NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

**SECURITY MANAGEMENT
OF
ELECTRONIC DATA INTERCHANGE**

by

Pao, Hua-Fu

June 1993

Thesis Advisor: Jon T. Butler

Approved for public release: distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Administrative Sciences Dept. Naval Postgraduate School; 6b. OFFICE SYMBOL (if applicable): AS	7a. NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification): SECURITY MANAGEMENT OF ELECTRONIC DATA INTERCHANGE

12. PERSONAL AUTHOR(S): Pao, Hua-Fu

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM TO	June 1993	139

16. SUPPLEMENTARY NOTATION: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Electronic Data Interchange, Digital Signature, Security Management, Security Management Model

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis considers the security management issue of electronic data interchange (EDI) and the security tools that are used for secure EDI implementation. Management considerations of EDI are considered to be an integration of EDI and security services. The background of EDI is surveyed along with EDI implementation procedures and Digital Signature (DS) techniques. The security services that are being used today are discussed in Chapter 5 including the EDI-related security standards. Finally, a security management model is presented that may be used to improve the security of an EDI implementation. The model is based on the perceived threats and vulnerabilities to the EDI, and the availability of security services and mechanisms. Included in the model are the setting of security goals.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
[X] UNCLASSIFIED/UNLIMITED [ ] SAME AS RPT. [ ] DTIC USERS	UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL: Jon T. Butler	22b. TELEPHONE (Include Area Code): (408) 656-3299	22c. OFFICE SYMBOL: EC/BU

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted; All other editions are obsolete

the determination of security policies, the determination of priorities for EDI security, and the construction of security architecture and management activities.

This thesis was written to survey the concepts of EDI and DS. The survey of EDI-related security standards may provide an opportunity to understand security services and security mechanisms available when designing an EDI system. Also, the security management model may help to improve the security of an organization.

Approved for public release: distribution is unlimited

SECURITY MANAGEMENT OF ELECTRONIC DATA INTERCHANGE

by
Pao, Hua-Fu
Lieutenant, R.O.C. Navy
B.S., National Defense Management College, 1987

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
June 1993

Author: *Pao, Hua-Fu*

Approved By: *Jon T. Butler*, Thesis Advisor

Roger Evered, Second Reader

David R. Whipple, Jr., Chairman,
Department of Administrative Sciences

ABSTRACT

This thesis considers the security management issue of electronic data interchange (EDI) and the security tools that are used for secure EDI implementation. Management considerations of EDI are considered to be an integration of EDI and security services. The background of EDI is surveyed along with EDI implementation procedures and Digital Signature (DS) techniques. The security services that are being used today are discussed in Chapter 5 including the EDI-related security standards. Finally, a security management model is presented that may be used to improve the security of an EDI implementation. The model is based on the perceived threats and vulnerabilities to the EDI, and the availability of security services and mechanisms. Included in the model are the setting of security goals, the determination of security policies, the determination of priorities for EDI security, and the construction of security architecture and management activities.

This thesis was written to survey the concepts of EDI and DS. The survey of EDI-related security standards may provide an opportunity to understand security services and security mechanisms available when designing an EDI system. Also, the security management model may help to improve the security of an organization.

TABLE OF CONTENTS

I. INTRODUCTION

A. BACKGROUND

The increasing use of electronic transmission and storage of documents has given rise to new requirements for document security and authentication. In particular, it is necessary to consider schemes that provide at least similar authenticity to that provided by traditional paper documents and written signatures.

B. PURPOSE

This thesis describes methods that may be used to improve the security of electronic data interchange (EDI). Additionally, it discusses issues of applying digital signature (DS) as security tools on EDI. These methods are based on perceived threats and vulnerabilities that could occur when using EDI. The thesis also presents the proper security management of organizations and their trading partners. The contribution of this thesis, first of all, is to formalize the concept of EDI and DS. Secondly, it surveys threats and vulnerabilities that can occur in the implementation of EDI. Thirdly, it surveys EDI-related security standards that impact security services and security mechanisms for EDI. Finally, a security management model is built that can be used to improve the security of an implementation of EDI. This process is based on perceived threats and vulnerabilities and considerations of security services and mechanisms. This process includes the setting of security goals, the determination of security policy, the determination of priorities for EDI security, the construction of security architecture, and management activities. The integration of EDI and security services is the principle management consideration of EDI.

C. METHODOLOGY AND ORGANIZATION

EDI and DS systems are first surveyed in Chapters II and III. The survey focuses on what information system (IS) managers need to know if they want to implement EDI and secure their business information. Having introduced the state-of-the-art of EDI and DS,

we discuss in Chapter IV the security problems that EDI users will face. The vulnerabilities and threats which emerge from the nature of EDI components (e.g., software, hardware, platform, and management skill) and the required security services and security mechanisms that can be utilized to prevent those security problems are also mentioned in Chapter IV.

How to apply security standards and their security structures that have been recommended by several international organizations is presented in Chapter V. The security services proposed by those standards are widely used by EDI vendors when implementing EDI. Finally, we build a security management model. This model will consider the security goals of organizations, the security policy and several managerial considerations of EDI implementations that will improve the security of an organization. Chapter VI concludes the thesis with suggestions.

II. ELEMENTS OF ELECTRONIC DATA INTERCHANGE

A. INTRODUCTION

The concept of EDI started over 30 years ago. During the 1960s, the Transportation Data Coordinating Committee (TDCC) of U.S. became concerned about the strangling effect of paperwork in the transportation industry. In 1975, the TDCC published the first set of rules for EDI, many of which still apply. At that time, computer hardware, software and networks were not capable of supporting these new business procedures. In 1979, the American National Standards Institute (ANSI) approved TDCC's approach to EDI access and use, as embodied in the ANSI X12 standard. Since then, the ANSI X12 Committee has generated over 20 cross-industry transaction data format variations of the basic theme to accommodate unique situations in diverse businesses.

B. DEFINITION OF EDI

EDI is the intercompany, computer-to-computer exchange of business documents in standard electronic data formats. Transaction data is transmitted from the sender company's application to the receiver company's application without human intervention. Transactions (also called transaction sets) include invoices, shipping schedules, advance ship notices, court filing, bills of lading, and purchase orders. These are transformed to a standard data format and electronically transferred between trading partners without utilizing hard copies and re-keying information. Standard transaction sets are approved by ANSI in the X.12 standard.

Although EDI and E-mail are similar, significant differences exist. The difference between EDI and E-mail has been defined according to [Ref. 1] as follows:

- EDI - Electronic communications made up of predefined, fixed-format message-length units.

- E-mail - Electronic communication made up of free-formatted message-length units.

C. TYPES OF EDI

Various systems implement different types of EDI. The specific implementation depends on the roles of the participants and the operation process of EDI. Here, we discuss two types of classification schemes of EDI.

1. Roles of Participants

One can classify the types of EDI by the roles of participants. These include:

a. One-to-Many System

The central point is a single company, sometimes, a large company (typically called "hub") with many trading partners (usually called "spokes"). These participants are usually the suppliers for the company, their dealers, or the customers. For example, a single automobile manufacturer buys various parts from numerous suppliers and then sells cars to their dealers and customers. This may be achieved by directly connecting the EDI systems of the single manufacturer to many trading partners.

b. Valued Added Network (VAN) System

In the VAN system, the central point is less defined and the application system takes on an electronic warehouse flavor, with many buyers and sellers interacting. It is a logical extension of the one-to-many system. For example, several automobile manufacturers are buying parts from the same suppliers and selling cars to the same dealers. They can link through a third-party network and buy the communication services from it. This third-party network may provide translation to different data formats, security services, network maintenance, EDI implementation consulting and other value-added services.

c. Incremental Paper Trail System

The incremental paper trail system is the extension of the VAN system. In the incremental paper trail system, transaction documents are not created at a single point. Instead, they pass through a chain of intermediaries, each of which adds to the information

and documents. For example, VAN systems can link with financial institutions and freight forwarders to transfer funds electronically and add documentations for export.

2. Process of EDI

Another way to classify EDI systems is by process.

a. Batch Processing EDI

Groups of transactions from various participants are sent to a platform, which places them into storage (like mailboxes). Trading partners then retrieve them at a later time after the transactions are updated by batch processing. The average processing time is from one hour to one day.

b. Real-time EDI

The communication link through the VAN or other platforms is maintained until the transaction is complete. Usually, real-time EDI ranges from one to 10 minutes. Examples of real-time applications are order processing, just-in-time inventory management (JIT, an ordering strategy that helps organizations to keep down inventory by careful scheduling and transportation), retail quick response (QR), transportation inquiries and price quotation systems. [Ref. 2]

c. Event-driven EDI

When a document is received by a company, the generation of another document is triggered. For example, if a merchandise release document comes from an auto manufacturer to its supplier, that system will automatically trigger an automatic ship notice. Event-driven EDI is between real-time and batch processing EDI. Another name is fast batch. [Ref. 3]

Real-time EDI seems to be the best choice, when using high volumes of small, standardized transaction sets. Unfortunately, not every EDI system is capable of working in real-time.

D. LEVEL OF EDI

There are three implementation levels of EDI: end-to-end communication, intra-integration and interexploitation. [Ref. 4]

1. End-to-end communication (Level 1)

This is the foundation of EDI installation. It consists of a computing platform, translation software and communication links. This is the least expensive approach and accordingly, offers the fewest benefits. Transactions received at this level may be keyed into another system or turned into paper to be delivered. No business process reengineering is required in implementing this EDI.

This system is viewed by the organization more as a communication system than an information system. Emphasis is on the trading partner-initiated aspects of the system. Usually, this system is implemented to respond to a small number of trading partners and a limited number of business applications. Most EDI implementations are of a Level 1 type. Level 1 EDI generally applies to smaller spoke companies on simple networks.

One example is a single product shop that has specific product suppliers. EDI software may be bought depending on what kind machines and type of network that already exist, without considering the integration of the information systems. Also, the application may be used to only the part of transactions needed by the EDI system for the suppliers. If other suppliers do not have an EDI system, they may still remain on paper work.

2. Intraintegration (Level 2)

This represents a Level 1 system that interoperates and integrates with other information systems like a database management system. Level 2 systems affect business operations and have a high likelihood to change business practices. Business processing reengineering is likely to have occurred at this level. Application integration and interoperabiliity with other systems are the primary characteristics of this EDI level. This is a moderately expensive solution, but has the potential for significant benefits. The Level 2 system heavily impacts intracompany activities. Usually a large number and perhaps

hundreds of trading partners are involved. Level 2 systems have the potential to be mission-critical. Level 2 approaches may extend JIT or QR across the business organization. These implementations are mostly being performed by large-scale EDI hub organizations on mainframe systems.

For example, a company may interconnect their EDI system with their database management system so that they can store and forward special transaction sets without re-keying (purchasing orders, invoices, etc.).

3. Interexploitation (Level 3)

This is a Level 2 system that is more thoroughly integrated with business practices on both ends of the communication link. Level 3 systems are expensive and require careful implementation over time. Their benefits may be very significant. The trading partners are providing information to each other that was formerly classified as internal information only. This system level allows trading partners to plan ahead for materials, machine time, capacity and people. Level 3 systems may be mission-critical systems. Because of complexity, only a few trading operations have achieved Level 3. This group includes several transportation companies that have integrated systems with their customers' needs. These systems require trust between partners.

An example of Level 3 could be a supermarket that shares its anticipated shelf withdrawals and special promotions with manufacturers who will use the information for their production planning schedules.

E. COMPONENTS OF EDI

According to the implementation needs of EDI users, we can divide EDI systems into:

1. EDI Software Packages

There are many EDI related software packages. Among the functions provided by these packages, the most important is translation.

a. Data Mapping System

Data mapping creates special data maps to interface one application with other applications. This system also has the ability to move any portion of data from the EDI transaction sets to the application records.

For example, a purchase order received electronically may be automatically entered and stored into the database system format through a data mapping system. Another example is an invoice developed from the accounts receivable software and transmitted to the recipient without human intervention. On-line data mapping modules can simplify EDI integration within applications.

b. Translation Software

Translation software programs validate and edit raw information from a data file, add control parameters and delimiters, and prepare the information to be communicated among trading partners using a trading partner file. Some translation software possesses the ability to turn around data in incoming EDI messages into outbound EDI transaction sets. Some have the ability to enter bar-code data. Retailers can carry handhold scanners, portable point-of-sale (POS) terminals and long-range laser scanning, allowing a bar code to be read from more than 15 feet away. These devices communicate via radio frequency (RF) signals with host systems or with stationary terminals. All data collected in this manner may be converted directly to EDI standard format without reentry. Inquiries may also be made directly to the database. [Ref. 5]

2. Networks Over Which EDI Occurs

These services represent a platform for EDI supplied as follows:

a. Standard Provided Network

One can send EDI documents on a network provided by telephone carriers or by national institutions. This platform is free of charge and includes normal telephone lines and gateway systems, such as the INTERNET and the CCITT X.400-based system.

b. Third-party EDI Networks

A VAN is usually a privately owned packet-switch network whose services are sold to the public. Such a system provides conversion of information between EDI systems with incompatible message formats, communication services, and protocols. VANs act as translators between different computer systems used by the buyers and suppliers. Normally, VANs are more costly than standard networks as VAN services charge fees, network while standard networks are free following development. VANs often offer the most practical, rapid and secure solution for communication network needs. Decisions about the use of VANs depend on the characteristics of the organization and the needs of the organizations involved, such as message conversions, protocol conversions, store and forward services to provide processing independence and other database/ information services (i. e., catalogues, schedules, etc.).

c. The Interoperability of Network.

As many trading partners will not be on the same EDI platform, interoperabiliity between platforms and the management of security of transactions over these multiple platforms is critical. Interoperability requires:

(1) The ability to exchange EDI data or cryptographic keys, both manually and automatically. In an automated environment, any party that implements this service will use both compatible options of the relevant standards and compatible communications facilities. These are issues that the mailbag protocol and CCITT X.435 are designed to solve.

(2) The ability to support standards. This capability will give VANs competitive advantages as standards can reduce the barriers of entry to EDI and increase the acceptance of newcomers.

(3) The ability to link E-mail, Fax and EDI communications. EDI carried by E-mail has been designed. The standard which applies is the CCITT X.435, on the market

one and half years. The problem of "How to link Fax with EDI without human intervention?" still exists. Future EDI systems are expected to become more friendly.

(4) The ability of integration. Integration capability allows for an enterprise-wide approach to EDI with multi-vendor computing environments that run multiple platforms for exchanging documents with trading partners.

(5) The ability of international communication. Since business is no longer limited to single countries, the ability to translate different languages with EDI software via international networks is a future trend of EDI implementation.

F. BENEFITS OF EDI

1. Increased Productivity

EDI mitigates repeated data entry and manual handling of transactions that occurs in non-EDI environments, making organizations more productive. An estimated 25% of the cost of processing business transactions is related to basic data entry and associated tasks, according to the estimates by Cashin [Ref. 6].

2. Decreased Lead Time and Inventories

As electronic order requests are transmitted more quickly and accurately than paper work, EDI's instantaneous communication makes inventory reduction safer. It also reduces the lead time previously necessary for purchase transactions. This can be achieved with the support of JIT systems and QR programs using real-time EDI systems for transportation inquiries and price quotation systems.

3. Increased Data Accuracy

EDI communication is direct, instantaneous, and immediately verifiable. EDI transaction processes may occur without human intervention. EDI transactions are working through the standard data format and interchanging electronically and automatically. Those reasons make the EDI data more accurate.

4. Cost Saving

Cost saving is achieved by eliminating paper work and improving bill reconciliation and telephone calls. Cost reduction is still a long-term issue, but vastly improved business procedures over a broad range of activities is an important goal as well.

5. Reducing Time of Processing Invoices

Prior to EDI, all invoices were processed by humans. Sending/receiving and even filling the format of invoices required significant staff. All this work may now be done with EDI software. These procedures include storage of the history invoices, sending/receiving documents automatically on EDI platforms, and translating various invoices of different trading partners, etc.

6. Establishing an Integration Base

Establishing the base for further integration of materials and manufacturing systems is another benefit of EDI. By improving communication with suppliers, EDI enhances production scheduling accuracy. This means less downtime due to late shipments from suppliers. Quality assurance information may be received electronically from suppliers, enabling production schedules to electronically feed directly into production control systems for shipments.

7. Improves Trading Partner Relations

According to most users, the most important benefit of EDI has been improved trading partner relations. Because both parties must add benefits to allow the EDI system to function, EDI encourages a tremendous degree of cooperation between trading partners. When designing an EDI system to serve the information-exchange needs of both parties, trading partners must share their experiences with each other.

8. As a Competitive Tool Among Industry Competitors

In the past, any company that implemented various proprietary EDI systems gained tremendous benefits. They possessed a greater ability to compete with those

competitors who did not implement EDI. In the future, as more companies accept EDI, security services and the difficulty of implementation of these services will become the main competition issues.

G. STANDARDS IN EDI

The exchange of electronic documents between companies in standard data formats is fundamental to the definition of EDI. Only by using strict EDI standards can computers transfer and process information automatically. There are two standards that apply:

1. External Communication Standards

a. CCITT X.400 Message Handling System (MHS)

CCITT X.400 is a member of the Open System Interconnection (OSI) standard family. The OSI standards are intended to maintain the ease of connections between heterogeneous networks. CCITT X.400 is a global store-and-forward message standard for E-mail. CCITT X.400 is frequently viewed as an intercompany messaging, mail interchange and transport system. The CCITT X.400 protocol will be the basis for many new applications that use interconnected E-mail system. CCITT X.400's acceptance has been slowed because the protocol's 1984 version lacked important futures such as security, but the 1988 version has overcome those limitations. This version offers security services to guarantee reliability in the global E-mail system. This security issue will be further addressed in Chapter V. The standard was also enhanced in 1992 to allow implementation of EDI, Electronic funds transfer (EFT), facsimile and multimedia applications.

