Security in Electronic Messaging Systems

Diana M. D'Angelo
Bruce McNair
Joseph E. Wilkes

Electronic messaging systems are quickly overtaking paper-based systems, not only in large businesses, but also in small businesses, residential offices, and the consumer marketplace. As they do, the security of electronic messaging will become an ever-increasing issue. Large businesses typically have a complete infrastructure in place to deal with security issues. Small businesses, however, are just beginning to use electronic messaging and electronic payment systems, and residential office users and consumers have generally had no exposure to this level of technology. This paper describes both the historical background of security controls in paper-based systems, and the current market needs, technology, standards, and future directions of security controls in electronic messaging systems.

Introduction

Electronic messaging systems are quickly replacing paper systems. The ability to create, send, file, search, retrieve, and process information electronically has, in large measure, been the impetus for this replacement. As these electronic systems proliferate and replace many paper systems in the business environment, they are also being used in small business settings and residential offices, and by the consumer. Witness the retail sales of PCs, fax machines, and other information movement and management systems as evidence of this fundamental change in the degree and character of usage prevalent in electronic messaging systems.

Big business is already using electronic messaging, and it has begun to put controls in place to deal with associated security issues. In some cases, however, the technology has evolved more rapidly than the controls. In other cases, the need for basic security controls has not yet been recognized, even for some of the more fundamental electronic messaging applications. This paper describes the broad set of security functions that are needed in messaging systems, and their current and future trends.

Historical Perspective

Security controls for paper-based messaging and transaction systems have evolved over thousands of years. The essential needs for the originator or recipient of a message in a paper-based system include the ability to:

- Identify the originator of a message or transaction,
- Verify the identity of the originator,
- Ensure that the message or transaction received is the same as that sent, without accidental or intentional modification,
- Prove to a third party that the transaction actually took place,
- Keep communications private, and
- Survive any failure of the communication system's supporting infrastructure.

In paper-based systems all these needs are addressed by conventions, widely recognized by users, that are applied almost without thought. We discuss these conventional paper-based security controls to motivate the need for security in electronic systems and to provide a basis for analogy with electronic systems.

Security Services

Security systems are characterized by the abilities described in the previous section. For the following discussion, we have drawn on the security services defined by the International Organization for Standardization (ISO)[1] and the European Computer Manufacturers Association (ECMA).[2] (See Panel 1 for

Panel 1. Abbreviations, Acronyms, and Terms

ANI — automatic number identification

DSS — Digital Signature Standard

ECMA — European Computer Manufacturers Association

hashing — In computer science, to apply an algorithm to a message that efficiently combines every piece (bit) of it into an output that represents the entire message. The hash value will generally be shorter than the original message. For security purposes, any change in the message will necessitate a change in the hash value.

ITSEC — Information Technology Security Evaluation Criteria

KDC — key distribution center

PIN — personal identification number

RSA — Rivest-Shamir-Adleman public key encryption algorithm

SHA — Secure Hash Algorithm

definitions of abbreviations, acronyms, and terms.)

Identification. In paper-based systems, the name and/or title of the sender serves as identification. In commercial transactions, preprinted stationery with a corporate logo often identifies the originator's organization or position when the specific person is unknown to the recipient.

Authentication. Written signatures are the primary form of authenticating the end-user. In early times and in certain cultures, wax seals were imprinted with the symbol of an important individual or office. Currently, the embossed seal of a notary public provides a higher level of authentication of an individual. The notary public serves as a trusted third party who verifies that an individual properly identified and authenticated himself or herself at a given place and time. Therefore, the trust that the recipient places in the notary public is transferred to the originator.

Integrity. In paper-based systems, transactions are recorded in ink on non-erasable documents. Special papers have been developed that display certain indicators if they are modified.

Nonrepudiation. To prevent an individual from denying having engaged in a paper-based transaction, procedures have been established to verify individual signatures, keep duplicate copies of transactions, and entrust third parties to adjudicate disputes.

Confidentiality. Physical protection mechanisms have evolved to ensure the privacy of paper-based transactions. The glue seal on an envelope, the wax seal on a document, double wrapping of sensitive documents, etc., are used to discourage others from reading a paper message. In early societies, where most people were illiterate, writing itself provided privacy from the masses. As more people became literate, additional privacy controls were added. Cryptography, the basis of much of the security in electronic messaging and transactions systems, is almost as old as writing itself. The Caesar cipher and the early Hebrew atbash ciphers are early examples of cryptography.[3]

Availability. By their very nature, paper-based systems are highly survivable, that is, even if a portion of a system were destroyed, the remainder would continue to operate. Transactions can be created on a wide variety of media using a vast array of tools. Broken pencils, a lack of 8-1/2 × 11 white paper, etc., do not prevent messages from being recorded. In the U.S., acts of nature, vandalism, or bombings of post offices and mail boxes may destroy several transactions in transit, but do not significantly impair the availability of the entire system. A high degree of parallelism and established procedures to deal with such incidents have evolved over time, as have other paper-based security controls.

