message. Conversely it may be possible to decrypt a message without knowing how the message was encrypted.

Throughout this article we denote public-key encryption and decryption variables with capital letters, using subscripts to denote the owner of the keys. For example, user A may have generated encryption and decryption keys denoted E_A and D_A, respectively. The encryption of "$DATA$" with E_A is shown as

$$E_A(DATA) = ENDATA$$

and the decryption of "$ENDATA$" with D_A is shown as

$$D_A(ENDATA) = DATA.$$

Here we assume that knowing D_A does not reveal E_A and vice versa.

If an encryption key is made public, then any user can encrypt a message using this key, but only the users with the corresponding decryption key can decrypt these messages. Conversely, if a user keeps an encryption key secret and makes public the corresponding decryption key, then anyone with the decryption key can not only decrypt messages encrypted by this user, but they will also know that the message has not been altered. Only the user with the encryption key could have originated the encrypted message. This property of unique origination permits this user with the secret encryption key to send "signed" messages.

Creating Digital Signatures

Suppose a person, say Alice, creates her own encryption key E_A, and the corresponding decryption key D_A. Assume that Alice will keep her encryption key E_A secret and make publicly known her decryption key, D_A. Thus, we assume the following:

- E_A is secret.
- D_A is made public.

Alice's non-secret decryption key may be placed in a trusted public directory where all public numbers are available to anyone. If Alice wants to send a signed message to Bob, she first computes a known hash function [6] [7] of the message, which is sometimes called a Manipulation Detection Code (MDC) [8]. The message M may be arbitrarily while the hash function of the message, denoted H, is fixed in length, typically about 512-b long. Any good hash function has the property that it is virtually impossible to find two messages that will result in the same computed hash function. The hash function is not secret and is assumed to be known to everyone.

Alice's signature for the message M is the encrypted hash function given by

$$S = E_A(H).$$

As illustrated in Figure 1a, Alice creates her digital signature by first computing a hash function H of the message M and using her secret encryption key E_A to encrypt it to create her digital signature S.

Verifying Digital Signatures

When Bob receives Alice's signed message, he can determine the authenticity of Alice's signature and verify the integrity of Alice's message. This is illustrated in Figure 1b, where Bob receives the message M' together with the signature S. Bob first computes the hash function of the message that he has received, H'. He then accesses the trusted public directory and obtains Alice's non-secret decryption key D_A. With this he obtains the hash function that was computed by Alice,

$$H = D_A(S)$$

and compares this with the hash function he computed for the

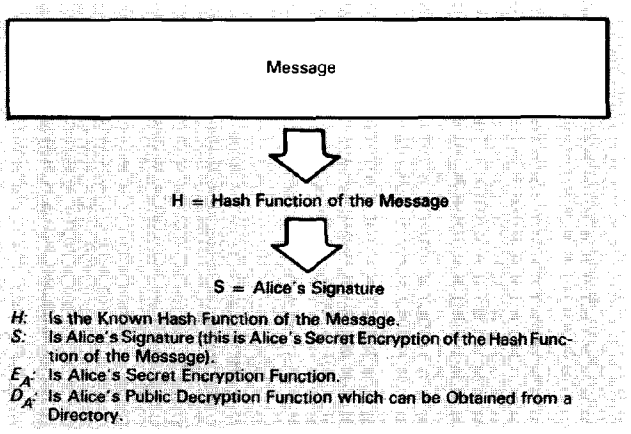

Fig. 1a. Creating Alice's signature.

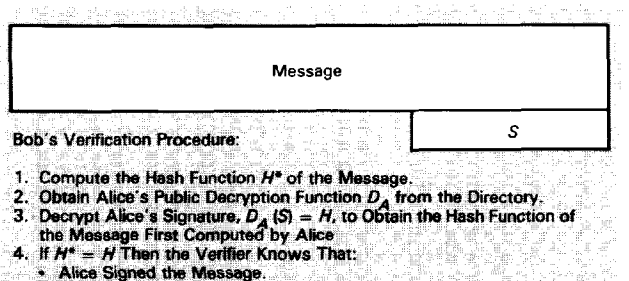

Fig. 1b. Verification with a trusted directory.

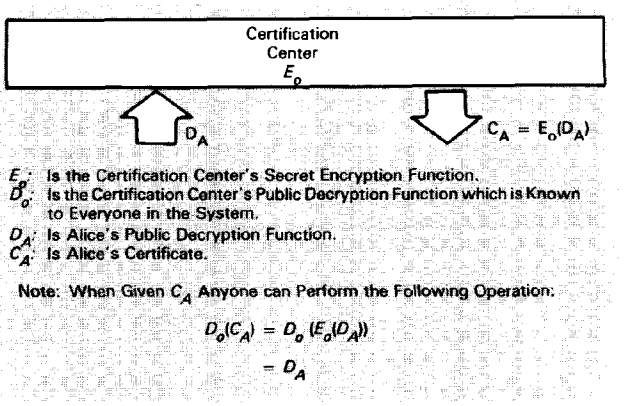

Fig. 1c. Creating Alice's certificate.

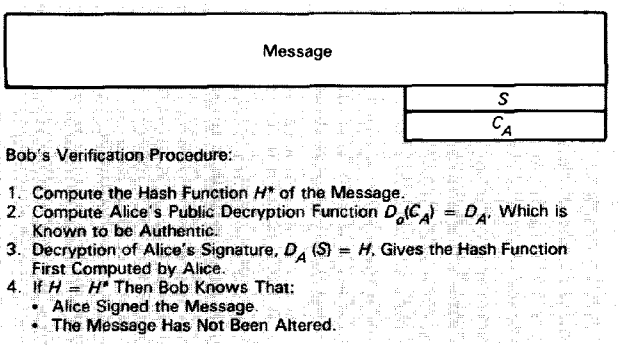

Fig. 1d. Verification with a certificate.

received message, H'. If the two hash functions agree, then he knows the following:

- Alice signed the message.
- The message has not been altered since Alice signed it.