The benefits of using X.400 MHS to carry EDI data include:

- a single communications path using X.400, where customers can expect to realize economies of scale which allow saving of from 10-30%, over previous multiple connection solutions depending on volume;

- the ability to exchange messages with multiple trading partners who may use different messaging systems, networks, or computers;

- the ability to communicate electronically to trading partners not using EDI;

- the ability to transmit multimedia from one interface (i.e., EDI, E-mail, fax, telex, etc.);

- enhanced security capabilities including encryption and network-based acknowledgment of receipt.

b. CCITT X.435

The title for CCITT X.435 "Message handling systems: EDI messaging system was adopted in 1991. This standard provides a comprehensive blueprint for a message handling system (MHS). The purpose of an MHS is to enable users to exchange messages on a store-and-forward basis. A message submitted on behalf of one user, the originator, is conveyed by the message transfer system (MTS) and subsequently delivered to the agents, the recipients. Access units (AU) link the MTS to other kinds of communication systems (e.g., postal systems). A user is assisted in the preparation, storage, and display of messages by a user agent (UA). Optionally, it is assisted in the storage of messages by a message store (MS). The MTS comprises a number of message transfer agents (MTA), which collectively perform the store-and-forward message transfer function.

The CCITT X.435 defines message handling applications called EDI messaging (EDIMG). It also defines a form of message handling tailored to the exchange of EDI information, a new message content type, and associated procedures. It is designed to meet the requirements of users of ISO 9735 (EDIFACT), and other commonly used EDI systems. The implementation of the CCITT X.435 standard should enable users to combine their messaging and EDI networks, which will reduce administrative overhead costs and simplify network management. In addition, CCITT X.435 contains enhanced security, tracking, and auditing.

Messages described in CCITT X.435 include the message body parts and the message envelope. The message body part consists of header, trailer and subbody part. The

message envelope for CCITT X.435 specifies that the message which contains an EDI document, allowing the user to quickly identify and then automatically route incoming messages to mailboxes or EDI applications. In addition, the CCITT X.435 header gives users additional information about the documents inside the envelope, such as the name and address of the sender and the translation software used. This will allow users to sort messages before processing them. This standard will also let users attach other message body parts to the message, which might contain such things as computer-aided design drawings, text files, spreadsheets, and encryption data. The standard enables users to cross-reference these message body parts with the original EDI document, enabling users to route the message body parts or documents to different people or applications within the users without losing track of documents that belong together. Most users interested in CCITT X.435 are users with well-established EDI programs and CCITT X.400 networks. Unfortunately, an implementation of CCITT X.435 is not yet possible with commercial software. [Ref. 7]

2. Document content Standards

The second major area for EDI standards is the content of the transaction record or document itself. It is application oriented. Once reliable communication is established, it is practically assured that the data sent is in fact the data received. The data format is what makes the data understood by both trading partners.

Document content standards solve this problem. Since both parties implement EDI documents in the same format, they can understand each other. As with communication standards, several document content standards exist. We now separate the standards into four types:

a. Proprietary Standards

From the beginning of EDI, only one or two companies have dominated the market. Therefore, those companies implementing EDI must use proprietary standards. This is especially true when a large hub is trading with a lot of spokes.

b. Industry-specific Standards

Industry trade groups started by establishing their own standards to address their special needs. In the U.S., those standards now fall under the ANSI X12 umbrella, including standards such as Transportation Data Coordinating Committee (TDCC), Uniform Code Standard (UCS), and Warehouse Information Network Standard (WINS), etc.

c. National Cross-Industry Standards

ANSI has made a significant contribution to the development of EDI transactions across industry lines. In 1978, the Credit Research Foundation asked ANSI to approve the formation of a committee to develop a national cross industry standard for EDI. ANSI sanctioned the Accredited Standards Committee X12 (ANSI ASC X12) to develop the necessary standards. This committee was made up of representatives of commercial and industrial organizations and vendors of services designed to facilitate the use of EDI standards. The X12 committee's charter not only included the development of a cross industry standard, but also the development of an entire family of related standards for electronic exchange of many different types of routine business transactions. Therefore, these standards have almost universal appeal.

The ANSI X12 standards consists of transaction set standards, data element dictionary, data segment dictionary and transmission control standards.

Transaction set standards define the format and context of data used within specified business documents, such as invoices. However, while the invoice transaction set conveys the functionality of the paper invoice, it does not contain as much descriptive information since it is not intended for human reading.

The data dictionary contains codes for types of information used in business documents. The data dictionary reduces vast amounts of information to two-digit codes (called data elements), and therefore, eliminates the need for descriptive information in an electronic document.

The data segment dictionary provides the definition and formats for data segments. These data segments consist of a precise sequence of data elements, separated by delimiters.

The transmission control standards define the formats for the information required to interchange data. Defined within the transmission control standards are data element delimiters, transaction set separators, and transmission envelope formats. [Ref. 8]

d. International Standards

In addition to ANSI ASC X12, there are primarily European standards that are being developed under the auspices of the United Nations known as UN/EDIFACT, an upgrade to EDIFACT. In 1985, the Economic Commission for Europe of UN (UN/ECE) initiated a project to develop intermediate EDI standards based on two existing bodies of work, ANSI ASC X12 and the UN Guidelines for Trade Data Interchange. Both North American and European experts were invited to participate. When applying the international message definitions, subsets are often agreed upon as not all segments are required.

Obviously, a company should choose an EDI standard that will facilitate communication with the maximum number of trading partners. In 1991, NIST published its Federal Information Processing Standard (FIPS) 161. This standard has helped to legitimize EDI in the federal government. FIPS 161 does not mandate the use of an EDI, but suggests the use of either UN/EDIFACT or the ANSI ASC X12. A problem of major concern to industry is the lack of supporting software and the cost and control problems associated with running different systems. There will be a great need to have the support of both ANSI ASC X12 and EDIFACT in the near future.

H. LEGAL ISSUES

1. Authority Aspects

When deciding to implement an EDI system, the first thing a company will face is how to replace the old purchasing order forms, invoices, and some other paper work

related to the original agreement among its trading partners. And, most importantly, who will sign-off on documents? In fact, most firms participating in the EDI survey thought of this as a major legal obstacle to the implementation of an EDI system.

2. Trading Partners Agreements

With documents being sent electronically, contract terms and conditions would no longer be sent to trading partners with each transaction. Often, a standard trading partner agreement is prepared and signed once, and is considered valid for all EDI transactions. Usually, the original (non-EDI) agreements' content are the best. Such agreements include:

a. Payment Terms

How fast should the payment be made? Could payments be completed by electronic funds transfer (EFT)? Such problems need to be stated in the agreement to prevent future disagreements.

b. Liability

What if documents were forged or modified by unauthorized people? Who is responsible for a transmission error? What security services and mechanisms are being used? Who will provide those security tools? All solutions of these problems need to be clarified in the agreement.

c. Need for Acknowledgment

EDI provides for instantaneous acknowledgment. This is accomplished in the "signed receipt requested" letter, when implemented in EDI.

d. Communication Charge

Who will pay the data transmission bill? What kind of EDI platform is going to be used? What documents will be sent by EDI? These questions should also be clearly stated in the agreement.

e. Expiration of the Agreement

The effective period should be stated in the agreement. If digital signatures are applied, the expiration of each signature should be stated.

3. Contracts with EDI Vendors

Liability is the key concern here. Currently, contracts with VANs and software providers leave the user bearing much of the risk of electronic communications. This includes loss or alternation of data during transmission and any business losses that may result. Contracts for off-the-shelf EDI software products are rather inflexible. VAN contracts may be negotiable, depending on the relative size of the user company.

I. IMPLEMENTATION OF EDI

1. Members Involved In EDI Implementation

EDI's development and installation will first involve at least a steering committee, including the following members. These members should report to the corporate executive.

a. Functional Business Managers

Functional business managers with responsibility for EDI feasibility studies within their own departments need to be involved in the implementation project. Those departments generally include purchasing, sales, accounting, inventory, etc.

b. Systems and Communications Manager

These people that assist in the technical system specification and the choice of EDI software and VANs.

c. Auditing and Legal Representatives

The implementation process needs to be monitored by auditing and legal representatives. All situations that occur during implementation should be reported and resolved.

d. Business Managers and Technical Support from Key Trading Partners

Since the feature of EDI is the interchange of information between trading partners, these partners should support the implementation project of the newcomers that includes input and output testing, information exchange, transaction set and implementation scheduling.

e. EDI Consultant

Consultants can provide experience and a broad perspective of EDI as a whole.

2. Procedures of EDI Implementation

Implementation is the aspect of the EDI project that requires an information system professional. Implementation should be done in phases. One must pay special attention to security and data recovery. The basic phases include implementing a pilot program, running the pilot in parallel with existing systems, bringing up the full-scale implementation and maintaining the EDI system.

a. Understand EDI and Develop a Strategic Plan

Start planning by developing a complete understanding of EDI and its implications. The planning phase should involve all departments whose basic functions will be impacted by the EDI system. Representatives from order sales administration, product distribution, purchasing, sales and marketing and any other pertinent department should be involved in the project from the beginning.

(1) *Analyze EDI's Opportunity*. The first thing to do is to decide whether to proceed. This can be done by a survey of system needs. The survey includes the volume of paperwork transaction documents that is handled manually, the number of suppliers, the length of the internal administration lead-time associated with the purchasing cycle, the need to develop headcount reductions, the need to increase professionalization of

purchasing personnel and the requisite from trading partners. These factors will be weighted by a key personnel from top management, e.g., accounting and finance, receiving, auditing, legal, purchasing and other related departments.

(2) *Develop Cost/Benefit Analysis*. The cost/benefit analysis will involve a before/after comparison of the effect of converting paper documents to electronic transactions. Factors such as reduction in cost per purchase order, integration of supplies with manufacturing, inventory cycle time reduction, positive integration of buyer/seller, efforts to improve overall productivity should be analyzed.

(3) *Develop Strategic Plan*. After analyzing the above factors, the project manager must make the decision to proceed. If the benefits are perceived to be low, then the effort should stop. If the perceived benefits appear to outweigh the associated costs, the work effort should continue. Also, the top management should to start a strategic plan for the whole project.

b. Form Agreements with Trading Partners

Agreements with trading partners are essential. It doesn't matter how elegant an EDI system is if no trading partners can access it. Assess the potential trading partners already have, whether they have pilot projects in the works, and whether they have requirements that will limit their own EDI implementations.

c. Redesign Business Systems

For the EDI system to be implemented successfully a complete understanding and mapping the information flow through the business must be a required. The information flow is then restructured to reap the benefits that automation and EDI offer. One must map data such as sales and marketing, inventory and general ledger information. Specify exactly which data will exist in which formats on which computer systems. Also, it will be necessary to map into automated data flows the paper systems into automated data flows that generate and use this information. Then redesign these flows with EDI in mind.

d. Develop Legal Considerations

Since a written signature contract will not accompany an EDI transmission, a separate contract governing the terms and conditions associated with doing business by EDI systems needs to be established.

e. Develop Auditing Considerations

Since EDI maintains data electronically, specific practices and techniques will have designed for in the audit process. These procedures will be somewhat differ from manual practices. The audit should be clearly established and an auditor should participate in the design and development of EDI applications.

f. Develop System Application Capability

It is important that a prototype EDI application be developed and tested. Data mapping must be built from the current system to ANSI ASC X12, with a translation protocol established. First, an interface with the existing system is done through a file extraction. Second, software is established for the translation. This will enable the purchaser and the supplier to understand how the EDI system operates. A prototype is useful. It should be a much smaller application than a full pilot test and should involve parties from both the buying and the selling firm.

g. Establish a Pilot Program

Before moving into full-scale EDI production, one can try it out on a small scale partner base. This is a good time to perform capacity studies, cost saving evaluations and trial audits. If there are trading partners that are already using EDI, those partners are good candidates for participation in the pilot program. If firms have multiple internal enterprises that do business with one another, these are also good candidates for pilot program partners.

h. Review Results of the Pilot Program and Modify

Once the pilot program developed and running, it should be run in parallel with existing paper systems. This phase is critical to work out the bugs in the EDI system while, maintaining business activities. When all these pilot system is working, a review of results, a record of errors, and a modification of the system may be made from comparisons with the old paper system.

i. Reevaluate Benefits and Costs

The EDI effort should be routinely monitored to review progress and continually improve its use. Problems need to be quickly identified so that solutions maybe developed to minimize any negative reactions. The EDI project manager and the implementation committee should have centralized responsibility to make changes in the EDI system when necessary. This committee should measure error rates from the system and prompt acceptance of the system. The committee should also be concerned with an audit function. They should document the apparent financial and nonfinancial costs and benefits associated with EDI. The reevaluation report will be delivered to top management as the reference of the next step. [Ref. 9]

J. JOBS OF EDI VENDORS

1. Implementation

Vendors need to help coordinate EDI program implementation. The company will help assess users' needs, identify and develop the necessary transactions, install mainframe software, and assist in integrating users' current applications with the EDI system. Vendors should communicate with users' trading partners and coordinate their installation and implementation schedules. They will also provide ongoing follow-up and support.

2. Installation

Vendors have to offer "hot-site" technical support for its software installation. If required, on-site installation support must be available.

3. Training

Vendors need to regularly offer training for EDI software users. These training courses must be available at the vendors' place or the users' facility.

4. Technical Support

Vendors should have the abilities to provide "hot-line" technical support for its users, such as a question deck service or on-line problem solving system. Since the EDI systems will involve more than a single software package, integration and interoperability techniques should be provided by the vendor.

5. Hub/spoke Testing

Vendors should provide a variety of services for EDI implementation. These include coordinating and testing trading partners' EDI implementations to ensure that documents and transactions are performing as planned.

6. Network Connection Services

Vendors need to coordinate the connection of a user's trading partners to the network of the user's choice. In addition, vendors act as the agent for several of the major VAN's, providing a turnkey EDI solution for hubs and spokes.

7. Serving Beta-site Report

It could be possible that vendors do not have the practical installation experience with the precise configuration that your organization requires. The organization may request to serve as a beta-site for the vendor. A beta-site user tests the new product in a real-world environment. In this situation many bugs may be discovered and fixed by working with the vendor.

8. Maintenance

Vendors should provide software updates to users regularly, incorporating new features and capabilities, as well as updates and changes of the EDI standards. Maintenance

of the EDI system requires dedicated IS resources. The paper maintenance once handled by clerks, such as maintaining lists of suppliers, managing databases of customer addresses, or ordering various forms, will shift to electronic maintenance done by IS personnel. Maintenance also means a continuing responsibility for monitoring and enhancements of the system. New partners will be brought on board periodically, different forms may need to be issued or received, with software upgrades and installation enhancements added as experience is gained.

K. AUDITING OF EDI

A company's use of EDI will have a profound effect on auditing activities as follows.

1. Payment Validation

One aspect of EDI is the validation of payment. This involves the reconciliation of the invoice with the purchase order and with the receiving documents to ensure correct payment. All of these documents are now computerized. Time and money will be saved compared to the manual process.

2. Audit Trail of Activities

Written signatures are replaced with codes and IDs. A date/time stamp should be used to keep track of all activities. There should be a requirement to identify each user to track the point of access and the data flow within the company. Also, the recording of authorization is critical to the EDI system. All of those can be achieved by the technology of electronic authentication and verification, such as a digital signature scheme. Lastly, a specific audit trail database is necessary.

3. Order and Payment Control

This activity requires security mechanisms to provide safeguards of the system to ensure only authorized sources can place orders and initiate payments. After the implementation of EDI, there will be less human involvement in the procedure and only

operators will deal with it (which means less control by authorization.) The training of these operators should be emphasized.

4. Accounting and Transaction Correspondence

Comparison of accounting record with actual transactions can insure that internal company data reflects actual inventory and dollar figures. All files are computerized, and there is no need for paper backup to verify records. So, spot checks of actual transactions with system files and verification of assets with different non-EDI data are very important. To assist in the auditing process, the company may generate accounting reports from EDI data within the company and from all their trading partners' records.

L. BARRIERS TO EDI

When the concept of EDI is introduced into corporations, several barriers may occur. When an organization decides to implement EDI, fundamental changes occurs in the way of doing business. Not all of those changes will be accepted by the organizations. Some become the barriers to the willingness to implement EDI. The largest obstacle to successful EDI implementation is the "distribution of EDI information" and "education," according to 60% of large-scale EDI users surveyed. [Ref. 10]

1. Distribution of EDI Information

Many potential trading partners in small and midsize firms are still unaware of EDI, despite years of effort to spread the news by private educational firms, EDI standards organizations and industry groups. EDI VANs are increasingly supporting this need by running implementation training sessions and expanding user meetings to focus on trading partner support.

2. Education and Training

The need for education is also important as EDI expands into new areas and as new industry groups implement EDI standards for communication among their members.

EDI education and training is a problem of many firms. Relying on vendors' training is not enough for the future development of EDI.

3. Adding Partners

The first and most pressing problem for hub implementations is the cost of adding trading partners. One way to keep costs low is to work closely with a VAN to get trading partners up and running. Most VANs offer trading partner support through optional consulting services and "hub" programs that identify, train and install EDI technology for smaller spoke partners of a VAN customer.

4. Communication Costs on the Rise

EDI hub companies in particular have to deal with the fact that as their EDI volume grows, communication costs rise. As total VAN communication costs increase, the likelihood exist that EDI hubs will move their EDI transmissions onto a dedicated EDI platform, bypassing the VANs. In essence, the hub can set up its own VAN and communicate directly with its many trading partners. The major VANs are sure to counter this trend with even lower communications costs for their largest customers. Some accept bulk transfers of messages sent during off hours and then sort and send messages later. Regardless of the eventual configuration of the VAN market, it is clear that EDI users can expect total EDI communication costs to keep rising, as partners are added. However, per-transaction costs will decrease over time.

5. Language Difference

International trading is becoming more common and the current restriction that EDI be transmitted in English seems questionable. Language translation systems, which can translate different languages into a unified code, seems to be a good solution in the future.

6. Applicability

EDI is best suited for high-volume, highly rigid order handling. It can't cope well with one-time, sporadic orders and customizing.