Current Market Needs

The same message or transaction that was handled in the past by a paper-based system is now being sent by an electronic system, to improve the speed of communications, cost of handling, and the compatibility with the generation, transmission, reception, or storage system. All evidence points to an ever-increasing rate of evolution toward electronic systems.

As this evolution progresses, the essential security needs of users are not diminishing. In fact, with easy access to more information on-line, the threats to confidentiality, integrity, availability, and other aspects of security are likely to increase. Attacks on paper-based messaging systems generally require physical access to one of the relatively few copies of a paper message.

In contrast, an electronic messaging system may store multiple copies of entire messages, or pieces of

Panel 2. The RSA Algorithm

Signing or sealing messages by analogy. Imagine that Alice wants to leave a paper message for Bob in a public place in such a way that Bob knows that only Alice could have left it. (Historically, Alice and Bob are the actors used in describing cryptographic protocols.) By prearrangement, Alice tells Bob the combination of a lock that she has had made for her exclusive use, and they agree on a suitable public location where messages will be left (similar to the lockers at a bus terminal). At some future time, Alice can deposit a message in the lockbox, securing it with her personal lock, and inform Bob that the message is ready to be retrieved. Bob can then unlock the combination lock with the combination that Alice has provided, quite confident that no one else has surreptitiously inserted a false message. The same arrangement can be used if Alice wants to send Bob a message that both are sure no third party can read. In this case, Bob provides the combination lock to Alice (in an unlocked state), but does not reveal the combination to her. Alice can then lock her message in the locker, being assured that no one, other than Bob, knows the combination and can retrieve the message.

The algorithm. Instead of using combination locks, paper messages, and lockers, the RSA algorithm uses hard mathematical algorithms to compute message exchanges similar to the exchanges above. Analogous to the lock and its combination, the RSA algorithm relies on two cryptographic keys, intimately related to each other but underivable from the other. To sign a message, m, Alice would reveal her public key, e_a, but not her secret key, d_a. Signing a message then requires only that the message be encrypted with e_a. Anyone can verify the signature by using d_a to decrypt the message. Correspondingly, RSA could be used to send messages secretly if Bob gave Alice his public key, e_b, to encrypt the message, and then used his private key, d_b, to decrypt the message.

Mathematically, the two keys used are multiplicative inverses of each other in a finite field of size n, where n is the product of two large prime numbers, p and q. Both p and q are kept secret, but n can be published. Anyone knowing p and q can pick an e, relatively prime to $(p-1)\times(q-1)$, and can easily compute d, the multiplicative inverse of e in the field. Anyone else, who could know only n and d, will have great difficulty in computing e, perhaps needing to expend as much effort as trying to break n into its two prime factors.

Encrypting (or signing) a message, m, requires that the originator calculate a cipher, c, where

$$c = m^e \pmod{n}$$

Decrypting (or verifying the signature) requires that the recipient calculate

$$m = c^d \pmod{n}$$

Of course, since c and d are multiplicative inverses,

$$(m^e)^d = m^{ed} = m^1 = m$$

and the original message is recovered.

messages. A sender or recipient of a message generally does not know exactly which nodes of a network carried his or her message. In fact, it has been reported that popular word processing packages do not always permanently erase sections of documents that the author has deleted. Therefore, when files are shared, some private notes, background material, or thoughts that have been temporarily entered into the document may be divulged to those who have sufficient knowledge of the word processing system's quirks. In the recent past, only a few highly trained, trusted people had access to the internal operations of the computers and applications programs that comprised electronic messaging systems.

More recently, with the availability of personal digital assistants, desktop PCs, low-cost file servers, etc., control of systems resources has shifted from a centralized, specialized staff who had in-depth knowledge of electronic security issues to an untrained staff. An end user may also be the system administrator and/or the programmer, giving that user unprecedented access to all messages, even those of co-workers. Because these users-turned-administrators may not understand the full security implications of their actions, or may not be trustworthy, unexpected security vulnerabilities may arise.