Thus, without knowing any secrets of Alice, Bob can verify the authenticity of Alice's signature and the integrity of her message. Moreover, he cannot counterfeit Alice's signature.

Certification Center

The use of a trusted public directory may be replaced by a certification center that acts like a notary service that certifies all public decryption keys. Suppose a trusted certification center has it own secret encryption key E_O, and the corresponding non-secret decryption key D_O. We assume that everyone in the system has knowledge of the certification center's public decryption key D_O. D_O could be available through the newspapers, the telephone directory, or embedded in any equipment that does signature verification.

Alice obtains certification of her public decryption key D_A as shown in Figure 1c. First she identifies herself to the certification center. This is assumed to be done without fraud. Next, the certification center encrypts Alice's public decryption key to create Alice's certificate,

$$C_A = E_O(D_A)$$

which is then given to Alice.

Since everyone has the decryption key D_O of the certification center, anyone can obtain Alice's public decryption key from her certificate C_A by

$$D_A = D_O(C_A).$$

Furthermore, that person would know for certain that this is Alice's public decryption key because only the certification center could have created this certificate.

Although not shown here, it is assumed that Alice's name is also included in her certificate so that the decrypted public number is identified with Alice.

Once Alice has a certificate, she can attach it to her signed messages. This allows Bob (or anyone else) to verify Alice's signature without going to any public directory. The certificate provides Alice's public decryption function in a counterfeit-proof form. With this system Alice's signed message will then consist of the message M, the signature S, and Alice's certificate C_A. Bob's verification procedure is illustrated in Figure 1d, which is the same as in Figure 1b, except that Alice's public decryption function D_A is obtained by Bob directly from the certificate C_A provided by Alice in the signed message.

Applications

We describe here several applications of these digital signatures. Some of these applications have already begun but most are still in the planning and testing stages.

Throughout the discussion of various applications of public-key cryptography we assume that all public numbers are certified or signed by some trusted authority. The International Consultative Committee for Telephone and Telegraph (CCITT) has adopted standard X.509, titled, "The Directory—Authentication Framework," which defines a hierarchy of trusted certification authorities [9]. Alcatel STK currently has a system based on this authentication framework [10]. For the remainder of this article we assume that a certification center is properly used; thus, public numbers cannot be counterfeited, and any public number can always be correctly identified.

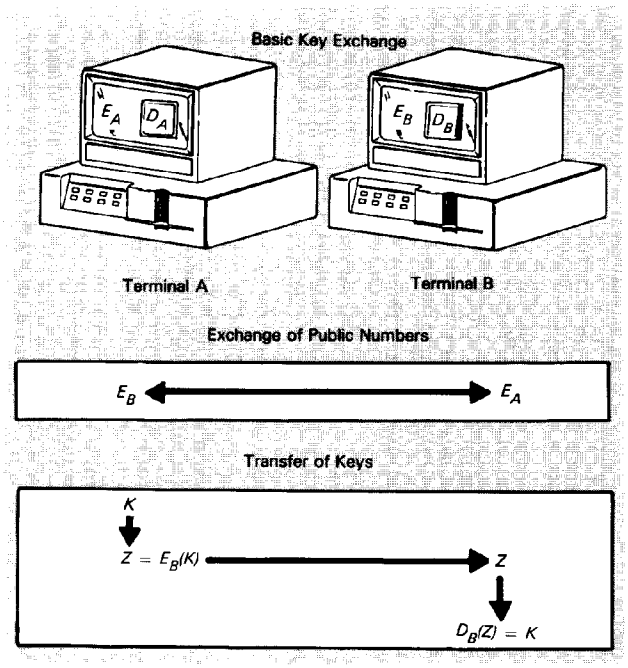

Fig. 2. Alice sending a key to Bob.

Key Management

Conventional encryption algorithms such as the Data Encryption Standard (DES) are commonly used to protect the privacy of information that is stored or transmitted. This is the electronic analogy to using an envelope to protect the contents of a letter in the mail system. In order for conventional encryption devices to work together, however, they must share the same secret cryptographic key. The distribution of secret cryptographic keys is the key management problem [11].

Until recently cryptographic keys were always distributed by couriers. In these systems the primary weakness is having people manually deliver secret keys. It is well-known that the best attack against conventional courier-based key management systems is the "key purchase" attack. In addition, the operational costs of courier-based key management typically dominate the life-cycle cost of using data encryption equipment.

Because public-key functions are computationally intensive, they are usually too slow for most continuous encryption of data. They are used primarily for key management with conventional encryption equipment. Public-key techniques are now used to automate key management in a way which is low cost and secure. Several standards groups and the National Institute of Standards and Technology (NIST) are currently considering these key management techniques [12–16].

Alice and Bob, for example, can now perform a key exchange using the following public-key technique in which the decryption functions are kept secret and the encryption functions are public:

- *Step 1*: Over the insecure communication link between them, Alice sends to Bob her non-secret encryption function E_A.
- *Step 2*: Bob randomly generates a number R and encrypts it using Alice's encryption function to obtain the encrypted message

$$V = E_A(R)$$

which he sends to Alice.