7. Contractual Problems

EDI doesn't contain with delivery arrangements, insuring of transported goods, tax issues, and other business aspects that are considered "contractual" between trading partners. [Ref. 11]

8. High Integration Cost

The integration of EDI cost is high. Many firms are using applications developed long before the advent of EDI; the cost of modifying or in some case, replacing these applications may be significant.

9. Various Proprietary Implementations

Different proprietary implementations are provided by different vendors. This will make it difficult for participants to decide which one to choose. This is especially true when different trading partners are using different proprietary implementations.

10. Implement EDI Means Starting Over

The use of EDI with new participants is a new phenomenon. All the processing procedures will have to be modified by the new technology.

11. Lack of Security

The security of E-mail is not well understood, and the security of the receiving documents is not easy to authenticate. This security issue is not so much that EDI documents will be forged. It also includes the security of the platform (network) that is used, the security management to be built inside the organization, and the security agreement between both sides of the trading partners.

12. Legal Position is not Clear

Fraud statutes mandate that certain transactions be signed and in writing. These requirements are unclear in the context of EDI. How do users keep adequate legal records of transactions when electronic records may be easily altered by those with adequate knowledge?

13. Lack of Integration

Most mainframe, midrange or front-end software packages offer some kind of mapping utility, including creation of special maps, ability to divert errors or exceptions for special processing, ability to enter bar-code data, and the ability to move any portion of data to or from EDI transaction sets to or from any portion of the application record. For any set of companies not using an entire set of sequences, there are still manual points where data has to be reentered. That interrupts a smooth, seamless data flow. This type of integration is the most complicated and sophisticated part of an EDI implementation. However, this will become the most beneficial aspect of EDI as a whole.

14. Dual System Problem

It is not easy for each company implementing an EDI pilot system to keep on doing the paper work. The benefits of EDI will not be realized if only a small fraction of trading partners participate and the rest continue to use a paper-based system. Even if only one trading partner continues to use a manual system, another partner will have to keep two systems to keep the business.

M. SUCCESS FACTORS OF EDI

According to a recent report by Market Intelligence, an international research firm specializing in telecommunication and information technologies, worldwide EDI revenues will grow eightfold, from $250 million in 1991 to nearly $2 billion in 1998 [Ref. 11].

There are four factors that make EDI a success.

1. Business Process Engineering

When businesses install EDI, they must face the problem of whether their business process is healthy or not. The approach should be a methodical, often structured, approach to initiating fundamental organizational changes by means of critically evaluating business activities. This involves the discarding or reworking of ineffective or inessential processes. New approaches are built that utilize new technology, to achieve major breakthroughs in improved business performance. This engineering work may allow businesses to rethink the cycle of the business process, to standardize and rationalize the processes.

2. Standardization and Infrastructure

As mentioned in the "standard of EDI" and the "platform of EDI", standardization and infrastructure make users more efficient.

3. Automation and Flexible Computing

The broad family of EDI software fits comfortably with a range of computer platform architectures, operating on mainframes, minicomputers and desktop (primarily DOS) machines. It may be used with both proprietary and UNIX environments. UNIX solutions have been emerging during the last two years, as EDI translation software for standardized transaction sets is critically important. As the mission-critical nature of EDI evolves, organizations are making decisions on the selection of EDI platforms, also including fault-tolerant machines.

There are several differences between the hub and the spoke. First of all, the two are using different computing platforms. Large companies will operate in mainframe environments with software purchased from independent software vendors. The spoke firms, sometimes small companies or companies enforced by their trading partners, may be operating on PCs or minicomputers and choosing software as compatible as possible with the requirements of hubs. The small firms will follow the EDI standards that have been dictated by the large hub firms, predominantly based on ANSI ASC X12.

The VANs have complex software requirements for message and transaction management services. These services are required to maintain the integrity of the envelope and the transaction set. This integrity may have to be maintained across more than one VAN.

4. Integration and Interoperability

It is very important to the companies who apply EDI to manage application integration and interoperability to achieve success. The integration and interoperability areas have organizational, platform and software components associated with them. The organizational issues are very complex and largely applicable to business process reengineering. Standards, and communications issues are mainly software issues.

III. FUNDAMENTALS OF DIGITAL SIGNATURE

A. INTRODUCTION

A digital signature (DS) is a digital signal, represented as a string of bits that attests to the authenticity of the document to which it is attached. Like handwritten signatures, it tends to be unique and easily applied to a specific sender. The first description of the concept of DS was in Diffe and Hellman's paper, "Multiuser cryptographic techniques", 1976. This was further discussed in Rivest, Shamir and Adleman (RSA) in 1978, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems" [Ref. 12]. This became known as the RSA algorithm. In 1992, the National Institute of Standards and Technology (NIST) proposed the Digital Signature Standard (DSS) based on the El Gamal's signature scheme [Ref. 13].

A digital signature is a message-dependent bit-pattern, and the signing process transforms the message into the signed message (or signature). This transformation, called the digital signature scheme, will be discussed in a later section. In general, the receiver can verify the DS as an authentication of the sender. The essential properties of the digital signature are that it is easy to produce, easy to authenticate, but difficult to forge.

DS is an important area of study because the lack of secure authentication has been a major obstacle of document digitalization. DS makes it possible to convert existing paper documents into an electronic form (like EDI).

B. LEGAL STATUS OF DIGITAL SIGNATURE

1. Introduction

The purpose of DS is to replace handwritten signatures. This means that DS must approach or equal the legality of hand written signatures. Currently, if two companies wish to digitally sign a series of contracts, it is recommended that they first sign a paper contract in which they agree in the future to be bound by any contracts digitally signed by them using a given signature method and minimum key size. NIST has stated that its proposed

Digital Signature Standard (DSS) should be capable of "proving to a third party that the data was actually signed by the generator of the signature."

2. Law Related to Digital Signatures

The legal significance of signatures and the use of writings bearing such signatures must be viewed from a perspective which encompass several branches of the law, including but not limited to the statute of Frauds, the Law of Acknowledgments, the Law of Agency, and the Uniform Commercial Code (UCC). The need for a device or process which satisfies the requirement of such branches of law in the context of electronic signatures can be appreciated by understanding some key areas of the law affecting hand-written signatures. The statute of Frauds began in 1677 and was enacted in England. It was designed to prevent fraud by excluding from consideration by the courts legal actions on certain contracts, unless there was written evidence of the agreement signed by the party to be charged or his duly authorized agent. [Ref. 14]

a. Law of Acknowledgments

Certain documents require acknowledgment or proof of the identity of the person who signs the document, and proof that it was signed on the stated date. This acknowledgment or proof is necessary to prevent the person who signed the document from claiming later that the signature is not genuine. Moreover, certain transactions require that the signature be witnessed by one or more persons. Such transactions may vary according to the law of the jurisdiction in which the document was executed.

Acknowledgment or proof of signature upon a legal document or instrument may normally be made before a judge, an official examiner or title, an official referee, or a notary public. Essentially, the form of an acknowledgment consists of the date the document was signed, names or parties involved, and notary public. Such acknowledgment together with the signed document are usually recorded in an official registry, like an office of the county clerk or secretary of state.

b. Law of Agency

The principles of agency law are essential for the conduct of business transactions. A corporation, as a legal entity, can function only through its agents. The law of partnership is to a large degree a special application of agency principles to that particular form of business organization. Agency is the fiduciary relation (involving a confidence or trust) which results from the manifestation of consent by one person to another that the other shall act on his behalf and subject to his control, and consent by the other so to act.

As a general rule, no particular formalities are required to create an agency relationship. The appointment may be either written or oral, and the relationship may be either expressed or implied. There are two situations in which formalities are required: (1) with a power of attorney, where a formally acknowledged instrument is used for conferring authority upon the agent; and (2) in a few states where it is required that the act which confers authority to perform a certain act must possess the same formalities as the act to be performed.

Generally, the law of agency applies to contracts or commercial paper. A principal (the person from whom an agent's authority derives) is bound by the duly authorized acts of his agent. If the agent does not possess the requisite authority, the principal in the most instances will not be bound. An agent who fails to bind his principal to an agreement because of the agent's failure to name the principal, or due to lack of the agent's authority, will usually be personally liable to third parties. Thus, the correct way for an agent to execute a contract or instrument is to affix the name of his principal followed by his own signature.

c. Uniform Commercial Code

The Uniform Commercial Code (UCC) is a comprehensive modernization and compilation of the various statutes relating to commercial transactions. Its primary objectives is to provide uniformity of commercial law throughout American jurisdictions.

The present articles relating to commercial paper, banking transactions, and investment securities are paper-based.

To accommodate electronic funds transfer systems, a special committee was formed to prepare amendments or supplements to these articles. Although the principles governing the transfer of paper-based stocks and bonds can generally be made applicable to the paperless variety, many technical and mechanical changes are needed to apply those principles to securities without certificates. According to the UCC,

> "Signed" includes any symbol executed or adopted by a party with present intention to authenticate a writing. [UCC:Sec.21-201(39)]

and, in case of commercial paper,

> A signature is made by use of any name, including any trade or assumed name, upon an instrument, or by any word or mark used in lieu of a written signature. [UCC:Sec.3-401(2)]

The inclusion of the word authenticate in the definition of signed clearly indicates that a complete handwritten signature is not necessary. This authentication may be printed, typed, stamped, or written; it may be initials or thumbprint. It may be on any part of the document, and in certain cases may be found in a billhead or letterhead. No catalog of possible authentication can be complete, and courts must use common sense and commercial experience in passing upon such matters. The question is always whether the symbol was executed or adopted by the party with the intention at that time of authenticating the writing.

A signature may be made by an agent or other representative, and his authority to make such a signature may be established according to the Law of Agency. No particular form of appointment is necessary to establish such authority. Such signature may be unauthorized if made by an agent who exceeds his actual, or apparent authority. An unauthorized signature is one made without actual, implied, or apparent authority, and includes those made by forgers, impostors, and fictitious payees.

The law of commercial paper also recognized the principle that the drawer, the one who creates a negotiable instrument, has voluntarily entered into relationships beyond his control with subsequent holders of the instrument. The law imposes on the drawer the responsibility of material alteration of the instrument later in the chain of transfer.

3. Legalizing Digital Signatures

With the lack of an overall statute governing paperless commercial transactions via electronic communications network, parties are free to enter into their own agreements. If disagreements arise later, the party seeking to enforce the contract will prevail only if the agreement complied with certain basic legal requirements. These requirements may include the provisions of statutes of frauds, acknowledgment, recording, and reasonableness. Modern statutes of frauds require some writing which indicates that a contract for sale has been made between the parties at a defined price, that it reasonably defines the subject matter, and that it is signed by either the party against whom enforcement is sought or by his duly authorized agent. Then, the digital signature can be legally accepted without doubt. If the initial written agreement defines the procedures and protocols of the utilization of the digital signature by both participants, then the problems of legality of digital signature will be solved. The initial written agreement will be discussed in the next session.

4. Initial Agreement and Legality of Signatures

Before we use digital signatures to generate signatures for legal usage, we have to make hand written signature agreements that govern their use.

In any case, the protocol for resolving disputes must be defined and agreed upon by both parties in advance. Different protocols determine different signature schemes even though the signing and verifying transformation may be the same. Although, both the laws and legal precedent regarding signed paper documents are well established, the situation regarding digital signatures is not as clear. It seems that, at present, if digital signatures are to have any standing in law, then an initial written agreement between the parties concerned

is required. This agreement should define the procedures for obtaining a digital signature, how it would be recorded, what each party's commitment would be and how disputes would be resolved. If one party to this agreement felt that another party had not honored his commitment then he could test this in court.

The legal problems are not as simple as mentioned above. There will be many of them that need to be addressed, such as the means of authenticating the signature, the obligation of a user to ensure the security of his signing transformation, the means of resolving disputes, penalties for the misuse of digital signatures, the use of stolen keys, signing false statements or misdating signed messages, and the action required in case of compromised keys. It will take a long time for adequate legislation to be introduced.

C. METHODS FOR DIGITAL SIGNATURE

1. Arbitrated Signature

Signed messages can be sent only by a trusted third party called the arbitrator. The recipient may be unable to verify the sender's signature directly, but is assured of its validity through the mediation of the arbitrator. The arbitrator is trusted by both parties and can give the confidence to the sender and recipient that the sender's signatures will not be forged and the recipient's received signatures are valid. In the real world, an arbitrator may be a banker or a lawyer. In computing environments, it could be a person, program or machine that manages application programs running on machine of the network.

2. Universal Signature

A universal signature (or so called true signature) is one in which the signer sends the signature directly to the receiver who is able to verify the signature without recourse to a third party. In this case, the signature acts as a broadcast cipher and all users can verify the origin of the message. However, only the sender can generate the signature from the original message. This is defined as a one-way authentication in Diffie and Hellman's paper [Ref. 15]. In arbitrated signature schemes, the receiver needs the cooperation of the arbitrator to perform verification of the signature. The true signature scheme relies on the

use of a special type of signature generation and verification called a one-way function. The fundamental property of this function is that it is easy to implement but hard to invert. One particular function is the trapdoor one-way function. It is a one-way function that allows someone in possession of specific secret information (the trapdoor) to compute an inverse. Several digital signature schemes based on one-way function and their trapdoors have been published. The concept of public key cryptosystem is based on a trapdoor one-way function.

Usually, the choice of digital signature schemes should be included within the initial written agreement or recorded in a designated registry that is available to the participants in advance. But it may also involve certain secret and/or nonsecret information provided to the receiver at the time the message is validated. [Ref. 16]

D. GENERAL SETTINGS OF DIGITAL SIGNATURE

1. Symmetric Key Cryptography

In a symmetric cryptosystem (sometimes called private key, single key, secret key, or conventional cryptosystem), both the encryption transformation and decryption transformation are determined by the same key. The secrecy of the key guarantees the authenticity of the message. The security of the encryption method is completely dependent on how well the key is protected. The Data Encryption Standard algorithm (called DEA by ANSI in standard X3.92 and DES in Federal Information Processing Standard 46), is a private key algorithm. The main problem is getting the sender and receiver to agree on the secret key without anyone else finding out. If they are in separate physical locations, they must trust a courier, or a phone system, or some other transmission system to not disclose the secret key they are communicating. Anyone who overhears or intercepts the key in transit can later read all messages encrypted using that key. This secret key cryptosystem process makes it difficult to ensure secure key management.

2. Public Key Cryptography

As stated in the above section the public key cryptosystem was invented in 1976 by Whitfield Diffie and Martin Hellman. It is based on the trapdoor one-way function. This cryptosystem may be used to solve the key management problem. But, this very first public key algorithm cannot be used for encrypting messages by itself, but is used to exchange keys to be used with other cryptosystems, and also for identification and related purposes. The security of the Diffie-Hellman public key system is based on the mathematical difficulty of the discrete logarithm problem. For this scheme to be secure, the keys should be long.

In the public key system, the key comes in inverse pairs. A message that is encrypted by one key can be decrypted by the other, but the possession of the public key does not allow the discovery of the other key. The separation of encryption and decryption makes it possible to display the member's public key in public and to achieve communication by the simple protocols:

a. Anybody using the public key (found in the public directory) can send a private message to the one who holds the encryption of the public key, but only that person can decrypt the message.

b. Any person can sign a message with his own secret key and everybody can authenticate that message by the corresponding public key. But, nobody can forge that message without the secret key (thus, the sender cannot deny he sent it.)

Protocol a. is the application of privacy or encryption and protocol b. could be used as a digital signature for authentication. This is a tremendous help for conventional end-to-end security in large communication networks. Thus, this cryptosystem can instill trust in a business telecommunication network. The availability of a signature that the receiver of a message cannot forge and the sender cannot disavow makes it possible to have negotiations and transactions. [Ref. 17]

Increased security is the primary advantage of a public key cryptosystem. The secret keys do not need to be transmitted or communicated to anyone. The primary

disadvantage is that speed of generating a pair of public keys and secret keys is slow, compared to the conventional secret key system.

3. Scheme with Probabilistic Verification

In arbitrated signatures, the signer shares his secret signing information with the arbitrator. This allows the arbitrator to verify a signature and pass on this assurance to the recipient. After signing the message, the signer reveals only part of his secret signing information (one of the subsets) to the recipient. This allows him to have confidence in the signature. The verification of the signature can be made, only when the signer reveals his secret signing information to arbitrator. The receiver's verification is a probabilistic verification. To validate this signature, the recipient randomly chooses a subset of the whole signing information and asks the signer to reveal the same subset. Then, the recipient verifies the revealed subset of the signing information by comparison. If and only if there is an exact match does the recipient accept the signature.

Any signature scheme of this type must be a one-time signature scheme which means the signer's secret information can be used to sign only one message. Thus, the scheme must have a registry of validating information used in the verifying process to validate the signer's revealed secret signing information.

E. IMPLEMENTATIONS OF DIGITAL SIGNATURE SCHEMES

1. El Gamal's Signature Scheme

El Gamal [Ref. 18] has proposed a public key cryptosystem and a digital signature scheme based on discrete logarithms. This scheme has the advantage that the public enciphering key and the public information used in the verifying transformation are the same. El Gamal's scheme was the basis for several signature methods that followed, including the one by Schnorr [Ref. 19], which because the basis for the Digital Signature Standard (DSS) proposed by NIST.

2. The Fiat-Shamir Signature Scheme

This scheme is based on interactive zero-knowledge protocols, but may be adapted for signatures. It is faster than RSA and is probably equivalent to factoring, but the signatures are much larger than RSA signatures.

3. The Goldwasser-Micali-Rivest Signature Scheme

Papers by Goldwasser and Micali and by Blum and Goldwasser introduced the probabilistic encryption method [Ref. 20]. Probabilistic encryption has the attraction of being resistant to a guessed ciphertext attack, but at a cost of data expansion. In this method, an attacker sees a ciphertext, guesses that the message might be "Attack at dawn", and encrypts his guess with the public key of the recipient. By comparison with the actual ciphertext, he will know if he was correct. This attack can be thwarted by appending some random bits to the message. At the same time, Goldwasser, Micali and Rivest define a specially designed permutation method and signature scheme that relies on the fact that squaring modulo n (where $n = pq$ with p and q secret large primes) is a trapdoor one-way function. The signature scheme of Goldwasser, Micali and Rivest show a tree structure to generate authenticated verification parameters. Different values for the verification parameters may be used for different messages. These various authenticated verification parameters can sign messages, and later on verify them.