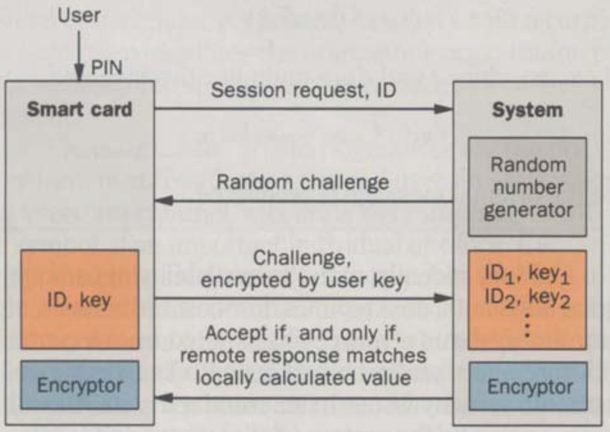
These issues, as well as the potential for undetectable, remote (that is, nonphysical) access to systems storing or transmitting electronic messages, require that system designers pay even more attention to system security requirements than did their paper-based-messaging predecessors. Fortunately, today a wide variety of technologies are becoming available to address many of these issues.

Panel 3. Strong User Authentication Methods

In an electronic system, authenticating a user's identity can be based on only a few classes of user-specific data. A user can be authenticated based on:

- What they know (information, e.g., a personal identification number [PIN] or password),
- What they have (a physical device, e.g., the key to a lock or an electronic device),
- What they are (physical properties of the user, e.g., fingerprint, sound of his/her voice, etc.), and
- Where they are (actual location in a network, e.g., automatic number identification [ANI] or network address).

The best user authentication techniques combine more than one piece of user data, reducing the likelihood that an attacker would possess several independent items simultaneously. One method of authenticating users that is gaining wide acceptance is the smart card. A user requesting access to a smart card system must have one and must know a PIN to unlock it. Such devices are often used to implement dynamic authentication techniques that avoid the pitfalls of a static, easily guessed user password. The authentication exchange in such a smart-card-based system might look like this:


User
PIN
Smart card
Session request, ID
System
Random challenge
Random number generator
Challenge, encrypted by user key
ID, key
ID_1, key_1
ID_2, key_2
⋮
Accept if, and only if, remote response matches locally calculated value
Encryptor
Encryptor

Technology

With the public availability of strong cryptographic systems[4] and the development of new methods of encryption (see Panel 2),[5] many essential technologies needed to protect electronic messaging systems already exist. For example, effective encryption systems provide ways to authenticate users (see Panel 3) and system nodes with high levels of confidence. Encrypting a message is the historical way of assuring that it remains confidential. Creating encrypted checksums[6] that can be appended to a message ensures its integrity in the face of intentional modification, the same way that error-detection codes protect a message from accidental modification as it travels through an error-prone network. In short, while cryptographic processing does not, in itself, guarantee that messages cannot be read, modified, or fictitiously generated, it does provide a strong foundation for dealing with many of these issues.

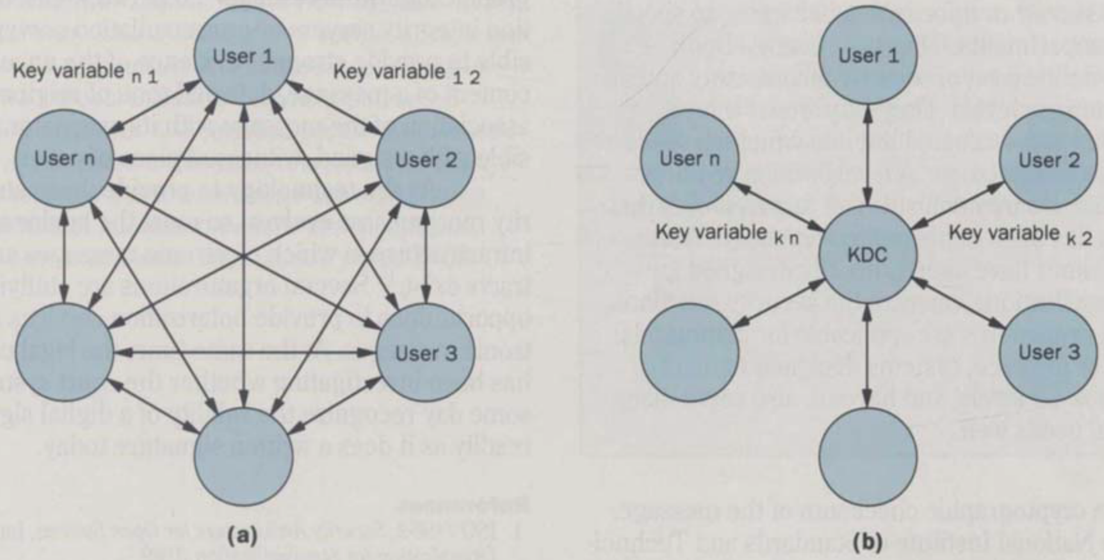
There are two distinct classes of cryptographic systems: *conventional cryptosystems* and *public key cryptosystems*. Conventional cryptosystems, also known as "private key," "symmetric," or "one-key" cryptosystems (see Panel 4), use the same encryption key variable to encrypt a message as they do to decrypt it. In such cryptosystems, all parties to a communication must share the same key variable for each subnetwork in which they wish to communicate securely. Public key cryptosystems, also known as "two-key," or "asymmetric," cryptosystems, use different key variables to encrypt information than they do to decrypt it. The Rivest-Shamir-Adleman (RSA) algorithm (see Panel 2) is one example of a public key cryptosystem. Both key variables are interrelated, but one cannot be easily derived from the other. Users may freely distribute their encryption key variables with little worry that someone will be able to decrypt their messages.