• *Step 3*: Alice uses her secret decryption function D_A to decrypt Bob's message to obtain his secret random number R by

$$R = D_A(V).$$

Alice and Bob now have a shared secret number R which can be used to form a common encryption key. In this case, however, Bob alone completely defines the common encryption key. This is generally not desirable.

Steps 1-3 above could be repeated again with Bob sending Alice his non-secret encryption function and Alice randomly generating a secret number which she sends to Bob after encrypting it with Bob's encryption function. Figure 2 illustrates this where Alice sends Bob the secret key K. In this manner, Alice and Bob each receive a random number generated by the other person, and they can then combine the two random numbers to form a common encryption key that is not completely specified by only one of them.

With the exchange of only non-secret numbers, any two encryption devices in a network can securely derive a shared secret encryption key which allows them to establish a secure link between them. The most widely used public-key method for key management today is the Diffie-Hellman key exchange system, referred to as Secure Electronic Exchange of Keys (SEEK) by CYLINK [11].

Strictly speaking, this Diffie-Hellman technique does not fit exactly the definition of an asymmetric key encryption technique, yet it has the same property for key exchange of allowing two people to share a secret key by exchanging only non-secret numbers over an insecure communication link. It is a public-key scheme in that there are secret numbers and corresponding public numbers involved.

In the SEEK system, all encryptors in the network have the same 512-b, non-secret constants:

- p is a large prime number which defines a field, denoted $GF(p)$, consisting of all non-negative integers less than p.
- a is an integer in the field $GF(p)$.

The cryptographic security of this scheme rests on the difficulty of computing discrete logarithms in $GF(p)$ when p is a large prime number and p-1 has a large prime factor. Specifically, given integer X, consider the exponentiation function

$$Y = a^X \bmod p.$$

Given X, it is relatively easy to compute Y. On the other hand, given Y, it is very difficult to compute the exponent X. This is the problem of finding the discrete logarithm of Y. Based on what is known about this problem, 512-b prime numbers are recommended.

Alice and Bob can now obtain a shared secret encryption key as illustrated in Figure 3 and described in the following steps:

- *Step 1*: Alice randomly generates a secret integer denoted X_A and computes the corresponding public number

$$Y_A = a^{X_A} \bmod p.$$

Bob similarly randomly generates his secret number X_B and computes the corresponding public number

$$Y_B = a^{X_B} \bmod p.$$

- *Step 2*: Over the insecure communication link between them Alice and Bob exchange their public numbers Y_A and Y_B.

Note that knowledge of the public numbers Y_A and Y_B does not reveal the secret numbers X_A and X_B since we assume that computing discrete logarithms is computationally infeasible.

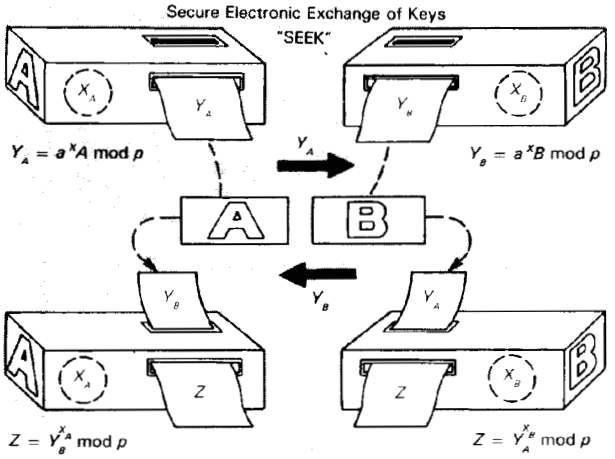

Fig. 3. The Diffie-Hellman key exchange.

• *Step 3*: Alice combines Bob's public number Y_B and her secret number X_A to compute a secret number Z by the formula

$$Z = Y_B^{X_A}.$$

Bob similarly combines Alice's public number Y_A and his secret number X_B to compute the same secret number Z by the formula

$$Z = Y_A^{X_B}.$$

Although Alice and Bob each use different numbers in their computations, they end up computing the same shared secret number Z. This is because multiplication of terms in the exponent of a are commutative, as shown in the relationship

$$\begin{aligned} Y_B^{X_A} &= a^{X_B X_A} \\ &= a^{X_A X_B} \\ &= Y_A^{X_B}. \end{aligned}$$

The Diffie-Hellman scheme thus allows Alice and Bob to compute a shared secret number Z by exchanging only non-secret numbers Y_A and Y_B. Parts of this shared secret number Z can be used as a common encryption key to encrypt the communication link between them.

Another important feature of the Diffie-Hellman scheme for key exchange is the fact that the shared secret number obtained depends on random secret numbers that both Alice and Bob independently generated. Thus Alice, for example, cannot control by herself the common encryption key.

Electronic Mail and Data Interchange

INTERNET has adopted public-key techniques for both key management for conventional encryption devices and for creating digital signatures for messages [13] [14]. These electronic signatures are also being considered for Electronic Data Interchange (EDI), where contracts and purchase orders can be signed and delivered electronically. The *non-repudiation* property of these digital signatures is what can make EDI work securely and force the timely controls needed in "just-in-time" manufacturing. The British banks have adopted a similar system for Electronic Funds Transfer (EFT) for Point Of Sales (POS) systems [15]. For Local Area Networks (LANs) the IEEE 802.10 LAN Security Working Group is currently drafting a security standard using public-key techniques for key management [16].

Access Control

Identification of individuals is one of the basic requirements of access control, whether it is for access into buildings, into authorization of credit at a point of sale, or into computers and communication networks. Here we address how public-key techniques are now being applied to the identification of individuals [17].