4. Digital Signature Using Rabin's Public Key Cryptosystem

Rabin proposed a system based on a one-way function that also uses the product of two large primes which is probably equivalent to factoring. This system has an advantage over RSA, where one may still have a lingering worry about an attack unrelated to factoring. Rabin's method is suspectable to an active chosen message attack, in which the attacker tricks a signer into signing a certain message.

Rabin's public key cryptosystem is not immediately acceptable as a digital signature scheme, since it is not possible to sign all the message by its decryption secret key after enciphering those messages. [Ref. 21] However, it is possible to combine this

cryptosystem with a suitably chosen hashing function to provide digital signatures. This will be explained in the next section.

5. Digital Signature Using RSA

RSA is a public key cryptosystem for both encryption and authentication. It was invented in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman, all professors of MIT. RSA is part of many official standards worldwide. The OSI 9796 standard lists RSA as an acceptable cryptographic algorithm, as does the CCITT X.509 security standard.

RSA is part of both the Society for Worldwide Interbank Financial Telecommunication (SWIFT) standard and the French financial industry's ETEBAC 5 standard. Also, the Australian Digital Signature Standard specifies RSA.

A major advantage of RSA over other public key cryptosystems is that it can be used for both encryption and authentication. Though RSA was developed 15 years ago and several signature schemes have been proposed after it, none have been successfully broken by any attacker [Ref. 22].

F. HASH FUNCTION

1. Introduction

A hash function is a computation that takes a variable size input and returns a string of fixed size, called the hash value. If the hash function is hard to invert, it is also called a message-digest function, and the result is called message digest. The name hash function has many synonyms, e.g., compression function, contraction function, message digest, fingerprint and cryptographic checksums [Ref. 23].

Digital signatures and hash functions can provide authentication and verification of message integrity at the same time. Another reason to combine hash functions with digital signatures is that the use of public key systems to generate signatures is much slower than conventional ciphers, such as Data Encryption Standard (DES). Other digital signature schemes are also relatively slow. Furthermore, some schemes produce signatures comparable in size to, and in some cases larger than, the message they sign. This results in

data expansion and an effectively lower bandwidth of transmission. Thus, it is usually not desirable to apply a digital signature directly to a long message. Probabilistic cryptographic algorithms (as mentioned in the digital signature setting), such as DES or the public-domain MD5, may be used as a one-way hash function to facilitate pattern matching of numerous huge data strings or files.

There are some problems with hash function. A collision could occur when two distinct messages are hashed into the same value. A proposed hash function must create message digests at a minimum length by padding in order to prevent attacks based on exhaustive search. For example, if a hash function produces 100-bit numbers, an exhaustive search would take 2^{100} attempts on the average to match a given value, and approximately 2^{50} attempts on an average to find two inputs producing the same digest [Ref. 24]. However, just being long enough doesn't guarantee the security of a hash function. The use of timestamp or sequence numbers in messages may prevent this problem.

2. Implementations of Hash Function

a. Message Digest (MD)

As mentioned above, the MD is the result of a hash function. Examples of MD are public-domain MD2, MD4 and MD5. Those are widely used hash functions designed by Ron Rivest specifically for cryptographic applications. The algorithm produces 128-bit "fingerprints" or MDs. It is conjectured that it is computationally infeasible to produce two messages having the same MD, or to produce any message having a given perspective target message digest. The MD5 algorithm, as a case, is intended for digital signature applications where a large file must be compressed in a secure manner before being encrypted with a private (secret) key under a public key cryptosystem. [Ref. 25]

Apart from their use with DS, cryptographic hash functions can provide origin verification and integrity protection for the message by appending a message digest prior to encryption. Generalizing this application to the case where a message is to be sent

to more than one recipient, much cryptoprocessing can be saved by computing a single MD and then encrypting it differently for each intended recipient.

b. Secure Hash Standard

The Secure Hash Standard (SHS) is a hash function proposed by NIST. It is designed for use with its proposed DSS. It produces a 160-bit hash value from a variable size input. SHS is structurally similar to MD4 and MD5. It is 25% slower than MD5 but may be more secure, because it produces message digests that are 25% longer than the MD functions [Ref. 26].

c. CCITT X.509 Recommended Hash Functions

CCITT X.509 recommended the requirements that hash function must be one-way and collision-free. Also, X.509 recommended the signing of a digital signature on the information summary after the information summary is produced by one-way hash function.

G. APPLICATIONS

1. Public Key Certification

Public key cryptosystems have advantages over conventional cryptosystems when used for key management, especially when network communication is used. In this system, every user's public key is signed by a certification authority (CA). Then, this key is stored in the public key list. The CA must be trusted, and its public verification transformation must be known to all users of the system with a need to know. The key together with the signature is then usually referred to as a certificate for that user.

In practice, it is necessary for the certificate to contain the name of the key owner, as well as the public key itself. In addition, the certificate may also contain an expiration date and/or an identifier for the algorithm with which the public key is to be used. The importance of the use of a one-way hash function is to prevent manipulation of the signed data and the certificates.

Certification of public keys is one of the most confidential application areas for digital signatures. The 1988 version of the CCITT X.500 Directory Recommendations specifies how public key certificates may be stored in user directories, and CCITT X.509 provides a security framework for these directories. This application of digital signatures has also been adopted to provide key management for Internet electronic mail security.

CCITT Recommendation X.509 also makes the provision for the case where more than one CA is used. This is done by allowing CAs to produce certificates for each other's public verification transformations. Sequences of such cross-certificates can be used to enable a user to obtain an authenticated copy of a CA's public verification transformation, and to check a key certificate produced by that CA. However, CCITT X.509 does not mandate which digital signature should be used. It is recommended that the secret signature process should be identical to the decryption process for a public key cryptosystem.

2. Applications of RSA Cryptosystem

Rather than the old symmetric cryptosystems, the RSA system uses a matched pair of encryption and decryption keys instead of using the same key to both encrypt and decrypt the data. Each key performs a one-way transformation upon the data. Each key is the inverse function of the other; what one does only the other can undo. The RSA public key is made publicly available by its owner, while the RSA private key is kept secret. To send a private message, an author scrambles the plaintext with the intended recipient's public key. Once so encrypted, the message can only be decoded with the recipient's private key. Inversely, a user may also scramble data using his private key. In other words, RSA keys work in either direction. This provides the basis for the digital signature. If a message may be unscrambled with someone's public key, their private key must be used to scramble it in the first place. Since only the owner can utilize his own private key, the scrambled message becomes an electronic signature - a document that nobody else can produce.

The applications using RSA algorithm and other security mechanisms are provided as follows:

a. The RSA Digital Envelope

RSA also provides a new level of privacy, which is called the RSA Digital Envelope (Trademark by RSA). Secure communication is not possible without any previous relationship between parties. Electronic mail may be sealed in a cryptographic "envelope" so that only the intended recipient can read it, despite the use of public communication channels or storage. No sharing or distribution of the secret keys is necessary. The process is as simple as dialing a person's telephone number. Combining RSA and DES provides a good solution. RSA handles key management and authentication and DES is used for bulk data encryption. In practice, the RSA algorithm may be combined with any other traditional "symmetric-block" or shared-key cipher, of which DES is just one example. The message author generates a random DES key every time he/she writes a message. When the message is completed, it is encrypted using this DES random key with the recipient's public key, then sent. Only the intended recipient may use his/her private key to recover the DES key, and then use it to decode the message. [Ref. 27]

b. RSA Digital Certificate

The RSA Digital Certificate is a practical product currently being used by industry. It is actually a copy of the author's public key that has itself been "digital signed" by a mutually trusted authority, such as a network security director. Every time someone sends you a message, they attach their certificate. The recipient of the message first uses the certificate to verify that the author's public key is authentic, then uses that public key to verify the message itself. In this way, only the public key, of the certifying authority has to be centrally stored or widely publicized. Anyone else can simply transmit their public key and valid certificate with their messages.

Using RSA digital certificates, an authentication chain can be established that corresponds to the organizational hierarchy, allowing convenient public key registration and certification in a distributed environment. [Ref. 28]

c. The RSA Digital Signature and Message Digest

The RSA Digital Signature not only reveals the identity of the author, but verifies the contents of a document as well. Identification of the author is provided by a basic property of the RSA cryptosystem. That is, when one decodes a message from someone using his/her public key, the message must have come from him/her, because only the correct private key could produce a coded message that descrambles with the matching public key. The checking of someone's handwritten signature against a previously produced sample signature is not the same as producing that signature. Likewise, with RSA, the public key verifies the private key as the source but reveals next to nothing about the corresponding private key. No secrets have to be shared and two people who have never communicated before may exchange messages with confidence of the recipient's identity. Handwritten signatures identify the author of a document, and vouch for the document's integrity. The RSA Digital Signature employs a cryptographic "hashing" algorithm to create an MD that is unique to each document, much like a fingerprint. If even a single bit of the document is changed, roughly 50% of the bits in the corresponding MD will change. Furthermore, the hashing algorithm is a one-way function: the document content cannot be reconstructed from the bits of the MD. Scrambling this MD with the author's private key produces the RSA Digital Signature, which is then attached to the document.

MD4, the MD algorithm used in RSA products, was developed by Dr. Ronald Rivest. It is fast becoming the worldwide de facto standard. MD4 is the fastest, most secure algorithm of its type. With its 128 bit message digest, the probability that different documents will have the same digest by coincidence is less than 1 in a trillion, effectively ensuring that two MDs will only match if they are identical. If someone receives a file with an RSA Digital Signature attached, the signature is separated from the document. Then, the

recipient uses MD4 to compute an independent MD based on the document he received. The recipient's program then descrambles the signature block with the author's public key to obtain the original MD, and then compresses it. If the two MDs match, the signature is undoubtedly authentic and the document has not been altered since its signing, either by tampering, virus infection, or transmission errors. [Ref. 29]

d. Public Key Cryptosystem Standard (PKCS) by RSA.

The new Public-Key Cryptography Standards (PKCS) have a number of components, called PKCS #1, #3, #5, #6, #7, #8 and #9. It is proposed for incorporation in future OSI (Open System Interconnection) Standards. PKCS #1 describes a method for encrypting data using the RSA public-key cryptosystem, producing signed data by digital signature, and transmitting messages by digital envelopes. For digital signatures, the content to be signed is first reduced to a message digest algorithm (such as MD4,MD5), and then an 8-bit string containing the message digest is encrypted with the RSA private key of the signer of the content. PKCS #1 also defines two digital signature algorithms for use in signing *CCITT X.509/PEM certificates and certificate-revocation lists (CRLs), and other* objects employing digital signatures, such as CCITT X.400 message tokens.

3. Authentication Using Digital Signature

The traditional way for communicating parties to verify one another's identity in computer networks is by using passwords. One of the alternatives is the use of cryptographic authentication protocols, for which standards are now emerging. Some of the most important of these protocols are based on digital signatures. CCITT Recommendation X.509 specifies three different protocols for authentication, all based on the use of digital signatures. All of these protocols are based on the use of a cryptographic data structure called a token. Like a certificate, a token is merely a series of data items with a signature appended. In general, tokens are used when a single communication occurred between two parties.[Ref. 30]

4. Electronic Mail Security Based on Digital Signatures

The 1988 version of the CCITT X.400 series of recommendations support a variety of security services, based on the use of digital signatures and public key cryptosystems. Key management is based on the use of public key certificates, as specified in CCITT X.509 which had been mentioned in the "Public Key Certification" section above. End-to-end security services are almost all based on the use of a structure called message-tokens, whose general form is similar to that of the token described in Section 3 above. Tokens are also present in the authentication protocols used by pairs of communicating CCITT X.400 entities. [Ref. 31]

5. Certificate-based Systems Employ Digital Signature

Digital signatures may be employed for authentication in the process of distributing public components in public key systems. In particular, if the system is certificate-based, the central issuing authority can sign certificates containing public components.

The notion of a CA can be extended to a hierarchical structure or tree structure. The CA serves as the root of the tree; leaves represent users. Intermediate nodes will be arranged in several levels and also represent users. Each node of the tree is responsible for authenticating its children. Thus, an authentication path is an organization; the organization may certify a unit; the unit may certify an individual user. Certification may be accomplished by having a parent node sign a certificate for the child node. To validate another user's certificate, a user may request the entire authentication path.

It is also possible for the tree to be replaced by a forest. For example, in a multinational system there may be a different tree for each country. Here, the roots must certify each other. An authentication path may then pass through two or more roots. More generally, an authentication structure can be an arbitrary directed graph, with a directed edge from A to B if A certifies B. Then, authentication paths may be constructed by

conjoining directed paths from two users to a common trusted node. This application is similar to the use of public key certification. [Ref. 32]

6. Signatures by Tamper-Resistent Electronic seal

There is a separation between encryption and decryption in a public key cryptosystem that makes such systems candidates for digital signature schemes. Since each of the two functions has its own key, it is possible to verify a signature using encryption without being able to generate a signature which requires decryption. For conventional systems, the same separation of keys does not exist. If the encryption and decryption transformations were successfully separated, then it would be possible to provide signatures using such a system. One way of achieving this separation is by physical means using tamper-resistant modules (TRMs) by a Tamper Detection System (TDS). Electronic seals generate random numbers or record the date and the time whenever the system's logical seals are removed or containers are opened. The concept is that every access will cause a different random number to be displayed. Tampering can then be detected by comparing the current number with its value prior to shipping. The main weakness of this approach is that an adversary can simply remove the device and replace it with a similar one that displays the original number. The false device may even continue to generate random numbers with successive triggers to avoid drawing suspicion to itself. TDS provides a cost-effective, reusable seal that cannot be forged, bypassed, or replaced. In fact, even if an authorized person accesses the equipment, he/she will not be able to do so without leaving evidence. This system consists of a display module, an array of tamper sensors, a random number generator, an audit counter and a physical key. When any sensor is activated, the display module generates a random number, referred to as the audit counter. The audit counter is then encrypted and stored, and a sequence number is incremented. To later recover this information, a trusted individual inserts a Datakey [Datakey is a trademark of Datakey, Inc., Burnsville, MN] containing the corresponding decrypt key. The decrypt key is extracted, and the encrypted audit counter is decrypted and

displayed along with the sequence number. Verification that the sequence number and audit counter have not changed since they were last examined provides assurance that access had not been gained. The combined operations of generating a random number each time access takes place and limiting its recovery only to authorized personnel in possession of a uniquely keyed and physically separate device, results in an unforgeable seal. This provides its users with a very high level of confidence that the enclosure's integrity has not been breached. This is the first example where it is explicitly stated that the security relies on tamper resistance. It achieves this by using tamper-resistant devices and by storing all devices that holds the secret key in physically secure locations. [Ref. 33]

7. Resolution of Disputes

A digital signature with the generator's identification and timing data (like, timestamp) can retain evidence of the original message. It, therefore, finds an application if a dispute arises between the sender and receiver over a message. The digital signature may be presented as evidence to a referee, who can then settle the dispute. Signatures also find application where the sender and receiver are not in dispute, but the receiver may wish to save the message and use it at a later date with the assurance that it has not been modified in the interim.

8. A Secure Telephone system

Diffie describes a secure telephone system for use with Integrated Services Digital Network (ISDN) which relies on digital signatures. To make a secure telephone call, a user places a smart card in the telephone and dials the required number. When the receiving telephone answers, the two telephones perform a Diffe-Hellman exponential key exchange. This exchange provides the telephone with a shared key which is a combination of secret pieces of information chosen at random by the two telephones.

9. Digital Signature Standard (DSS)

This standard specifies a Digital Signature Algorithm (DSA) appropriate for applications requiring a digital rather than a written signature. The DSA is a pair of large numbers represented in a computer as strings of binary digits. The DS is a computing algorithm using a set of rules and a set of parameters enabling it to be used to verify the identity of the originator and the integrity of the data.

The DSA includes signature generation and verification. Generation makes use of a private key to generate a DS. Verification of the signature makes use of a public key that corresponds to, but is not the same as, the private key used to generate the signature. Each user possesses a private and public key pair. Public keys are assumed to be known to all members of a group of users (or to the public in general). Private keys must be known only by their creators. Anyone can verify the signature of a user by employing that user's public key. Signature generation may be performed only by the possessor of the user's private key.

A hash function is used in the signature generation process to obtain a condensed version of data, that is, a message digest. The message digest is then signed. The DS is sent to the intended recipient along with the signed data (often called a message). The recipient of the message and signature verifies the signature by using the sender's public key. The same hash function must also be used in the verification process. The hash function will be specified in a separate standard. Similar procedures may be used to generate and verify signatures for stored as well as transmitted data.

The DSA authenticates the integrity of the signed data and the identity of the signer. The DSA may also be used in proving to a third party that data was actually signed by the generator of the signature. The DSA is intended for use in E-mail, EFT, EDI, software distribution, data storage, and other applications which require data integrity assurance and data origin authentication. [Ref. 34]

10. Privacy Enhancement Mail (PEM)

The privacy enhancement for Internet E-mail proposed in the Network Working Group RFC 1421, RFC 1422, RFC 1423, and RFC 1424 is stated for use with security services. These security services include the algorithms of message encryption, message integrity check, symmetric key management, asymmetric key management, asymmetric encryption, and asymmetric signature. It is designed to be compatible with current Internet E-mail formats. It is an inclusive standard, and allows the use of both public-key and secret-key cryptosystems. Multiple cryptographic tools are supported; for each mail message, the specific hash function, encryption algorithm, signature algorithm, etc., are specified in the message header. PEM also supports the use of certificates, endorsing CCITT X.509 for certificate structure.

IV. SECURITY PROBLEMS OF ELECTRONIC DATA INTERCHANGE

This chapter classifies the following threats and vulnerabilities associated with EDI: 1. application program (data security), 2. platform (network security), 3. security management, and 4. interchange security. The relationship between these aspects and the DS will be discussed as well. The next chapter will discuss the details of security management.