Another relevant technology that has recently become available is that of trusted systems (see Panel 5). In response to the needs of the military and intelligence community, several vendors of computer operating systems have created systems that can be formally verified to meet certain security requirements. For instance, trusted systems can offer their operators assurance that one user or process being served by the operating system cannot access the information of another user or process. In the commercial world, not every application or system needs to, or can afford to, be inspected or

Panel 4. One-Key Cryptosystems

All parties wishing to share messages in a one-key cryptosystem use the same key to encrypt or decrypt their messages. If they would rather communicate in several distinct subnetworks — sharing messages with certain other members of the network, while keeping some of their messages secret from the rest — they may need many individual keys. In fact, without some type of hierarchical key management, n users will require on the order of n^2 different key variables, as illustrated in (a) below.

To avoid such a large number of distinct key variables, hierarchical key schemes are used where individual users or nodes have data-encrypting key variables for their messages, and key-encrypting key variables for exchanging key variables. A scheme that minimizes the number of keys in this type of system relies on a centralized key distribution center (KDC), as shown in (b) below. Such a system can operate with n key variables for n users.



verified with the necessary rigor applied to a military system. However, for critical electronic messaging systems, building on such a secure foundation often provides more assurance of design integrity.

There are two dimensions of security in trusted systems: *security functionality* and *design assurance*. The security functionality dimension in a trusted system includes the types of access control mechanisms that the system provides, preventing one set of users from accessing information owned by another set of users. The second dimension of security is related to the level of design assurance, or the "correctness," of the design and its implementation. While the U. S. Orange Book combines these two dimensions of security into one measure of the "trustworthiness" of the system design, the European Information Technology Security Evaluation Criteria (ITSEC)[7] separates the two. For many commercial messaging systems, high levels of security functionality may be more important than an assurance that the correctness of implementation can be proven mathematically.

Standards

The paper by Griesmer and Jesmajian, in this issue of the *AT&T Technical Journal*, presents an in-depth discussion of messaging system standards. In this paper, we focus on a few of the key standards for security. Earlier we pointed out that it is necessary to provide a secure way to seal and sign an electronic message if it is to be used for the same purposes that paper messages and contracts are used today. Central to this is the ability to compute a cryptographic checksum of the message.

Panel 5. Trusted Systems

The National Computer Security Center (NCSC) has issued a series of design and evaluation guidelines for stand-alone and networked computer systems.[8–11] In this series — often referred to as the Rainbow Series, because the cover of each handbook is bound in a different color — implementers of computer systems and networks are given security guidelines to incorporate into the design of their systems. These guidelines are mandatory for certain military and intelligence applications, where the systems may be processing classified information (e.g., confidential, secret, top secret) or information belonging to specific security compartments. The NCSC Orange Book guidelines define a set of security functionality and design assurance levels. They vary from "D level," for a system that was evaluated but into which no specific security was designed, to "A level," which requires sophisticated security controls and formal proofs that the system has been implemented correctly. While these guidelines have been primarily designed for classified applications, many of the security functions and design procedures are applicable for commercial systems. For instance, systems designed to meet Orange Book C2 levels, and beyond, also serve many commercial needs well.

The National Institute of Standards and Technology has recently published the Secure Hash Algorithm (SHA), to be used with its Digital Signature Standard (DSS). The SHA computes a 160-bit "message digest," or hash, of an arbitrary length message. Any modification of the message is likely to produce significant changes to the resulting message digest, making attempts to modify the message easily detectable. Like all secure hashing functions, the SHA is designed to discourage computing a second (fraudulent) message that has the same hash value as the original message.