Typically, authentication of an individual can be based on what a person *has*, what a person *knows*, and what a person *is* as measured by biometrics.

For access into computers, the most common form of identification relies on a password which is kept secret and known only by the individual and is also stored in the computer. Because passwords are easily compromised, some computer access control systems for dial in access use a "call back" system in addition to passwords. Others use dynamic passwords which are generated by some algorithm, computed in a small battery-powered device (sometimes called a "token"). Some of these tokens double as calculators and can only be activated when the correct Personal Identification Number (PIN) is entered into the token [18].

The small personal tokens described above do not require any special hardware at the terminal where the individual is entering the computer network. Some of these tokens contain a unique secret algorithm (an encryption algorithm with a secret key is commonly used) which is also known to the computer being accessed. At the time of access, the computer can send a challenge number C displayed on the access terminal screen and ask the individual to compute with his or her token the proper response R, which one then types into the terminal. This system identifies a person based on something one has (the token) and something one knows (the PIN to activate the token).

In most access control systems, personal data of each user is stored in a secure computer. This personal data must be retrieved each time someone, usually using a password or token, wants to access the computer network. To avoid the extra communication to the computer storing this database, off-line systems require the user to provide this data using a magnetic stripe card, a data key, a floppy disk, or a "smart card" [19]. Certainly the operational procedures would be simpler if each user presented his or her own personal data to the terminal.

Devices that contain some personal data that can be used for identification usually require some extra hardware at the access terminal to connect to the personal device. Magnetic stripe cards, for example, require a magnetic stripe card reader. These generally do not contain much data. Smart cards are now replacing magnetic stripe cards in many applications. Smart cards are standard size plastic cards with an embedded Integrated Circuit (IC). These card ICs not only contain memory cells for holding considerable amount of personal data but they also have Central Processing Unit (CPU) cells capable of performing computations.

In France today, 60,000 smart cards a day are being issued. These contain a single IC chip with a CPU, Read-Only Memory (ROM), Random Access Memory (RAM), and Electrically Programmable Read-Only Memory (EPROM) or Electrically Erasable/Programmable Read-Only Memory (EEPROM). Such smart cards will become common throughout the world and replace our current credit cards and ID badges. The primary advantage of smart cards is their convenient small size and ability to securely hold lots of personal data where some of it can be updated or changed. With EEPROM cells, transaction data can be stored in these cards. Several European institutions are now adopting these smart cards for financial transactions including their use in telephones with card readers.

For the purpose of identification, we assume that what a person *has* is a smart card with his personal data stored on it. What a person *knows* may be a password which may be part of the stored data. Biometrics which measure what a person *is*

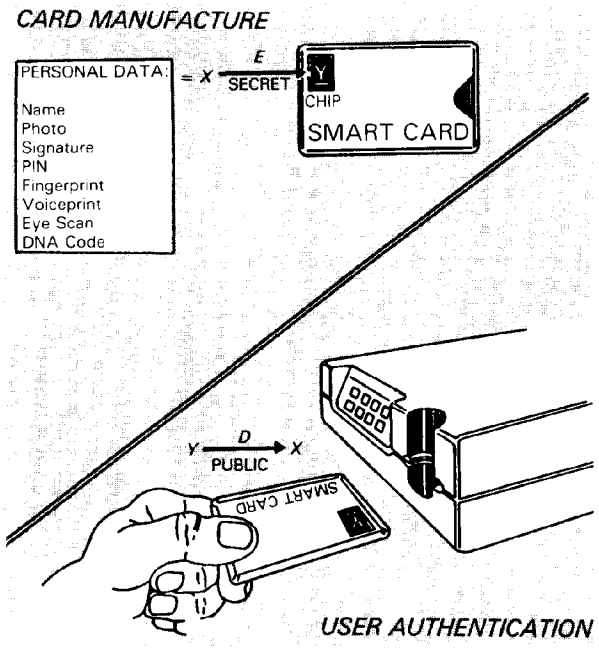

Fig. 4. Authentication of smart card data.

may be one's voice print, one's fingerprint, one's eye scan, one's picture, one's handwritten signature, or eventually one's DNA code. Biometrics data can also be stored in the person's smart card. Today the primary problem with biometrics is the cost of its measuring equipment. However for a high level of security, biometrics measurements are sometimes required at each point of access.

The personal smart cards often hold the data necessary for anyone to authenticate the user's identification. This means that the authenticating terminal or host computer does not need to maintain a database of all users' data. In this type of off-line system, this smart card data is crucial to the identification of a person accessing a host computer from a terminal. It is, therefore, important for the terminal or host computer to first authenticate this smart card data.

Public-key provides a means of authenticating this crucial data in off-line systems by providing digital signatures that any terminal can verify. A certification center creates a digital signature for the data on each smart card, and this signature is also included in the smart card data. Any alteration of the personal identification data in the smart card will result in an incorrect certification center digital signature, which can be immediately detected.

A terminal can check the signature in the smart card using a non-secret number of the certification center. Because this non-secret number is the same for verifying all signatures made by the certification center, it can be embedded into each terminal at the factory. With the certification center's non-secret number each terminal can verify that the data in each smart card is authentic, as illustrated in Figure 4. Thus, each user carries in one's own smart card the necessary data to identify oneself. The integrity of this data can be established by the attached certification center's digital signature. The first systems of this kind were used by Sandia National Laboratories in 1979 [20] [21].