A. INTRODUCTION

The international standard ISO 7498 part 2 (Basic Reference Model for Open System Interconnect: Security Architecture) suggests that the term "security" means minimizing the vulnerabilities of assets and resources. A threat can be any person, object, event, or idea that, if realized, could potentially cause damage to the EDI system. Threats can be malicious, such as the intentional modification of sensitive EDI documents, or can be accidental, such as the EDI format entry error of the operator. Vulnerabilities are weaknesses in an EDI system that may be exploited by a threat.

A security service is the collection of security mechanisms, procedures, etc., that are implemented on an EDI system to protect the EDI from threats. For instance, the identification and authentication service could be designed to help protect the EDI system from unauthorized EDI access by requiring that senders or recipients identify themselves through digital signatures. Security mechanisms are the controls implemented to provide the security services that are needed to protect the EDI system. For example, a token based authentication system (which requires that the user be in possession of a required token) may be the mechanism implemented to provide the authentication and identification service.

This chapter assesses the possible threats and vulnerabilities of EDI, and presents the required security services and mechanisms. Proper security tools for implementing EDI documents transmission will also be discussed. The only current practical method of protection for EDI documents transmission is through the use cryptographic techniques.

which have been implemented as interchange security tools. The interchange security of EDI documents transmission is discussed at the end of this chapter. The managerial security issues will be discussed in Chapter V.

B. THREATS ASSESSMENT OF EDI

This section reviews the eight major threats to EDI in an EDI environment. The primary vulnerabilities are related to those threats and will be discussed in the next section. The threats may occur in EDI systems, network system and message handling system including unauthorized access, inter-document threats, and in addition. Attacks on open system platform of EDI will address potential weakness and may comprise a number of threats.

1. Unauthorized Access

Unauthorized user access into an EDI system is a primary security threat. The system is vulnerable if it lacks identification and authentication facilities such as an access control list or access control matrix, password management or Trojan horse/back door avoidance programs. Sometimes, problems are caused by an unprotected modem, lack of entrance control facilities in the computer center, etc. Prevention of unauthorized access will greatly reduce the threats to EDI.

2. Inter-document Threats

The inter-document threats are also called message sequencing threats. This type of threat occurs when part or all of the message is repeated, time-shifted, or re-ordered to fool, confuse, or even violate the initial agreement. This can be used to exploit the authentication information in a valid message and resequence or time-shift valid messages. Inter-document threats arise from unauthorized agents who are external to the message communication, and can manifest themselves in the following ways:

a. Masquerade

A masquerade occurs when an imposter pretends to be an authorized user. Masquerade threats include the following:

(1) impersonation and misuse of the Message Transfer System (MTS);

(2) false acknowledge receipt;

(3) false claim to originate a message;

(4) impersonation of an Message Transfer Agent (MTA) to an MTS user;

(5) impersonation of an MTA to another MTA.

A masquerade usually consists of other forms of attack and in a secure system may involve authentication sequences from valid users, e.g., in replay or modification of messages. [Ref. 35]

b. Traffic Analysis

The observation of information between users can reveal to an eavesdropper how much data is being sent and how often. However, even the eavesdropper cannot determine the actual contents of the document formats, he can still deduce a certain amount of information from the rate of traffic flow. Under certain OSI layers, traffic analysis is not naturally restricted. Under a private VAN, this will be easier to prevent but may not be able to be eliminated.

c. Documents Manipulation or Modification

The replacement, insertion, deletion or misordering of the content of EDI documents may occur during transmission by unauthorized agent. The ease of this occurring is based on the level of cryptographic checksums.

3. Intra-document Threats

Intra-message threats are those performed by the actual message communication participants themselves, and can manifest themselves in the following ways:

a. Document Repudiation

This occurs when the sender or receiver denies that they sent or received the document. It is serious when the documents lack a digital signature or timestamp to prove who the sender is and when the document was signed. This is also very serious when the transfer of critical contracts or order forms is being made. Also, it will happen in CCITT X.435 EDI-MHS when an MTS user may later deny submitting, receiving, or originating a message.

b. Security Level Violation

If a management domain within EDI employs different security clearance levels (e.g., public, private, business, confidential, etc.), then, users must be prevented from sending or receiving any EDI to trading partner that have inadequate security clearance levels.

4. Denial of Service

Denial of service occurs when someone or something dominates the EDI system resources and tries to stop or slow down the system or EDI platform performance to achieve a delay. This may result in a denial of access, a denial of communications, a deliberate suppression of a message to a particular recipient, or a fabrication of extra traffic. In EDI-MHS, the MTS can be denied if an MTA fails to operate incorrectly. In addition, an MTS user may cause the MTS to deny a service to other users by flooding the service with EDI messages which might overload the switching capability of an MTA or fill up all available EDI message storage space.

Networks are also vulnerable to denial of service attacks. In attacks of this kind, an attacker prevents legitimate users from using the network. There are three common types of the network denial of service attacks: service overloading, message flooding, and signal grounding. Service overloading occurs when floods of network requests are made to a server daemon on a single computer. These requests may be initiated in a number of ways, many intentional. The result of these floods can cause machines to be so busy servicing

interrupt requests and network packets that it is unable to process local tasks in a timely fashion. Such attacks may also mask an attack on another machine by preventing audit records and remote login requests from being processed in a timely manner. Message flooding occurs when a user slows down the processing of a system on the network to prevent it from processing its normal workload by "flooding" the machine with network messages addressed to it. The flood of messages overwhelms the target so that most of its resources are being used to respond to the messages. In extreme cases, it will block the path of EDI transmission causing errors or lack of memory.

5. Data Interception or Taps

Data interception means the observation by an unauthorized user of communication by an authorized user. There are two basic types of taps:

a. Passive Taps

Passive taps threaten the secrecy of the information that is being transmitted. The intruder of this kind will attack through eavesdropping or monitoring the data but doesn't modify it.

b. Active Taps

Active taps threaten the authenticity of the information that is being transmitted. The intruder will threaten the transmission by changing the original document, misrouting it to another destination, replaying a previous document, or forging an acknowledgment of a genuine sending. [Ref. 36]

6. Data Store Threats

a. Mis-routing

The unauthorized modification of the directory's contents could lead to messages being mis-routed or ever lost, while unauthorized modification could mislead or confuse the intended recipient. This problem will naturally occur in Open System

Interconnection (OSI) layers 1 to 3 when using an open system as EDI platform, but will be less if using a private VAN.

b. Preplay

An unauthorized agent could make a copy of a deferred delivery document and send this copy to the intended recipient while the original was still being held for delivery. This could fool the document recipient into replying to the document originator before the originator was expecting a reply or could simply mislead or confuse the original intended document recipient. In an EDI system, these threats will cause dual ordering or document duplication.

7. Leakage of Information

Information may be acquired by an unauthorized party by monitoring transmissions, or by unauthorized access to information stored in any data storage entity. In some cases, the presence of an MTS user on the system may be sensitive and its anonymity may have to be preserved. An MTS user other than the intended recipient may obtain a message. This might result from impersonation and misuse of the MTS or by causing an MTA to operate incorrectly. Further details on the information flowing in an MTS may be obtained from observing the traffic.

8. Other Threats

In a multi- or single-level secure system, a number of threats may exist that relate to security labelling, e.g., routing through a node that cannot be trusted with information of a particular value, or where systems use different labelling policies. Threats may exist to the enforcement of a security policy based on logical separation using security labels. An MTS user may originate a message and assign it a label for which it is not cleared. An MTS user or MTA may set up or accept an association with a security context for which it does not have clearance.

C. VULNERABILITIES ASSESSMENT ON EDI

There are many points that vulnerabilities may occur when documents are being interchanged.

1. Media

Media, including telephone lines, the cabling, microwave, the radio transmission, and even the optical fiber, may be vulnerable. Different types of media will have different levels of vulnerability.

2. EDI Application Equipment

EDI application equipment such as the switching systems, signaling equipment, or testing equipment can be attacked. The access points of an EDI user interface on the communication lines are very vulnerable to attack. In addition preventing unauthorized access, one has to be concerned about the communication environment. For instance, the network itself should be tolerant to natural disasters and data error.

3. Security Management

Security management includes physical/logical security management, such as EDI system hardware management, EDI system software management, EDI system operation personnel management, EDI documents flow management, etc. Various management issues will cause different types of vulnerabilities, and will influence the organization as a whole.

4. The Weakest Point

It's often easier to break into a system over a network than it is onsite. Using a telephone or network to transmit EDI makes it easier for a cracker to break in midway.

D. SECURITY MECHANISMS

1. . Authentication Exchange

There are two grades of authentication:

a. Simple Authentication

This relies on the originator supplying its name and password, which are checked by the recipient.

b. Strong Authentication

Strong authentication relies on the use of cryptographic techniques to protect the exchange of validating information. This cryptosystem may be an asymmetric scheme or a symmetric scheme using a private key system or public key system. The authentication exchange mechanism is used to support the peer entity authentication service.

2. Encryption

Either symmetric or asymmetric cryptosystems may be used to encrypt data and generate digital signatures. In the case of EDI transmission, encryption is performed on an EDI formatted message before initiating interchange and decryption is performed after receipt. This service supports the data confidentiality and maintains the privacy of messages.

3. Data Integrity

This mechanism involves the encryption of a compressed string of the relevant data to be transferred. Together with plain data, this message is sent to the recipient. The recipient repeats the compression and subsequent encipherment of the plain data and compares the results with that created by the originator to prove integrity. This mechanism also involves the generation and verification of integrity checks which exist at the originating end and at the receiving end, respectively. Integrity mechanisms employ cryptographic techniques to produce integrity checksums which can be used to determine whether there has been any insertion, deletion or reordering of documents. This is done by using chaining techniques. For example, the Message Authentication Code (MAC) may be generated by the cipher block chaining (CBC) technique. The data integrity mechanism is combined with encipherment of the compressed plain data by either an asymmetric scheme

or a symmetric scheme. Both symmetric and asymmetric schemes are provided by the DS mechanism. The data integrity mechanism supports the data integrity service. It also partially supports the non-repudiation service and can be fully met by using the DS mechanism.

4. Non-repudiation

The common mechanism for providing non-repudiation relies on the use of digital signatures. Typically, the digital signature is calculated using the public key algorithm. A hash function is needed when it is necessary to sign long messages. After generation of this digital signature, it is attached to the document before being sent out.

5. Digital signature

This mechanism involves the encipherment by the originator's secret key of a compressed string of the relevant data to be transferred. The digital signature together with the plain data is sent to the recipient. Similar data integrity mechanisms, this message is processed by the recipient to prove integrity. The DS mechanism also proves the authenticity of the originator and the unambiguous relationship between the originator and the data that was transferred. The DS mechanism supports the data integrity service and also supports the non-repudiation service.

E. THE REQUIREMENTS OF SECURITY SERVICES

We now consider the security services required to preventing the threats that are mentioned above:

1. Identification and Authentication

This service provides a mechanism to verify whether the sender and recipients are who they claim to be. It also provides the confidence that, at the time of request, an entity is not attempting to masquerade or mount a replay attack. Two types of different peer entity authentication services follow.

a. Single Entity Authentication

In this case, identification and authentication is required by the person who want to access the system. This service is implemented by the mechanism of userids, passwords, tokens, etc.

b. Mutual Authentication

This method allows both the sender and recipient to authenticate each other.

When requesting a peer entity authentication service, the two users agree whether their identities will be protected or not. In a general situation, when two parties wish to authenticate each other, they may need to involve one or more third parties. The nature of trust between each party and the third party is an important issue in determining the assurance of the service. Examples of third parties include authentication servers, key management servers and certification authorities. This service may be used to protect against masquerade and replay.

2. Access Control and Authorization

This service provides the ability to limit and control the access right to host systems or the platform where EDI application software is located. Also, this service can protects the EDI information that is stored in the application program or database system. In an access control scheme, certain entities (initiators) attempt to access other entities (targets) for the purpose of getting information or modifying data accidentally or maliciously.

a. Decision Process of Access Control

Access control information used in the decision process includes the following:

- Individual identities of initiators and targets.
- Group identities of initiators and targets.

- Security labels of initiators (e.g., clearances) and targets (e.g., classifications).

- Roles (e.g., system administrator, manager) of initiators and targets.

- The actions or operations that can be allowed to perform on target.

- Other contextual information, which may include time periods, routing information, and location information.

b. Rules of Simple Access Control

The simple access control rules include:

- Check every access

- Allow least privilege

- Verify acceptable usage

This service is achieved through discretionary access control (DAC) or mandatory access control (MAC) using the facilities of an access control list, access control matrix or capability granting. The difference between discretionary access control and mandatory access control is that the former makes the security decision by users and the latter are driven by the results of a comparison between the user's trust level or clearance and the sensitivity designation of the information. This service may be used to protect against the unauthorized use of resources.

3. Data Confidentiality

This service is used to provide for protection of data from unauthorized disclosure and protects against data interception. The mechanisms that are typically used to provide confidentiality are based on cryptographic techniques. In an operating system environment, it may be sufficient to protect the confidentiality of data just by combination with access control mechanisms. In a network environment, link-to-link or end-to-end encryption can be provided. Examples include DES - Symmetric encryption, and RSA - public key encryption. Confidentiality attributes include secret keys, public, and private keys. The

process of data confidentiality may be initiated after the necessary keys for encipherment have been exchanged.

4. Data Integrity

This service provides proof of the integrity of data in the communication and can be used to detect and protect against the manipulation and modification of data.

5. Non-repudiation

This service provides proof of the integrity and origin or delivery of data, in an unforgeable relationship, which may be verified by any third party at any time. This protects the sender against the threat of false denial by the recipient and protects the recipient against the threat of false denial by the sender.

6. Auditing and Accountability

The security audit service is complementary to all the security services described above in that it is not directly involved in the prevention of security violations but assists in their detection. Security auditing is crucial both within a system and over the network. The main role of this service is to collect security relevant information from the system operation, application program facility, and network platforms that are used to test the adequacy of the security controls. Following an audit analysis, an entity may be held accountable for its actions so that violations or attempted violations of system security may be traced uniquely to it. To audit information, this service may use other services such as authentication, integrity, confidentiality and non-repudiation.

7. Availability and Prevention of Denial of Service

Denial of a service is usually regarded as an extreme case of information modification in which the information transfer is either blocked or drastically delayed. A measure against such an attack is provided by arranging the periodic exchange of information between entities to ensure that an open path exists between them. The greater the frequency of such a request response mechanism, the shorter the time period during

which the denial of service attack will remain undetected. The disadvantage of this service is that the effective bandwidth of the network is reduced.

8. Interchange Security

Since the EDI TSs is transmitted on the network, the security of interchange EDI information is more important than the security of the data itself. The purpose of interchange security is to protect the confidentiality of the message, to verify the originator of the message, and to verify the integrity of the message. These interchange security services include encryption, authentication, response messages, and key management.

Features of the public key cryptosystem are the prevention of attack by hard to solve problems (i.e., one-way function, or large number computing), and the separation of public key and secret key. In practice, most attacks on public key cryptosystems will probably be aimed at the key management levels (e.g., storage of the secret key, publication of the public key, distribution of the secret key, etc.), rather than at the cryptographic algorithm itself. In general, key management includes key generation, key distribution, key validation, and key storage, etc. We will discuss some key management systems covered by the EDI-related security standards. Other related discussions on key management issues may be found in PEM, RSA, PKCS and DSS.

V. SECURITY MANAGEMENT OF ELECTRONIC DATA INTERCHANGE

A. INTRODUCTION

Security management of EDI and DS was discussed in Chapter II and III. Chapter IV discussed the technical aspects of vulnerabilities, threats and required security mechanisms for preventing attacks. In this chapter, we discuss the managerial aspects, such as the internal controls of the EDI environment, auditing, security of EDI platforms, personnel security, etc.

Does a digital signature provide enough security in an EDI system? What types of security tools are available when interchanging EDI information? What are the main security services provided by various security standards? How can the EDI users be best protected?

The first part of this chapter surveys standards that provide security services and mechanisms that protect EDI implementations from threats. The main focus here is on security standards regarding the usage of cryptographic systems for interchange security for EDI transactions. Additionally, a security management model will be suggested for EDI system implementation. Questions to address include: What are the security goals of the organization? What is the security policy? What are the concerns of this model? What are the business considerations when implementing EDI? All these EDI security management issues will be discussed.

B. EDI-RELATED SECURITY STANDARDS

1. Introduction

This section surveys the security standards relevant to EDI users. This security includes the ANSI X.9 series for the financial community, the ANSI X.12 series for the EDI community and CCITT X.400, X.500, X.509, etc. for messaging systems. These EDI-related security standards are complicated and ambiguous, and can be difficult for new EDI participants to implement. Some security mechanisms recommended by these standards are

becoming obsolete because of the emergence of new advanced cryptosystem techniques. The purpose of this section, however, is to survey those security services and mechanisms work.

2. The Use of ANSI X.9

a. Introduction

In the late 1970s, the financial community realized that the traditional password systems would neither provide the requisite security nor the transaction processing growth potential for the future. A working group under the Financial Services Committee of ANSI developed a message authentication standard which is coded X9, which represented a step toward solving these problems.

Because the public key algorithm was not yet available at that time, standards, like ANSI X9.9-1982, Financial Institution Message Authentication (Wholesale), utilized the Data Encryption Algorithm and secret key. A cryptographically derived checksums called a Message Authentication Code (MAC) was computed from the financial message ANSI X12.58 EDI Security Structure standard, which uses ANSI X9.23 (Encryption of Wholesale Financial message) as a security model. The X9.23 has a number of modes of operation for encryption which are selected by a user prior to use.

b. ANSI X9.9: Message Authentication Standard

ANSI X9.9 defines a method for authenticating the integrity of a message. It is used by a message receiver to verify the identity of the data originator and to detect modifications of the data in the message. Also, ANSI X9.9 provides different authentication options:

(1) *Binary Data*. The MAC is applied to the entire body of the message, regardless of the coding of the data in the message. For example, coding may consist of EBCDIC or ASCII characters, or of machine code. Although a unique message identifier is not mandated by the standard for this option, it is highly recommended in order to provide for the detection of message replay and deletion.