Once a message digest has been computed, it can be signed using the DSS, in a manner similar to that used for other public key algorithms. The advantage of signing the message digest, instead of the message itself, is that the resulting large-modulus arithmetic need only be performed on the comparatively short message digest. Security of the exchange is not compromised, because the attacker would have to work backwards through the signature function and the hashing algorithm, both of which are designed to make this as difficult as possible.

Future Directions

Many vendors are responding to the needs of customers to provide the same level of security in electronic media that is available in paper media.[12] In some cases, the capabilities provided in electronic systems are difficult, if not impossible, to duplicate in paper-based systems. For instance, based on the strong cryptographic algorithms available in network-based information integrity servers or nonrepudiation servers, it is possible to provide stronger evidence of the uncorrupted content of a message, date and time of origination, or association of the message with its originator than is possible with a signed, witnessed piece of paper.

As the technology to provide these strong security mechanisms evolves, so must the business and legal infrastructure in which electronic messages and contracts exist.[13] Several organizations are studying the opportunities to provide notarization services for electronic messages. At the same time, the legal community has been investigating whether the court system will some day recognize the validity of a digital signature as readily as it does a written signature today.

References

1. ISO 7498-2, *Security Architecture for Open Systems*, International Organization for Standardization, 1989.
2. ECMA/TR-46, *Security in Open Systems — A Security Framework*, European Computer Manufacturers Association, July 1988.
3. Philip Kahn, *The Codebreakers*, Macmillan, 1979.
4. FIPS PUB 46, *Federal Information Processing Standard*, National Bureau of Standards, January 1977.
5. Bruce Schneier, *Applied Cryptography*, John Wiley & Sons, 1994.
6. 6. ANSI X9.9, *American National Standard for Financial Institution Message Authentication (Wholesale)*, American Bankers Association, 1982.
7. Information Technology Security Evaluation Criteria (ITSEC), *Harmonized Criteria of France — Germany — the Netherlands — the United Kingdom*, May 2, 1990.
8. *The Orange Book — Trusted Computer System Evaluation Criteria*, DOD 5200.28-STD, Department of Defense, December, 1985.
9. *The Red Book — Trusted Network Interpretation of TCSEC*, NCSC-TG-005, National Computer Security Center, July 31, 1987.
10. L. K. Barker and L. D. Nelson, "Security Standards — Government and Commercial," *AT&T Technical Journal*, Vol. 67, No. 3, May/June 1988, pp. 9–18.
11. C. W. Flink and J. D. Weiss, "System V/MLS Labeling and Mandatory Policy Alternatives," *AT&T Technical Journal*, Vol. 67, No. 3,

May/June 1988, pp. 53–64.
12. "Bellcore spins off new company to offer services," *OTC NewsAlert*, March 22, 1994.
13. Announcement of meeting of American Bar Association's Notarization and Nonrepudiation Work Group, *Internet TELECOM Digest*, March 21, 1994.

(Manuscript approved April 1994)

***Diana M. D'Angelo** is a member of technical staff in the Security and System Reliability Architecture Group of AT&T Bell Laboratories in Holmdel, New Jersey. Currently, she is working on a special assignment in the Applications Architecture Department of AT&T Global Business Communications Systems in Denver, Colorado. She develops secure system architectures and is interested in computer and network security. Ms. D'Angelo received a B.S. from Baylor University, Waco, Texas, and an M.S.E. from the University of Pennsylvania, Philadelphia, both in computer science. She joined AT&T in 1988.*

***Bruce McNair** is technical manager of the Security and System Reliability Architecture Group in the Applications Architecture Department of AT&T Bell Laboratories in Holmdel, New Jersey. He is responsible for both assessing system requirements and specifying solutions for security, privacy, fraud, and reliability issues. Mr. McNair joined AT&T in 1978 after receiving a B.E. in engineering and an M.S. in electrical engineering from Stevens Institute of Technology, Hoboken, New Jersey.*

***Joseph E. Wilkes** is a distinguished member of technical staff in the Applications Architecture Department of AT&T Bell Laboratories in Holmdel, New Jersey. He works in the areas of wireless communications, privacy, security, and consumer fraud. Mr. Wilkes joined AT&T in 1972, after receiving a B.E. from The City College of the City of New York, Manhattan, and an M.S. and Ph.D. from The Polytechnic Institute of Brooklyn, all in electrical engineering.*