This type of digital signature is also used in the French smart cards where the certification center is the issuer of the smart cards [22]. As the cards are issued and a signature is added to the data in each smart card, the certification center has the responsibility of correctly identifying the person receiving the smart card. This one time identification of an au-

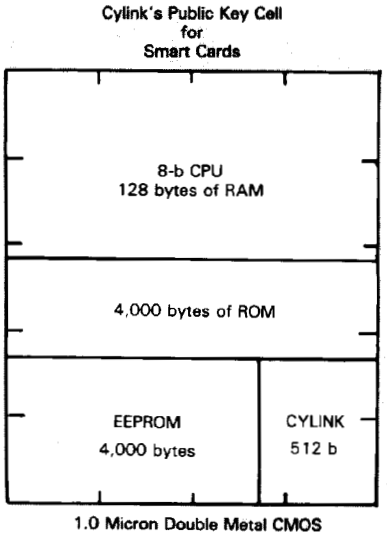

Fig. 5. Floor plan of Cylink's smart card chip.

thorized user and the issuance of a smart card are the most critical steps in this system.

Besides having terminals that have smart card readers, more secure access control systems also use biometrics devices. Biometrics devices that are available today include those that measure fingerprints, eye retina, hand signature dynamics, voice characteristics, and hand geometry [23] [24]. Some of these also include digitized photographs, and someday may measure the DNA code of an individual. Here, each person's biometrics data is stored in one's smart card. Once a terminal authenticates this data, it can be used to further check the person's identification using a biometrics device.

Audit Trails

Computer crimes are primarily due to authorized users of a system. Computer network users are initially tempted to defraud a system when they observe that certain entry errors are undetected. One way to inhibit such fraud is to trace errors to individual users. With a smart card, each access time and terminal location can be recorded onto the smart card by the terminal. Each user can then be required to periodically log in their smart card audit trail data to a host computer.

In addition to aiding in the identification process, smart cards will eventually be used to create personal digital signatures. To do this each smart card must be able to compute the required digital signature and provide the non-secret number that will allow anyone to verify its signatures. This non-secret number used to verify the smart card signatures must in turn be signed by the certification center so that anyone can authenticate it.

Generally the existing CPU on smart cards cannot do the computationally intensive public-key calculations required in any reasonable time. To facilitate these types of calculations, special mathematical cells can be integrated into the smart card chips. Smart cards containing these cells are now being designed [25–27].

CYLINK has been a leader in the development of public-key chips with a patented design that achieves the smallest area cells for computing modulo addition, multiplication, and exponentiation. These mathematical operations are necessary for creating digital signatures. Until now it was generally believed that a public key cell small enough for smart cards was not possible.

CYLINK has modified its 512-b chip design to reduce the chip area further at a sacrifice of some computation speed. Using a 1.0 μ double metal Complementary Metal Oxide Semiconductor (CMOS) process, this new design for a public-key chip occupies an area of 2 sq mm. This is less than 10% of the 22 sq mm area of current International Organization for Standardization (ISO) smart cards. With a 10-MHz clock, this chip will do a 512-b modulo exponentiation with an average time of 2 s. Figure 5 shows a floor plan of a 5 mm by 4 mm area smart card chip with an 8-b CPU including 128 bytes of RAM, 4 Kbytes of ROM, 4 Kbytes of EEPROM, and the CYLINK public-key processor cell.

Secure identification and personal digital signatures using smart cards will be the foundation of computer access control systems of the future. With this capability, most paper work can be eliminated and we can achieve a secure electronic information society. All transactions can be signed by the individual's smart card so that a distributed audit trail is left behind for anyone to track. All electronic messages can also be signed by the personal smart card. Digital Equipment Corporation has recently described a distributed system security architecture that includes many of these ideas based on smart cards and public-key techniques [28].

With the use of a smart card capable of public-key functions, each individual can carry a trusted device that will allow anyone to identify him or her without any database. Each person's transactions will be signed with non-repudiation signatures. This will then allow for secure means of carrying out transactions on public telephones and home terminals. For example, any telephone with a smart card reader, such as those in France, can be used as an Automatic Teller Machine (ATM) where a person can dial one's bank and have money transferred electronically to one's own smart card. This card can then serve as an electronic wallet from which merchants will debit the sales price from the smart card using smart card reader terminals. All of these transactions would be signed by both parties in each transaction using these non-repudiation signatures. A home terminal with a smart card reader can also be used to provide all kinds of home services, such as with the Minitel in France, the Prodigy system (IBM and Sears) in the United States, and Nintendo's home service system in Japan. Perhaps someday such a system will be used in a secure electronic voting system [29].

Another application of smart cards with public-key functions include digital radios with smart card readers. With such radio devices in automobiles, highway toll fees and parking fees can be electronically paid in the same way that electronic transactions are handled with point of sale terminals. With digital radios these automobile transactions can be done fast

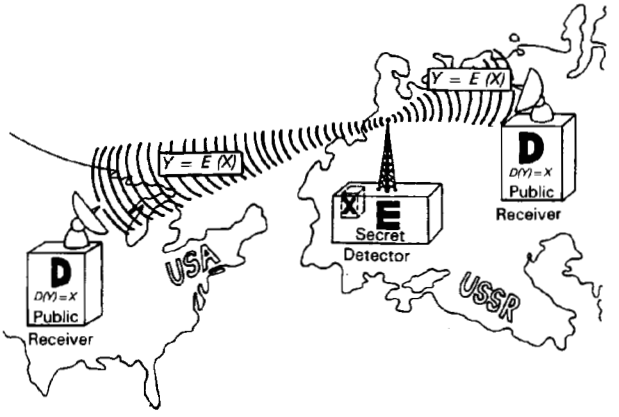

Fig. 6. Test ban treaty verification.

enough that traffic congestion is eliminated at the toll booths. The new personal pocket telephones such as CT-2 in England can also be used with these access control and audit trail schemes based on public-key techniques.