(2) *Coded Characters.* Coded characters include the entire message and extracted message elements. All characters to be authenticated are represented by eight-bit ASCII characters. If the message is transformed into a different character set after MAC computation by the message originator (i.e., before or during transmission), the inverse transformation must be applied before the receiver begins the authentication process. The date of message origination and a message identifier must be present in messages authenticated using these options and are included in the list of elements that are submitted to the authentication algorithm.

(3) *Field Delimiters.* A delimiter is a group of characters used to identify the beginning and end of one or more data fields or message elements. These are required for the coded character set options so that the application can determine the message elements to be authenticated. Beginning and ending delimiters must occur in complementary pairs without intervening delimiters. Information may occur in the message outside a complementary pair of delimiters. This information will not be authenticated when a coded character set authentication option is selected. *Explicit delimiters* are specific strings of one or more characters which are used to demarcate message elements. ANSI X9.9 identifies explicit delimiters for the date of message origination, the identity of the message authentication key, the MAC, the message identifier and all other message elements. Implicit delimiters may be used if the position of a message element is fixed or unambiguously identified in a message by format rules. Both the originator and the recipient must have previously agreed to their use. Note that the format rules could consist of the definition of explicit delimiters other than those defined in ANSI X9.9.

c. ANSI X9.17: Key Management Standard

(1) *General Key Management Requirements.* The ANSI X9.17 Financial Institution Key Management (for wholesale) Standard describes a standard level of protection to assure the security of keying material. This includes cryptographic keys and other related information needed to manage the keys and the Key Management Facility

(KMF), the physical enclosure containing the cryptographic elements. The minimum requirements specified by the standard include:

- Control of keying material during the entire life of the keys. The key must be randomly generated and exists until the key is expired or compromised (should be expired at that time). Keys must either be physically secured or encrypted to protect keys from unintended or unauthorized disclosure. When physical protection is not possible, keying material may be cryptographically authenticated in conjunction with a counter.

- Secure distribution of keys to permit interoperability between communicating parties and their various cryptographic equipment or facilities. To support the varying needs of financial institutions and to permit interoperability, the standard defines both manual and automated methods for the exchange of cryptographic keys.

- Ensuring integrity of keys and the KMF. Overall key management must be put in place with the proper procedural controls. The keys, KMF, and procedural controls must be continually tested and monitored to ensure that the entire key management process is secure.

- Recovery in the event of failure, that is, ability to maintain a defined level of protection when the integrity of a key of the key management process is compromised.

(2) *Automated Key management architecture*. The ANSI X9.17 standard is based on a hierarchically structured set of cryptographic keys designed for automated distribution of keys over a computer network. Three distinct classes of keys form the hierarchy:

- Manually distributed key encrypting keys (KKMs). KKMs may be single key pairs or may encrypt other keys for distribution. Since key encrypting keys typically have longer cryptoperiods than data keys, key pairs may be used for increased protection. The KKMs form the basis for a keying relationship between two communicating parties.

- Automatically distributed key encrypting keys (KKs). In order to automatically exchange keys over a computer network, both parties must share a single KKM or set up a KKM pair with a trusted third party or key center.

- Automatically distributed data keys (KDs). Single KDs may authenticate messages used to distribute keys and encrypt or authenticate data. Once the KKM is shared between two parties or a common key center, additional keys may be encrypted for automatic distribution over a computer network. At a minimum, the standard requires a two-layer architecture in which KKMs encrypt KDs for distribution. An optional three-layer architecture may be used in which KKMs encrypt KKs for distribution and KKs encrypt KDs for distribution. In both cases, initialization vectors (IVs) are used to destroy any residual pattern contained in the encryption data. In some cases, the same IV must be used for decryption at the receiving site as at the transmitting site to allow the decryption to be successful. Therefore, synchronization of the IV's is required. If the IVs are encrypted, they must be encrypted by KDs.

(3) *Key Generation.* The methods used to generate new keys and IV's must be secure. If the methods involve manual procedures, then they must be protected as under dual control. Automated procedures must be physically and logically protected. New key's and IVs must be generated so that they are random. Any one of the possible keys or IVs must be equally likely to be generated. Each key or IV generated must have no apparent relationship to its predecessors or successors. The protection provided in the key generation process must be at least as great as that required for the data, so that the cryptographic protection obtained by encryption and authentication with the generated keys is not compromised by the key generation process.

(4) *Key and IV Encryption and Decryption.* The standard recommends using the ANSI DEA to encrypt and decrypt keys and IVs for automated distribution. Single KKMs and KKs encrypt and decrypt other single keys using DEA in the Electronic Code Book (ECB) mode. Single keys may not be used to encrypt and decrypt key pairs. KKMs or KKs may be used to encrypt and decrypt single keys and other key pairs using DEA in the ECB mode to perform multiple encryption. To perform multiple encryption of a key with a key pair, the key is encrypted by the first key in the key pair, the result is then encrypted by the second key in the key pair, and finally the second result is encrypted again

by the first key in the key pair. Even if a KMF is designed to use key pairs, encryption may be performed by a single key using the same key as both the first and second keys of a key pair. If IVs are encrypted, then KDs encrypt and decrypt the IVs using DEA in the ECB mode.

(5) *Key Counters and Key Offsetting*. The standard requires the use of key counters to control the automated distribution of encrypted keys. By counting the messages transmitted over a computer network distributing encrypted keys, the replay of previously transmitted messages may be detected and messages received out of sequence may be recognized. When two communicating parties exchange keys directly between themselves, two separate counters (a transmit count and a receive count) must be maintain by both parties. The counters are kept with KKM or KK used to encrypt keys for distribution. Under normal conditions, the originating party's transmit count should equal the receiving party's receive count for the message to be accepted. The standard requires that all keys encrypted for distribution must be protected by key offsetting. Key offsetting simply combines the transmit count with the KKM before it encrypts other keys for distribution. The encrypted key must be encrypted with the same count. This prevents previously distributed encrypted keys, whose content may have been disclosed for some reason, from being retransmitted. The encrypted key must be transmitted with a new count but can only be decrypted using the original count.

(6) *Key Notarization*. Another protective cryptographic feature of the standard is the support of key notarization. Key notarization is similar to the action of a notary public who first requires a customer to be identified before the customer's signature is notarized on a document. In automated key distribution, electronic notarization seals encrypt keys with the identities of the originator and the intended recipient. A notary seal is formed by combining the KKM or KK to be used with the identities of the two parties. This notary seal is then offset by the counter associated with the KKM or KK to form a notarizing key to encrypt keys for distribution. Once keys are sealed or notarized, they may only be decrypted by the same notarizing key. The standard requires that all keys generated

by a key center for use by two particular parties be notarized with the identities of those parties. Security is increased as the notarized keys cannot be decrypted and used by any other parties. In an EDI environment, all keys sent in cryptographic service message (CSM) TSs shall be notarized. Notarization as defined in Section 7 of ANSI X9.17 is interpreted (for EDI) to graphically combine the Security Originator Name and the Security Recipient Name as they appear in the header of a CSM. This guarantees that the key is the one designated for use by the two named parties.

(7) *Automated Key Distribution Protocols.*

- CSM and authentication. CSMs are exchanged between two communicating parties to establish new keys and to discontinue existing keys. These fixed-format messages (transmitted in plaintext) may carry encrypted keys, IVs, and other keying material, such as the identities of the two parties, the identities of the keys, and the counts. To guarantee the integrity of a CSM when it is transmitted over a computer network, its originator must authenticate the entire contents of the message in accordance with the ANSI X9.9 Message Authentication Standard. The results of MAC in the CSM must be included so it can be verified by the recipient. Certain CSMs must be authenticated with the KDs being established or discontinued. Since these KDs will be encrypted with a KKM or KK shared only between the actual originator and the recipient of the KDs, the identity of the actual originator of the CSM may then be verified by its recipient.

- Point-to-point environment. A point-to-point environment exists when two parties share a KKM or KK pair so that further key encrypting keys and KDs may be exchanged. At least one of the parties must have the capability to generate or otherwise acquire keys. The implementation of the point-to-point environment is the minimum required for automated key distribution and will not be efficient for systems of greater than 20 partners.

- Key center environment. Key center environment includes the key distribution center (KDC) environment and the key translation center (KTC) environment. A KDC exists for the purpose of distributing generated or acquired KDs to two parties who

wish to communicate with each other but do not currently share keys. Each may share a KKM or KK pair with the KDC but may not have the ability to generate keys. One party may request KD from the KDC for later communication to another party. The KDC generates or acquires the KDs and sends two identical sets to the originator using a KKM or KK pair shared with the originator to key offset and encrypt one set of keys, and a KKM or KK pair shared with the recipient to key offset and encrypt the other set of keys. The originator then sends the second set to the recipient.

A KTC is used to translate keys for future communication between parties who have the similar situation with a KDC. KKMs or KKs and KDs may be translated and exchanged, but only one of the two types is actually processed by the center at one time. A key to be used for future communication with the other party is sent to the KTC, encrypted under the offset key encrypting key pair shared between the originator and the KTC. The KTC decrypts this key, reencrypts using notarization and sends the reencrypted version back to the originator. The originator then sends the reencypted version to the recipient.

d. ANSI X9.23: Encryption of Financial Messages

ANSI X9.23 defines a method for the encryption and decryption of the entire message to provide confidentiality (privacy) of the data in the message.This standard also includes several filtering algorithms.

(1) *Mode of Operation*. Several modes of encryption have been identified for encryption using the DES algorithm. The modes of operation used by implementation of ANSI X9.23 are:

- Cipher Block Chaining (CBC). The CBC encryption mode is used to encrypt 64-bit blocks of information. Successive ciphertext and plaintext blocks are chained together until the last plaintext block is encrypted. A 64-bit IV is used during the encryption of each message in the CBC mode. If a new IV is not provided for each message, then a new 64-bit Initial Text Sequence (ITS) must immediately precede the message data to be encrypted. If the data to be encrypted does not consist of an even multiple of 64 bits,

then the final block is padded with additional bits before encryption. The resulting ciphertext which includes the encrypted padding bits is sent to the receiver. After decryption by the receiver, the padding bits are discarded. The padding may consist of either bits or octets (group of 8-bits). If the logical content of the data to be encrypted is in 8-bit units (e.g., ASCII or EBCDIC characters), octets should be used for padding. The CBC mode of encryption is desirable when more than a minimal amount of data is to be processed or an immediate reaction to each character is not required (e.g., during an interactive session). The addition of a few bits of padding are insignificant when compared to the advantage gained by encrypting 64-bits at a time.

- 1-Bit Cipher Feedback (CFB-1). The CFB-1 mode is used to encrypt 1 bit of plaintext information at a time. 64 bits of output are obtained from the encryption process; one bit is combined with the plaintext bit, and the other 63 bits of output are discarded. A new IV is used for each message to be encrypted. CFB-1 is, therefore, the most inefficient mode of encryption, but can be useful in an interactive session when the receiver needs to react immediately to bit patterns other than multiples of 8 or 64 bits.

- 8-Bit Feedback (CFB-8). The CFB-8 mode is used to encrypt 8 bits of plaintext information at a time. 64 bits of output are obtained from the encryption process; eight bits of the output are combined with eight bits of plaintext, and the remaining 56 bits are discarded. A new IV is used for each message to be encrypted. CFB-8 is less efficient than CBC for the encryption of more than minimal amount of data. If an interactive session is in progress, CFB-8 allows the receiver to respond to each 8-bit group as it is received and is encrypted.

(2) *Encryption and decryption of entire messages and parts of messages*:

- Entire messages. When an entire message is to be encrypted (the IV and ITS) the message including the MAC and the padding field will be encrypted as a unit. Following encryption, the resulting ciphertext may be filtered, as discussed below. The receiver defiliters the receiver message and encrypts the ciphertext as a unit. The encryption of an entire message is required to maintain confidentiality between the originator and the

receiver. Historically, cases exist that show that much information can be derived from mere bits and pieces of information.

- Parts of messages. Plaintext encryption elements are those part of a plaintext message to be encrypted. After encryption, the resulting ciphertext may be filtered for transmission. This filtering nature may be used in non-transparent networks for the reason of control. The receiver defiliters the message prior to decrypting the ciphertext. The encryption of plaintext encryption elements will allow the encryption of parts of a message using different keys, e.g., the encryption of portions of a message by different people with different responsibilities. The encryption of plaintext encryption does not afford the same level of security as entire message encryption.

(3) *Methods for Encryption.* Three methods for encryption of these elements within a message and of the resulting ciphertext encryption elements are permitted by this standard:

- Plaintext encryption elements are independently encrypted into ciphertext encryption elements. Each ciphertext encryption element is placed in the message in the location of the original plaintext encryption element. Ciphertext encryption elements may be longer than their corresponding plaintext encryption elements due to the addition of padding prior to encryption or due to the use of a filtering mechanism. The receiver independently decrypts each received ciphertext encryption element, obtaining the original encryption element.

- Plaintext encryption elements are extracted from the message, concatenated and encrypted as a single string. The ciphertext string may be longer than the plaintext string due to the addition of padding prior to encryption or to the use of a filtering mechanism. It is quite possible that the number of additional bits could be fewer than those required if plaintext encryption was used. The message originator then transmits the resulting message. The receiver decrypts the ciphertext string and inserts the plaintext encryption elements into the message text. This requires that the plaintext string contain some identification of the location of the plaintext encryption elements in that string.

- Plaintext encryption elements are extracted from the message, concatenated and encrypted as a single data string. The resulting ciphertext string is divided into ciphertext substrings which are inserted in place of the plaintext encryption elements in the message. Each bit of the plaintext encryption element is replaced by a single ciphertext bit, selected in sequence, except for the last encryption element which contains all the remaining ciphertext bits. This last element may be longer than the original plaintext encryption element due to the addition of padding prior to encryption or the use of a filtering mechanism. The receiver concatenates the ciphertext substrings and decrypts the result as a single ciphertext string. The resulting plaintext encryption elements are inserted into the message text.

(4) *Filtering*. ANSI X9.23 identifies the following filtering techniques for encrypted data prior to transmission:

- No Filtering. This may be used when the communication system is not sensitive to control characters or strings, or if a process later on is responsible for message filtering.

- Hexadecimal Filtering. Hexadecimal filtering is accomplished by converting every four bits of ciphertext into one of 16 characters (from 0 to 9 and A to F) accepted by the communication system. This technique might be used when communicating within a network which automatically converts lower case letters to uppercase or is sensitive to lower case letters and punctuation characters. Since ASCII characters are represented as eight binary bits, this results in an expansion of 100% (i.e., an eight bit ASCII character replaces four bits of ciphertext).

- ASCII Filtering. This technique is useful in communications systems in which all non-blank, non-hyphen, printable characters are acceptable. A simple table is used for the conversion from ciphertext bits to characters which are to be transmitted. If the ciphertext was produced by the CFB-1 or CFB-8 modes of encryption, a padding of 4 to 16 bits is added to the ciphertext during the filtering process. If the CBC encryption mode was

used, the padding will consist of 0-13 bits. When eight bit ASCII characters are transmitted, this filtering technique results in an average expansion factor of 23%.

- BAUDOT or ASCII/BAUDOT Filtering. This technique might be used in networks using BAUDOT. The ciphertext is mapped into 20 letters which can be represented as either ASCII or BAUDOT characters. If the ciphertext was produced by the CFB-1 or CFB-8 modes of encryption, a padding of 4 to 16 bits is added to the ciphertext during the filtering process. If the CBC encryption mode was used, the padding will consist of 0-13 bits. When five bit BAUDOT characters are used, the average expansion factor is 16%. When eight bit ASCII characters are used, the average expansion factor is 86%.

- User-defined Filtering. This type of filtering is defined by a bilateral agreement between the originator and the receiver.

3. Security Structure in ANSI ASC X12

a. Introduction

This standard series defines the data formats required for authentication and encryption that provide integrity, confidentiality, and verification of the security originator to the security recipient for two levels of exchange of EDI. The two levels are the functional group (FG) level and the transaction set (TS) level. A TS includes a header control segment, one or more data segments, and a trailer control segment. The TS may also include a transaction security header control segment and transaction security trailer control segment when it is desirable to send the data within the transaction envelope in an authenticated form. The data element in the TS header identifies the type of TS. A FG contains one or more related TSs preceded by FG header control segment, and terminated by a FG trailer control segment. The FG may also include a functional security header control segment and a functional security trailer control segment when it is desirable to send the data in an authenticated form. Related security management standards are defined in ANSI X12. ANSI X12.6 shows the application control structure, whereas ANSI X12.58 discusses the security structure of EDI. X12.42 is used to encrypt the distribution of the keying material

to data security entities based on the ANSI X9.17 process with encoding according to EDI syntax rules. The security issues of these three standards are discussed below.

b. ANSI X12.6: EDI Application Control Structure

This standard defines the structure of business transactions for computer-to-computer interchange. This structure is expressed using a symbolic representation of ANSI X12 data independent of the physical representation (e.g., character set encoding). The symbolic representation is expressed in terms of both the design and use of X12 structures. This includes the control segments used to bound loops of data segments together, such as TSs and groups of related TSs. The special control segments used to bound TSs and groups of TSs for security purposes are also presented.

(1) *Two Levels of EDI Interchange Data Formats.*

- Transaction set (TS). The TS is the basic unit of information exchanged between trading partners. The TS shall consist of a TS header segment, an optional transaction security header segment, one or more data segments in a specific order, a transaction security trailer segment whenever the security header segment is present, and a TS trailer segment. The TS may also include a transaction security header control segment and transaction security trailer control segment when it is desirable to send the data within the transaction envelope in an authenticated or encrypted form, or both. Encryption and authentication are processes that are external to the X12 syntax processor. The overview syntax of an X12.6 secure TS is listed below. A discussion of the syntax notation and rules is beyond the scope of this thesis. The data elements used in the security header and security trailer segments are defined in the ANSI X12.58 security structure. The syntax notation used in this standard is the Abstract Syntax Notation one (ASN.1), defined in CCITT X.208.