Software Verification and Virus Detection

A software package can be thought of as a message which can be signed by the producer of the software. With public-key signatures anyone can verify the authenticity of the signature and the integrity of the software package. A computer can then periodically check the integrity of software it runs regularly as well as the integrity of each new program it is asked to run. Although this does not protect against copies, this use of public-key signatures can be used to verify that the software has not been altered since the manufacturer signed it. Game machines, for example, can be designed to run only authorized game programs with the appropriate signatures.

This software verification scheme assumes that there is a trusted, protected module that does the verification of signatures, and that this unit cannot be bypassed. With the use of personal smart cards with the public-key signature function and a computer with a smart card reader, any person can essentially sign any program or database that he creates. For example, Alice (with her smart card) might sign a database that she then stores in the hard disk of her personal computer. Later, when she retrieves this database, she can use her card to verify the signature. If the signature is verified then she knows that the data has not been altered since she last signed it. Indeed,

With the development of smart cards capable of creating digital signatures, the personal, all-purpose identification card will move us closer to the secure paperless electronic society.

anyone else can also verify the signature and know that the database has not been altered since Alice signed it [30]. This is, of course, one means of detecting a "computer virus."

Nuclear Test Ban Detector

Figure 6 illustrates a novel idea described by Simmons [31] [32] where public-key signatures are used to verify test ban treaty compliance. If the United States and the Soviet Union were to have a complete ban on nuclear testing, then the two countries might install seismometers in various locations in the other's country. These detectors would detect any ground movement due to nuclear explosions. The host country might suspect that the other country's seismometers are used to send spying data as well as seismographic data. It would, therefore, be important that the data from these devices be available to both countries. At the same time, each country needs to be assured that the data it receives from the seismometers that they installed is authentic and not fake data that might be substituted by the host country.

The proposed seismometer contains a tamper proof module that is capable of creating its own unique digital signature, which is used to sign each block of seismic data that is recorded. This data is then transmitted to both the Soviet Union and the United States, and perhaps to a United Nations site as well. The United States can authenticate the signature and verify the integrity of the data while the Soviets can do the same. The Soviets, however, would not be able to counterfeit the seismometer data of devices installed in their soil by the United States

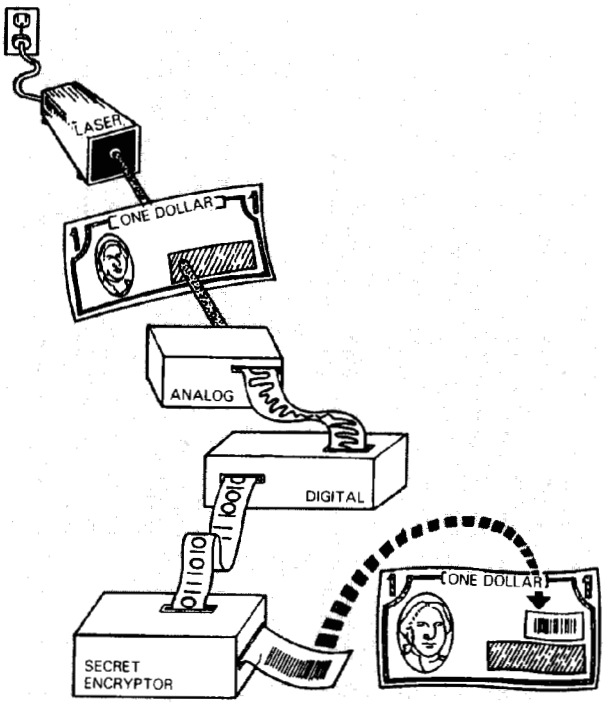

Fig. 7. Counterfeit-proof bank notes.

since the encryption function is unknown to them. This seems to satisfy the requirements of both countries. Similar Soviet seismometers might be placed in key locations in the United States.

Counterfeit-Proof Objects

Besides creating counterfeit-proof signatures, the basic public-key signatures can be used to make counterfeit-proof objects [32]. Light Signatures [33] has proposed a counterfeit-proof stock certificate system that is illustrated in Figure 7. The basic idea is to measure some unique "fingerprint" of the paper and to sign (encrypt) it using the secret key of the manufacturer of the stock certificate. For currency, this would be the secret encryption key of the United States Bureau of Printing. The fingerprint is obtained by moving a narrow intense light beam along a line on the paper and measuring the light intensity that passes through the paper. The light intensity function determined by the unique random pattern of paper fibers along the line then forms the fingerprint of the particular piece of paper. This fingerprint is then digitized and encrypted by the secret encryption function. The encrypted fingerprint is then printed onto the paper in digital form such as a bar code.

Suppose at some later date the authenticity of the stock certificate is to be verified at some bank. The Light Signatures verification system located at the bank only needs the non-secret public decryption function to decrypt the encrypted data on the paper and reconstruct the intensity function that was signed. Next, the actual intensity function of the stock certificate is measured. If this newly measured intensity function agrees with the intensity function reconstructed from the decrypted data according to some mean square error threshold, the document is declared authentic.