<transaction_set>::=<secured_tran_set>

<secured_trans_set>::=<trans_set_header><trans_security_header>

<data_segment_group>{<data_segment_group>}

<trans_security_trailer><trans_set_trailer>

<trans_security_header>::=S2S<security_header><tr>

<security_header>::=<gs><security_type>

.....<gs><security_originator>

.....<gs><security_recipient>

.....<gs><authent_key_name>

.....<gs><authent_serv_code>

.....<gs><encrypt_key_name>

.....<gs><encrypt_serv_code>

.....<gs><data_length>

.....<gs><initialization_vector>

<security_type>::=<id>

<security_originator>::=<string>

<security_recipient>::=<string>

<authent_key_name>::=<string>

<authent_serv_code>::=<id>

<encrypt_key_name>::=<string>

<encrypt_serv_code>::=<id>

<data_length>::=<numeric>

<initialization_vector>(16/16)::=<fixed_length_string>

<trans_security_trailer>::=<tr>S2E<gs><MAC>

<MAC>(09/09)::=<fixed_length_string>

- Functional Group (FG). The FG shall consist of a FG header segment, an optional functional security header segment, one or more similar TSs, a functional security trailer and a FG trailer segment. The functional identifier (<function_id>) defines the group of transactions that may be included within the FG. An example of a secure FG is shown below:

<secured_function_group>::=<function_header>

<function_security_header>

<transaction_set>{<transaction_set>}

<function_security_trailer>

<function_trailer>

<function_security_header>::=S1S<security_header><tr> (security header is defined above)

<function_security_trailer>::=S1E<gs><MAC><tr>

(2) *Control Segment.* A control segment is used for transferring control information rather than application information. This control segment includes loop control segments, TS control segments, functional group control segments, and security control segments. All control segments are used to delineate segment information. The relations of control segments are as follows:

GS-Functional Group Header, starts a group of related TSs.

S1S-Function Security Header which defines the starting boundary of authenticated and/or encrypted data within a functional group.

ST-TS Header, starts a TS.

S2S-Transaction Security Header which defines the starting boundary of authenticated and/or encrypted data within a TS.

LS-Loop Header, starts a bounded loop of data segments but is not part of the loop.

LS-Loop Header, starts an inner, nested, bounded loop.

LE-Loop Trailer, ends an inner, nested, bounded loop.

LE-Loop Trailer, ends a bounded loop of data segments but is not part of the loop.

S2E-Transaction Security Trailer which defines the ending boundary of authenticated and/or encrypted data within a TS.

SE-Transaction Group Trailer, ends a TS.

S1E-Function Security Trailer which defines the ending boundary of authenticated and/or encrypted data within a functional group.

GE-Functional group Trailer, ends a group of related TSs.

More than one ST/SE pair (each representing a TS) may be used within one functional group. Also, more than one LS/LE pair (each representing a bounded loop) may be used within one TS. All security segments are optional in this standard. [Ref.37]

c. *ANSI X12.58: EDI Cryptographic Security Structures Standard*

EDI transactions may be protected, using authentication, against unauthorized alteration during interchange or storage. EDI transactions may also be protected, using encryption, from unauthorized viewing during interchange and storage. These are complementary and not competing processes. Either or both of these processes may be used simultaneously to protect EDI transactions being transmitted from computer to computer or stored within a computer system. EDI transactions may be protected at the functional group or TS levels of an EDI interchange or both. A single TS may be authenticated and encrypted and an includeed functional group may also be authenticated and encrypted. Accordingly, it is possible to have portions of a single interchange authenticated and encrypted at various levels for different business purposes. The ANSI X12.58 standard allows for authentication alone, for authentication and encryption, and for encryption alone. The purpose of this standard is to define the data formats required for authentication and encryption that provide integrity, confidentiality, and verification of the security originator to the security recipient for two levels of exchange of EDI formatted data (the FG level and the TS level). This standard defines data formate for authentication and encryption of EDI encoded data. The business requirements address in this standard encompasses:

- Verification of the security originator of a message to the security recipient.
- Verification of data integrity.
- Confidentiality of business data.
- Detection of replay, insertion, modification, deletion, or impersonation.

(1) *ANSI X12 Security Segment Specifications*. The sequences of both the FG and TS levels of segments are illustrated in the introduction of ANSI X12.6. The principles of the standard structure are established as implemented at the various levels. The security header (SxS, x=1 or 2) and trailer (SxE) segments are defined for the two levels using identical formats, differing only in the naming of the segments and the reference designators. The same key shall not used for more than one function (e.g., authentication

and encryption) at any one level, and a key used to provide protection at one level shall not intentionally be used at another level for the same or a different purpose. When used, the security header segment is placed immediately following the first segment of the authenticated group of segments. The security header segment identifies the security services, authentication, and the encryption keys to be used. The security trailer segment ends the authentication and encryption processes initiated at the corresponding security header segment. SxE01 is the MAC generated by applying the authentication algorithm against the designated segments. The format of the SxS and SxE segment is shown in Figure 1 and 2.

The preparation of the X12 text and formatting of security elements consist of:

- Designating the segments applicable for authentication or encryption.

- Inserting the SxS segment after the first segment and the SxE segment before the last segment of those designated.

- Setting the Security Type (SxS01) to reflect the combination of authentication or encryption to be used and the algorithm to be employed.

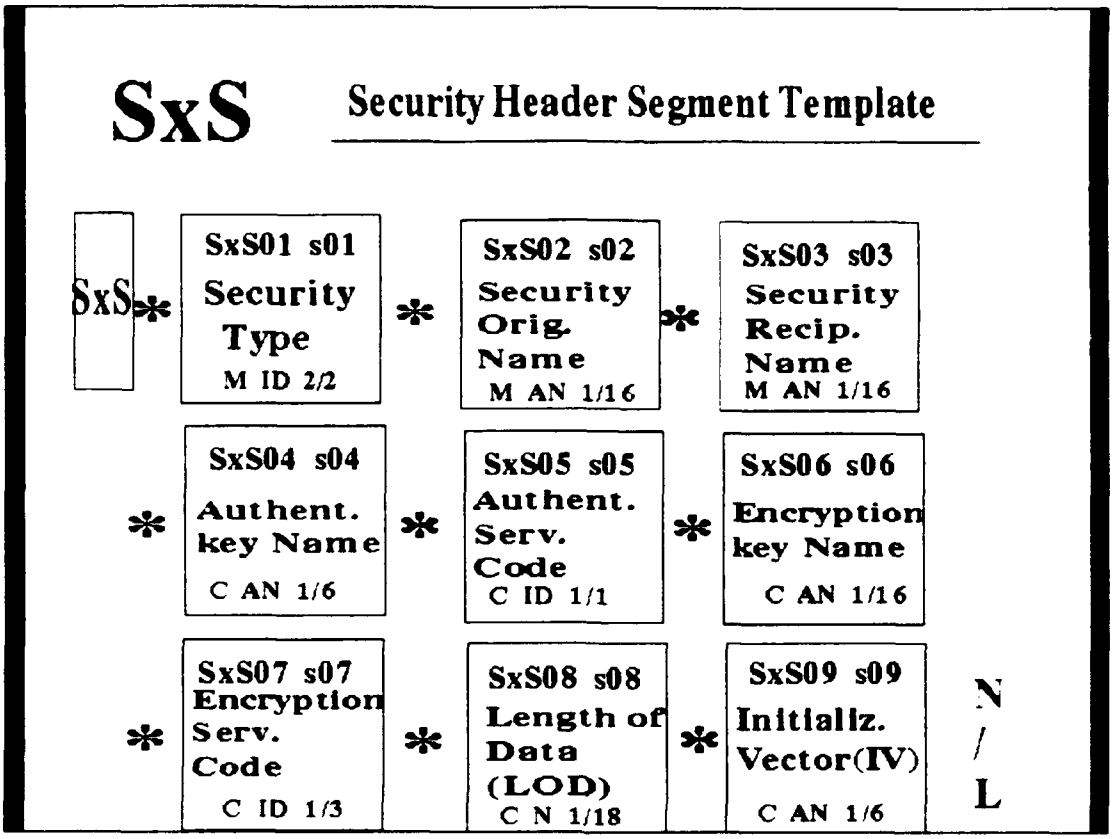

Figure 1. ANSI X12 Security Header Segment (SxS)

- Filling in the Security Originator Name (SxS02) and Security Recipient Name (SxS03) appropriate to the business partners of the transaction(s) being processed.

- If authentication is being used, the Authentication Key Name (SxS04) and Authentication Service Code (SxS05) is inserted; if authentication is not being used, these optional fields are set to empty (i.e., length zero) and the MAC code (SxE01) is set as blank for the minimum field length.

- If encryption is being used, the Encryption Key Name (SxS06), Encryption Service Code (SxS07), and IV (SxS09), and the Length of Data (SxS08) fields are set to

empty; if encryption is not being used, all four data elements and their preceding data element separators are omitted according to ANSI X12 coding conventions.

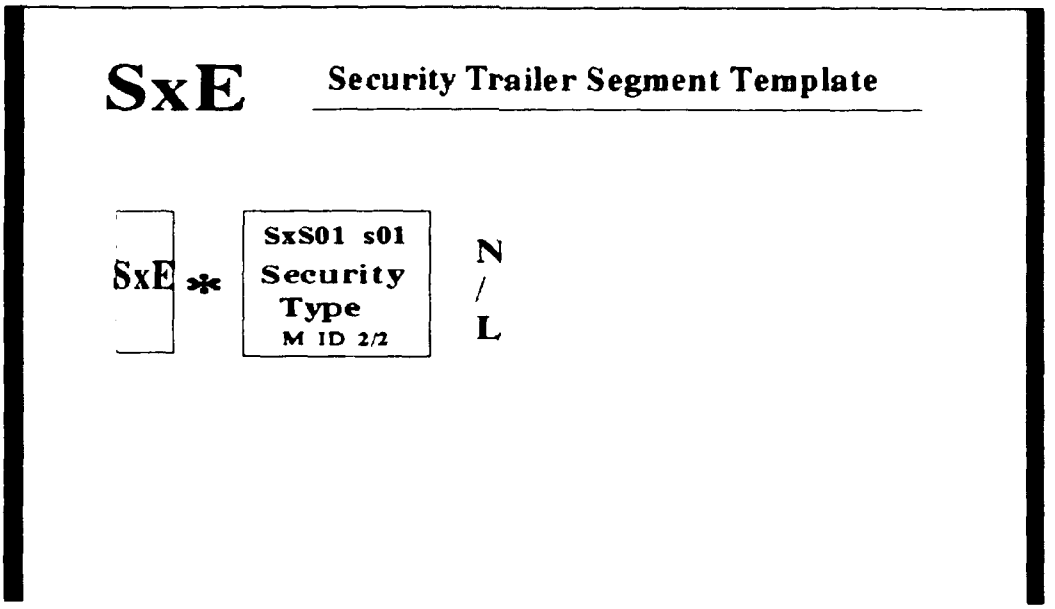

Figure 2. ANSI X12 Security Trailer Segment (SxE)

(2) *The Authentication Process in ANSI X.12.58.* Authentication is a technique used to verify the identity of the originator of a message to the recipient in order to detect spoofing or impersonation. The integrity of the message is verified by detecting changes (modifications) in a message (including transmission errors) introduced between the security originator and the security recipient. A unique message identifier is used to detect attempts at insertion, deletion, or replay of messages. The ANSI X9.9 standards employs ANSI X3.92 to compute a Message Authentication Code (MAC). The MAC is a cryptographic derived hash code used to verify the originator to the recipient and to protect the integrity of the applicable data. The MAC protects the data whether the data is numeric, text, or punctuation. The key used to perform the authentication should be known only to

these two parties (or their authentication authority), and the message must not be modified after the authentication process is performed by the originator. Authentication does not modify the text input in the process. It adds a nine-character code to the message that is expressed in the same character set as the message. The authenticated message can then pass through any communications equipment with the ability to handle an unauthenticated message. The receiving process must be able to reconstruct the original message from the data delivered. Basic requirements for a secured business interchange include the need to detect attempts at insertion, deletion, and replay of messages. These requirements may be satisfied by having the originating application place a combination of date and message identifiers (such as a purchase order number) (according to the requirements of ANSI X9.9) into the authenticated message so that the receiving application can uniquely identify each message sent. Then the recipient can detect a repeated message from its repeated date and message identifier. Deleted messages may be detected at origination by the lack of an authenticated response or at receipt by reconciling the message received against a list sent separately by the originating application in a separate authenticated message. Inserted messages may be detected by the receiving application by authentication failure. Authentication in ANSI X12.58 does not protect the confidentiality of the message as information is interchanged in plaintext form. Message encryption may be used to provide confidentiality when using authentication to provide integrity protection of the same data. The scope of the authentication process is defined as follows:

- Authentication begins with the first character of the segment designated for authentication (i.e., the first character of the segment preceding the "SxS" segment).

- If encryption is also to be performed, the authentication process is performed through the SxS07 element and then continues with the segment terminator of the SxS segment (i.e., as if the Length of Data element, the IV element, and their preceding segment separators were not present).

- Authentication includes and ends with the data element separator preceding the MAC code.

(3) *The Encryption Process of ANSI X12.58.* The confidentiality of EDI encoded data is protected by the proper use of encryption. This standard provides a method for protecting the confidentiality of X12 formatted data that is independent of the actual structure and the data content. Encryption of X12 messages may be performed using ANSI X9.23. In all cases, the structures defined in this standard apply. Use of the encryption option of this standard does not provide detection of accidental or deliberate alternation of messages between the originator and the recipient. Authentication can be used to provide integrity while using encryption to provide confidentiality of the same data. The scope of the encryption process is defined as follows.

- Encryption begins with the Initialization Vector field (SxS09).

- The Initialization Vector (IV) is a 64-bit binary value that is encrypted using Electronic Code Book (ECB). A new IV shall be intentionally reused. Text following the IV is encrypted using the mode specified in the Encryption Service Code (SxS07).

- Encryption ends with the last character before the segment preceding SxE.

- Filtering is applied to both the encrypted IV and the encrypted data.

- When present, the Length of Data data element counts the number of octets in the encrypted and filtered and data.

- When the CBC mode of encryption is used, the plaintext data is padded to a multiple of eight octets by adding text consisting of 0 to 7 ASCII-zeroes followed by a length value consisting of one ASCII character with value ranging from "1" to "8" that specifies the number of octets used in the padding. This padding meets the requirements of ANSI X9.23 and is completely deterministic. Because of the padding, the pre-encrypted text will always be a multiple of eight octets prior to filtering, and thus fits any 64-bit block-oriented encryption mode of operation. The padding is added immediately prior to encryption and is removed by the encryption process. It is not included in the calculation of the MAC that is computed prior to encryption.

(4) *Information Transmitted.* The security segments provides space for the information needed to secure EDI data. The security type and the name of the recipient and

originator are always present. The authentication or encryption information is only included when those options are selected by the type code.

(5) *Segment Placement.* The placement of the security segments, S1S and S1E for functional groups and S2S and S2E for TSs, are discussed above. The header is placed immediately following the group start or TS start segments, and the trailer is placed immediately preceding the group trailer or TS trailer. Thus, the security segments form an envelope around the data to be secured and must always be paired (one start segment with one trailer segment). Note that the security count added to an existing EDI TS will increase the segment count in the SE segment.

(6) *Security Control Segments.* The security control segments that are part of EDI control segment are used to delineate information that is to be encrypted or authenticated or both. After designating the segments applicable for authentication or encryption or both (which encompass either a functional group or TS), the S1S or S2S is inserted after the first designated segment (a functional group header or TS header, respectively). The S1S and S2S specify a code identifying the security algorithms and methods that are to be employed for this level of interchange. The S1E and S2E trailer segments contain a MAC code, is used to verify the integrity of the authenticated segments.

d. ANSI X12.42:Cryptographic Service Messages (CSM) Transaction Set

When keying materials are exchanged in an EDI environment, the content of the key exchange message is generated according to the rules of ANSI X9.17 and the format of the message is translated into an ANSI ASC X12 CSM Transaction Set (Transaction Set 815). ANSI X12.42 provides for the distribution of keying material to data security entities based on the ANSI X9.17 process with encoding according to EDI syntax rules.

ANSI X12.42 has defined some of the EDI options from ANSI X9.17 (as discussed in the previous section):

(1) The use of key encrypting key pairs rather than single key encrypting keys.

(2) The use of key notarization in all environments.

(3) IVs are not to be sent in CSMs. They appear, when necessary, in the headers of TSs and functional groups.

(4) All keys are to be named, and the names are always to be used ANSI X9.17 allows the use of unnamed keys if only one key of a type is shared. This procedure is sometimes unclear if a key name is to appear in a message when the key is named.

4. Security Issue in CCITT X.400

The distributed nature of CCITT X.400 Message Handling System (MHS) makes it desirable that mechanisms are available to protect against various security threats that can arise. In this standard, a security model has been built which includes the procedure for secure access management and administration, and secure messaging.

a. Secure Access Management and Administration

The capability of this section covers the establishment of an authenticated association between adjacent components and security parameters for the association. These parameters may be applied to any components in the MHS.

b. Secure Messaging

The capabilities of secure messaging covers the application of security features to protect messages in the MHS in accordance with a defined security policy. This includes security services enabling various components to verify the origin of messages and the integrity of the content, and to prevent unauthorized disclosure of the message content. Also, protect message directly submitted to the message transfer system by a user agent, message store, or an access unit. They do not cover the application of security features to protect communication between users and the MHS, or MH user-to-MH user

communication (a large part of MH user-to-MH user communication is protected between two UAs).

Many of the secure messaging elements of service provide an originator to recipient capability and require the use of user agents with security capabilities. They do not require the use of a message transfer system with security features. For example, content confidentialities can be applied by enciphering the message content by the originator, and deciphering by the recipient, with various security parameters transferred within the message envelope.