The same basic idea can be applied to all kinds of objects to detect counterfeits. In the automobile industry there has been a problem of dishonest distributors supplying auto makers with fake, low-quality parts [34]. Again the idea for detection of

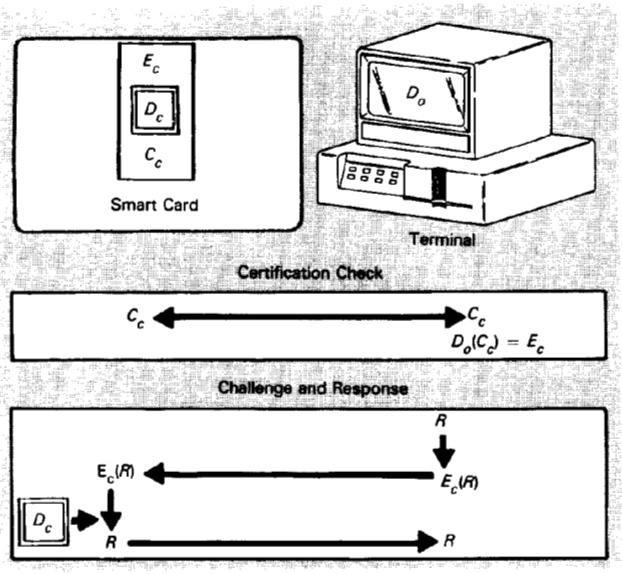

Fig. 8. Challenge response verification of a smart card.

counterfeit parts depends on the measurement or creation of a fingerprint that is unique to each object. For example, a manufacturer can place a fingerprint on each brake it manufactures in the form of special paint that contains material which forms random patterns that can be measured and digitized. The digitized fingerprint can then be encrypted using the manufacturer's secret encryption function. This encrypted data is then painted in bar code onto the brake along with the fingerprint. When this brake gets to the auto maker, say Chrysler, its authenticity can be checked using only the non-secret decryption function of the brake manufacturer. The verification process involves decrypting the encrypted fingerprint, reconstructing the fingerprint from this data, then comparing this with the measured fingerprint that is also on the brake. If these agree to some correlation threshold, the brake is declared authentic.

The key to making counterfeit-proof objects is to create a unique random pattern for each object that cannot be removed [35]. This idea is essentially the same as detecting counterfeit individuals where some biometrics is used such as the person's actual fingerprint. Rather than sticking the encrypted version of the digitized fingerprint on the person, we recommend that each person carry a smart card containing in its memory this fingerprint which is signed (encrypted) by the issuer of the smart card. Since the encryption function is kept secret by the manufacturer, the encrypted fingerprint data cannot be created by anyone else. The verification process, however, only requires the non-secret decryption key of the smart card issuer. Thus verification is easy, while making a counterfeit is difficult.

Challenge Response

Aircraft radar systems often need to identify aircraft when they appear on the radar screen. With public-key signatures each aircraft might have the capability to sign any signal that it receives and to transmit this back. If an aircraft is equipped with a module that can perform this public-key signature function, then the radar operator can randomly generate a challenge signal R and send this by radio signals to the aircraft. The aircraft can then sign this challenge and radio the signature back to the radar. The radar operator can then verify the signature and authenticate the identity of the aircraft. Only the aircraft can create its own unique signatures while anyone can verify its authenticity.

Such challenge response systems are now being proposed for access control with smart cards [19] [21–23] [28] to authenticate users' smart cards. Figure 8 illustrates how this is done. As in the aircraft challenge response scheme, here the terminal issues a challenge to the smart card, which in turn responds with a signature on the challenge. This signature can be verified by the terminal, which can then be assured that the smart card is authentic. As stated earlier, the user typically has to enter one's PIN to activate one's smart card. In addition, there may be a biometrics device that measures his or her fingerprint, and this data also enters the card and must be verified by the card before it is activated.

Conclusion

The first applications of public-key cryptography at Sandia [20] [21] came shortly after the original 1976 public-key article by Diffie and Hellman. Widespread applications, however, are just now taking place and several national and international standards for the use of public-key techniques for key management and digital signatures are being proposed. With the development of smart cards capable of creating digital signatures, the personal, all-purpose identification card will move us closer to the secure paperless electronic society.

References

[1] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Trans. on Info. Theory*, vol. IT-22, pp. 644–654, Nov. 1976.

[2] J. L. Massey, "An Introduction to Contemporary Cryptology," W. Diffie, "The First Ten Years of Public-Key Cryptography," and G. J. Simmons, "A Survey of Information Authentication," *Proc. of the IEEE*, vol. 76, May 1988.

[3] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Commun. of the Assoc. of Comp. Mach.*, vol. 21, pp. 120–126, Feb. 1978.

[4] T. El Gamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Trans. on Info. Theory*, vol. IT-31, pp. 469–472, July 1985.

[5] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell Sys. Tech. J.*, vol. 28, pp. 656–715, Oct. 1949.

[6] I. B. Damgard, "Design Principles for Hash Functions," *Proc. of CRYPTO '89*, pp. 395–406, Aug. 1989.

[7] R. C. Merkle, "One Way Hash Functions and DES," *Proc. of CRYPTO '89*, pp. 407–420, Aug. 1989.

[8] R. R. Jueneman, "Electronic Document Authentication," *IEEE Network Mag.*, vol. 1, pp. 17–23, Apr. 1987.

[9] "The Directory," ISO 9594/1-8 (CCITT X.500 series).

[10] C. Fritzner, K. Presttun, J.-T. Richardsen, and G. Soberg, "Experimental Secure Distributed Information System," *Alcatel STK Tech. Rep.: Elect. Commun.*, vol. 62, no. 3/4, Oslo, Norway, 1988.

[11] D. B. Newman, Jr., J. K. Omura, and R. L. Pickholtz, "Public-Key Management for Network Security," *IEEE Network Mag.*, vol. 1, no. 2, pp. 11–16, Apr. 1987.

[12] J. Graff, "Practical Key Management Implementations of Public-Key Techniques," *Proc. of ISSA '90*, St. Louis, Mo., Mar. 19-20, 1990.

[13] "Internet to Use Privacy Code," *San Jose Mercury News*, Mar. 26, 1989.

[14] J. Linn and S. T. Kent, "Privacy for DARPA-INTERNET Mail," *Proc. of the 12th Nat'l Comp. Sec. Conf.*, pp. 215–229, Oct. 1989.

[15] "National EFT POS to Use Public Key Cryptography," *Info. Sec. Mon.*, vol. 2, no. 12, p. 1, Nov. 1987.

[16] L. K. Baker and K. Kirkpatrick, "The SILS Model for LAN Security," *Proc. of the 12th Nat'l Comp. Sec. Conf.*, Baltimore, pp. 267–276, Oct. 1989.

[17] J. K. Omura, "A Computer Dial Access System Based on Public-Key Techniques," *IEEE Commun. Mag.*, vol. 25, pp. 73–79, July 1987.

[18] M. Smid, J. Dray, and R. B. J. Warnar, "A Token-Based Access Control System for Computer Networks," *Proc. of the 12th Nat'l Comp. Sec. Conf.*, Baltimore, pp. 232–253, Oct. 1989.

[19] M. E. Haykin and R. B. J. Warnar, "Smart Card Technology: New Methods for Computer Access Control," *NIST Spec. Publ. 500-157, Comp. Sci. and Tech.*, Sept. 1988.

[20] P. D. Merillat, "Secure Stand-Alone Positive Personnel Identity Verification System (SSA-PPIV)," *Sandia Nat'l Lab. Tech. Report SAND79-0070*, Mar. 1979.

[21] G. J. Simmons, "A System for Verifying the Identity and Authorization at the Point-Of-Sale or Access," *Cryptologia*, vol. 8, no. 1, pp. 1–21, Jan. 1984.
[22] L. C. Guillou, "Smart Cards and Conditional Access," *Proc. of EUROCRYPT '84*, Paris, pp. 480–489, Apr. 1984. Also in *Advances in Cryptology*, Germany: Springer Verlag, 1985.
[23] T. C. Feustel and G. Velius, "Access Security: Using Voice and Smart Cards to Protect Assets," *Dig. of Tech. Info.*, Bellcore publication SR-TSY-000104, vol. 6, no. 9, Dec. 1989.
[24] For up to the date information see the newsletter, *Pers. Ident. News*, published by Warfel & Miller (ISSN 0883-5608).
[25] J. K. Omura, "A Smart Card to Create Electronic Signatures," *Proc. of ICC '89*, session 37, Boston, June 1989.
[26] J. K. Omura, "A New Smart Card Chip for Creating Signatures," *Proc. of SCAT '89*, Washington, D.C., pp. 37–47, May 1989.
[27] J.-J. Quisquater, "A New Step in Smart Card," *CRYPTO '89*, UC Santa Barbara, Aug. 1989. Also see the flyer by Philips Components, Aug. 10, 1989.
[28] M. Gasser, A. Goldstein, C. Kaufman, and B. Lampson, "The Digital Distributed System Security Architecture," *Proc. of the 12th Nat'l Comp. Sec. Conf.*, pp. 305–319, Oct. 1989.
[29] J. D. Cohen and M. J. Fischer, "A Robust and Verifiable Cryptographically Secure Election Scheme," *FOCS '85*, pp. 372–382.
[30] M. M. Pozzo and T. E. Gray, "An Approach to Containing Computer Viruses," *Comp. and Sec.*, vol. 6, pp. 321–331, 1987.
[31] G. J. Simmons, "How to Insure that Data Acquired to Verify Treaty Compliance are Trustworthy," *Proc of the IEEE*, vol. 76, no. 5, pp. 621–627, May 1988.
[32] C. L. Henderson and A. M. Fine, "Motion, Intrusion and Tamper Detection for Surveillance and Containment," *Sandia Nat'l Lab Rep. SAND79-0792*, Mar. 1980; also published by the International Safeguards Project Office for the International Atomic Energy Agency (IAEA) as ISPO Rep. 91, 1980.
[33] "Counterfeit-Proof Stock Certificates," private communication with Light Signatures, a subsidiary of Telecredit, Inc., 1986. Also see Simmons [2].
[34] "Iacocca Counters Counterfeiters," *Info. Week*, p. 50, Mar. 23, 1987.
[35] G. J. Simmons, "A Survey of information Authentication," *Proc. of the IEEE*, vol. 76, no. 5, pp. 603–620, May 1988.

Biography

Jim K. Omura is Professor of Electrical Engineering at UCLA and Chairman of CYLINK Corporation. He received his B.S. and M.S. degrees from MIT and his Ph.D. from Stanford in 1966. He joined the faculty of UCLA in 1969. His research interests are in coding theory, analysis and design of communication systems, data compression and rate distortion theory, digital radio techniques, spread spectrum systems, satellite communication systems, and cryptography. Dr. Omura now works full-time at CYLINK, which he co-founded. He is a Fellow of IEEE, and co-authored (with A. J. Viterbi) *Principles of Digital Communication and Coding*, and (with Simon, Scholtz, and Levitt) *Spread Spectrum Communications, Vols. I, II, III.*