Some of the secure messaging elements involve an interaction with the message transfer system, and require the use of message transfer agents with security capabilities. Some apply to MS, as well as UAs and MTAs, such as message security labelling. In general, the MS is transparent to security features that apply between the originators' and the recipients' UAs. This describes which MHS component is the "provider" or which is the "user" of the security service. For example, probe origin authentication is provided by the originating UA, and may be used by the MTAs through which the probe passes.

c. MHS Security Capabilities

An overview of these capabilities follows [Ref.38]:

(1) *Message Origin Authentication*. Enable the recipient or any MTA through which the message passes to authenticate the identity of the originator of a message.

(2) *Report Origin Authentication*. Allows the originator to authenticate the origin of a delivery/non-delivery report.

(3) *Probe Origin Authentication*. Enable any MTA through which the probe passes to authenticate the origin of the probe.

(4) *Proof of Delivery*. Enables the originator of a message to authenticate the delivered message, its content, and the identity of the recipient(s).

(5) *Proof of Submission*. Enables the originator of a message to authenticate that the message was submitted to the MTS for delivery to the originally specified recipient(s).

(6) *Secure Access Management*. Provides for authentication between adjacent components and the setting up of the security context.

(7) *Content Integrity*. Enables the recipient to verify that the original content of a message has not been modified.

(8) *Content confidentiality*. Prevents the authorized disclosure of the content of a message to a party other than the intended recipient.

(9) *Message flow confidentiality*. Allows the originator of a message to conceal the message flow through MHS.

(10) *Message Sequence Integrity*. Allows the originator to provide to a recipient that the sequence of message has been preserved.

(11) *Non-repudiation of Origin*. Provides the recipient(s) with proof of origin of the message and its content which protects against any attempt by the originator to falsely deny sending the message or its content.

(12) *Non-repudiation of Delivery*. Provides the originator with a proof of delivery message to protect against any attempt by the recipient(s) to falsely deny receiving the message of its content.

(13) *Non-repudiation of Submission*. Provides the originator with proof of submission of the message, which protects against any attempt by the MTS to falsely deny that the message was submitted for delivery to the originally specified recipient(s).

(14) *Message Security Labelling*. Provides the capability to categorize a message, indicating its sensitivity, which determines the handling of a message in accordance with the security policy.

5. The Security Model in CCITT X.402

Security features are an optional extension to the MHS that may be used to minimize the risk of exposure of assets and resources to violations of security threats. Threats may be countered by the use of physical security, computer security (COMPUSEC), or security services provided by the MHS. Depending on the perceived threats, certain MHS security services will be selected in combination with appropriate physical security and COMPUSEC measures. The security services supported by the MHS are described below. [Ref. 39]

a. Security Policy

Security services in the MHS must be capable of supporting a wide range of security policies which extend beyond the confines of the MHS itself. The services selected and the threats addressed will depend on the individual application and levels of the trust in parts of the system.

A security policy defines how the risks and exposure of the assets may be reduced to an acceptable level.

b. Security Services in X.402

Massage transfer security services include:

(1) *Origin Authentication Security Service.* This security service provides for the authentication of the identity of communicating peer entities and sources of data. This service includes: data origin authentication security services that provide corroboration of the origin of a message, probe, or report to all concerned entities (i.e., MTAs or recipient MTS-users). These security services cannot protect against duplication of the message. Proof of submission/delivery security services enables the originator of a message to obtain corroboration that it has been received/delivered by the MTS for delivery to the originally specified/intended recipients.

(2) *Secure Access Management Security Service.* This service is concerned with providing protection for resources against their unauthorized use. Two components of

this system are peer entity authentication and the security context security services. The latter is used to limit the scope of passage of a message between entities by reference to the message security labels. Therefore, this service is closely related to the message security labelling security service.

(3) *Data Confidentiality Security Services.* These security services provide for the protection of data against unauthorized disclosure. Data confidentiality has been discussed in Chapter II, including: connection confidentiality, content confidentiality and message flow confidentiality.

(4) *Data Integrity Security Services.* These services are provided to counter active threats to the MHS. As discussed in Chapter II the features of data integrity are Connection integrity, content integrity and message sequence integrity.

(5) *Non-repudiation Security Services.* These services provide irrevocable proof to a third party after the message has been submitted, sent, or delivered, that the submission, sending, or receipt did not occur as claimed. For this to function correctly, the security policy must explicitly cover the message of asymmetric keys for the purpose of non-repudiation services.

(6) *Message Security Labelling Security Service.* This security service allows security labels to be associated with all entities in the MHS, i.e., MTAs and MTS-users. In conjunction with the security context security services, the implementation of security policies defines which parts of the MHS may handle messages with specified associated security labels.

(7) *Security Management Service.* A number of security management services are needed by the MHS. The only management services provided in this standard are concerned with changing credentials and registering MTS-user security labels. The former service enables one entity in the MHS to change its credentials as held by another entity in the MHS; the latter service enables the establishment of permissible security labels for one particular MTS-user.

6. The security issues covered by CCITT X.435

This standard is a form of message handling tailored to exchange of EDI information. The information objects that users exchange in EDI messaging are of two kinds: EDI messages (EDIM), and EDI notifications (EDIN). The security concerned in this standard for exchanging EDI is listed below.

a. EDIM

An EDIM is a member of the primary class of information objects conveyed between users in EDI messaging. An EDIM consists of a Heading and a Body. A Heading is a set of Heading Fields, each an information item giving a characteristic of the EDIM. The Body is a sequence of one or more body parts. Information items of several kinds appear throughout the Heading. The Heading includes the codes of the interchange recipient/sender, the identification code, the identification code qualifier and the routing address.

An example of the Heading Fields, according to the need of interchange security, are listed below. The syntax notation use in this standard is the abstract syntax notation one (ANS.1), that is defined in CCITT X.208.

Heading ::= SEQUENCE {

.....this-EDIM ..... [1] ThisEDIField,

.....edi-application-security-elements...[11]EDIApplicationsecurityElementsField OPTIONAL,

.....heading-extensions .....[19] HeadingExtensionsField OPTIONAL }

EDI Application Security Elements field allows an EDI application to exchange security elements having an end-to-end significance. For example,

EDIApplicationSecurityElementsField ::= SEQUENCE {

.....edi-application-security-element .....{0}

EDIApplicationSecurityElement..... ::=BIT STRING

EDIApplicationSecurityElement..... ::=SET OF EDIApplicationSecurityExtension

EDIApplicationSecurityExtension..... ::= ExtensionField

::}

The Body has one primary Body part that contains an EDI information object, and optionally, its delivery envelope. This body part is either an EDI interchange itself or a forwarded EDIM. The delivery envelope shall be present if security services are invoked.

b. EDIN

An EDIN is a member of the secondary class of information objects conveyed between users in EDI messaging. It includes the follows:

- Positive notification report its originator's acceptance of Responsibility of an EDIM;

- Negative notification reports its originator's refusal to accept responsibility of an EDIM; and

- Forwarded notification reports that responsibility of an EDIM has been forwarded together with the subject EDIM.

The security Element field in EDIN, is used to provide "proof/non repudiation of content received "EDI application security" services.

```
SecurityElementField ::= SEQUENCE {
    original-content                    [0] Content OPTIONAL
    original-content-integrity-check    [1] ContentIntegrityCheck OPTIONAL
    edi-application-security-elements   [2] EDIApplicationSecurityElementsField OPTIONAL
    security-extensions                 [3] SecurityExtensionField OPTIONAL }
SecurityExtensionField ::= SET OF SecurityExtensionSubfield
SecurityExtensionsSubField ::= ExtensionField
```

Security services are available only if the MHS supports secure messaging.

c. *Enhanced Security Model*

Annex I of CCITT X.435 describes the enhancements required to the security model defined in CCITT X.402. [Ref.40]

(1) *Security Services*. The additional security services include: Non-repudiation/Proof of Reception, Non-repudiation/proof of Retrieval, Non-repudiation/ Proof of Transfer, and Non-repudiation of Content.

(2) *Enhancements from CCITT X.402*. The Changes to CCITT X.402 include EDIM Responsibility authentication services and Non-repudiation of EDIM Responsibility. EDIM Responsibility authentication services include:

- Proof of EDI Notification. This security service enables the originator of a message to obtain corroboration that his message has been received and that EDIM Responsibility has been accepted, forwarded, or refused. This service may be provided using the Content Integrity check on message submission applied to the EDI Notification of the subject EDIM.

- Proof of Retrieval. This security service enables the MS administrator to obtain corroboration that a particular message has been retrieved from the EDI Message Store by the EDI User Agent. Implementation of this security service is a local issue which is useful between Message Transfer Agents within an Management Domain.

- Proof of Transfer. This security service enables a Message Transfer Agent or a Management Domain to obtain corroboration that a message has been transferred

(relayed) to another Message Transfer Agent within another domain. Implementation of this security service is also a local issue.

Non-repudiation of EDIM Responsibility services include:

- Non-repudiation of EDI Notification. This security service provides the Originator of a message with irrevocable proof that a message has been received, and that EDI responsibility has been accepted, forwarded, or refused.

- Non-repudiation of Retrieval. This security service provides the EDI Message Store administrator and the EDI User Agent with irrevocable proof that a message has been retrieved from the EDI Message Store by the EDI User Agent. Implementation of this security service is a local issue.

- Non-repudiation of Transfer. This security service provides a Message Transfer Agent or a Management Domain with irrevocable proof that a message has been transferred (relayed) to another MTA within another Domain. Implementation of this security service is a local issue.

- Non-repudiation of Content. This security service provides an EDI Messaging user with irrevocable proof of the authenticity and integrity of the content of the message. This security service may be provided in two ways, using a Notarization Mechanism or using the Non-repudiation of Origin security service applied to the subject message and the EDI Notification of the subject message, provided the EDI Notification includes irrevocable proof of the content of the subject message.

All these security services are available only if the MTS supports secure messaging. Many of the techniques employed rely on encryption mechanisms. The security services in the MHS allow for flexibility in the choice of algorithms. The problems with those standards are that they still use asymmetric encryption systems. The high technology provides symmetric encryption, as discussed in Chapter II. This technology will become the solution of next generation security mechanisms.

7. Security Model in CCITT X.501: The Directory - Models

a. Authorization Policy

Authorization policy includes the methods of:

(1) Specify Access Rights.

(2) Enforce Access Rights (access control).

(3) Maintain Access Rights.

b. Authentication Policy

Authentication policy includes the methods of:

(1) The identity of directory system agents and directory users.

(2) The identity of the origin of received information at the access point.

Actually, CCITT X.501 does not define a security policy, but makes a link to CCITT X.509 which defines authentication procedures. The Directory Access Protocol (DAP) and Directory System Protocol (DSP) may provide strong authentication of the initiator by the signing of the request, data integrity of the request by signing of the request, and strong authentication of the responder and data integrity of the result by the signing of the result. The DAP may provide simple authentication between a directory system agent and a directory user agent. The DSP may provide authentication between two directory system agents. Administrative authorities of applications making use of the directory may use their own security policy. The directory supports applications by holding authentication information (e.g., distinguished names, passwords, certificates) about communication entities. This is further described in CCITT X.509.

8. CCITT X.509:The Directory - Authentication Framework

a. Simple Authentication

Simple authentication is intended to provide local authorization based upon a distinguished name of a user, use a bilaterally agreed (optional) password and a bilateral understanding of this password within a single domain. Utilization of simple authentication

is primarily intended for local use only, i.e., for peer entity authentication between a directory user agent and a directory system agent or between a directory system agent and a directory system agent. Simple authentication may be achieved by several means: first, the transfer of the user's distinguished name and (optional) password to the recipient for evaluation. Secondly, the transfer of the user's distinguished name, password, and random number and/or a timestamp. Finally, the transfer of the protected information described above together with a random number and/or timestamp. All of these methods are protected by applying a one-way function. The procedure for achieving simple authentication is as following:

(1) An originating user A sends its distinguished name and password to a recipient user B;

(2) B sends the distinguished name and password of A to the directory, where the password is checked against the user password attribute within the directory entry for A;

(3) The directory confirms (or denies) to B that the credentials are valid;

(4) The success (or failure) of authentication may be conveyed to A.

CCITT X.509 also provides one-way functions and the timestamp for encryption and to prevent replay. [Ref. 41]

b. Strong Authentication

As mentioned in Chapter III, strong authentication is one of two mechanisms used that relies on the use of cryptographic techniques to protect the exchange of validating information. There is no mandated cryptosystem in this recommendation.

(1) The approach to strong authentication makes use of the properties of a family of cryptographic systems, known as PKCS (see Chapter III for details of PKCS, Public Key Cryptosystems). To achieve interoperability, cryptosystems must support both sides of the transaction.

Authentication relies on each user possessing a unique distinguished name. The allocation of distinguished names is the responsibility of the Naming Authorities. Each

user must trust that the Naming Authorities do not issue duplicate distinguished names. Each user is identified by secret key. A second user is able to determine if a communication partner possesses the secret key. This information is used to corroborate that the communication partner is, in fact, the user. The validity of this corroboration is dependent upon the secret key remaining confidential. In order for a user to determine that a communication partner is in possession of another user's secret key, that user must also be in possession of the second user's public key.

(2) Procedures of Strong Authentication. In general, a specific application determines the appropriate procedures to meet security policy of the application. There are three types of strong authentication.

- One-way authentication

This authentication method involves a single transfer of information from one user (A) intended for another (B). The following steps are involved:

i. A generates a non-repeating number, used to detect replay attacks and to prevent forgery.

ii. A sends the following message to B,

B -> A, A{tA, rA, B}

with a timestamp (tA) consisting of a generation time and expiration date. Alternatively, if data origin authentication of "sgnData" is to be provided by the DS:

B -> A, A{tA, rA, B, sgnData}

In cases, the information is to be conveyed that will subsequently be used as a secret key (this information is referred to as "encData", and the public key of B is referred to Bp):

B -> A, A, A{tA, rA, B, sgnData, Bp{encData}}

The use of "encData" as a secret key implies that it must be chosen carefully, e.g., to be a strong key for whatever cryptosystem is used, as indicated in the "sgnData" field of the token.

iii. B carries out the following actions:

- obtains a public key of A from B to A, checking that A's certificate has not expired;

- verifies the signature, and thus the integrity of the signed information;

- checks that B itself is the intended recipient;

- checks that the timestamp is current;

- optionally, checks that rA has not been replayed.

rA is valid until the expiration date indicated by tA, rA is always accompanied by a sequential part, indicating that A will not repeat the token during the timestamp tA and therefore that checking of the value of rA itself is not required.

It is reasonable for party B to store the sequential part together with timestamp tA together with the hashed part of the token during time range tA.

Other than the steps listed above, we need to know that the identity of A and the authentication token was generated by A; the identity of B, and that the authentication token actually was intended to be sent to B; and the integrity and "originality" of the authentication token being transferred.

- Two-way authentication.

The following steps are involved:

i. as for one-way authentication step i.

ii. as for one-way authentication step ii.

iii. as for one-way authentication step iii.

iv. B generates rB, a non-repeating number, which is similar to the usage of rA.

v. B sends the following authentication token to A,

B{tB, rB, A, rA}

with a timestamp (tB) consisting of the same feature data as tA. Alternatively, if data origin authentication of "sgnData" is to be provided by the DS:

B{tB, rB, A, rA, sgnData}

In cases where information is to be conveyed that will be subsequently used as a secret key (this information is referred to as "encData", and the public key of A referred to " Ap):

B{tB, rB, A, rA, sgnData, Ap[encData]}

The use of "encData" as a secret key implies that it must be chosen carefully, e.g., to be a strong key for whatever cryptosystem is being used, as indicated in the "sgnData" field of the token.

vi. A carries out the following actions:

- verifies the signature, and thus the integrity of the signed information:
- checks that A is the intended recipient:
- checks that the timestamp tB is the current one;
- optionally, checks that rB has not been replayed.
- Three-way authentication

The following steps are involved:

i. as for two-way authentication step i.

ii. as for two-way authentication step ii. Timestamp tA may be zero.

iii. as for two-way authentication step iii, except that the timestamp need not be checked.

iv. as for two-way authentication step iv.

v. as for two-way authentication step v. Timestamp tB may be zero.

vi. as for two-way authentication step vi. except that the timestamp need not be checked.

vii. A checks that the received rA is identical to the rA which was sent.

viii. A sends the following authentication token to B:

A{rB}

ix. B carries out the following actions:

- checks the signature and thus the integrity of the signed information:

- checks that the received rB is identical to the rB which was sent by B. [Ref. 42]

c. *Threats Protected Against By The Strong Authentication Method*

The strong authentication method described in the CCITT X.509 offers protection against the threats described in Chapter III. In addition, a range of potential threats are specific to the strong method itself. These are: [Ref. 43]

(1) *Compromise of the User's Secret Key.* One of the basic principles of the strong authentication is that the user's secret key remains secure. A number of partial methods are available for the user to hold his or her secret key in a manner that provides adequate security. The consequences of the compromise are limited to subversion of communication involving that user.

(2) *Compromise of the CA's Secret Key.* That the secret key of a CA remains secure is also a basic principle of strong authentication. Physical security and "need-to-know" methods apply. The consequences of the compromise are limited to subversion of communication involving any user certificate by that CA.

(3) *Misleading CA into Producing an Invalid Certificate.* The fact that CAs are off-line affords some protection. It is the responsibility of the CA to check that credentials are valid before creating a certificate. The consequence of the compromise is limited to subversion of the communication involving the user for whom the certificate was created, and anyone impacted by the invalid certificate.

(4) *Collusion Between a Rogue CA and User.* A collusive attack will defeat the method. This constitutes a betrayal of the trust placed in the CA. The consequence of a rogue CA is limited to subversion of communication involving any user certified by that CA.

(5) *Forging of a Certificate.* The strong authentication method protects against forging of a certificate by the CA's signature. This method depends upon the maintenance of the security of the CA's secret key.

(6) *Forging of a Token*. The strong authentication method protects against the forging of a token by the sender's signature. This method depends upon maintenance of the secrecy of the sender's secret key.

(7) *Replay of a Token*. One- and two- way authentication methods protect against the replay of a token by the inclusion of a timestamp in the token. The three-way method does this by checking the random numbers.

(8) *Attack on the Cryptographic System*. The likelihood of effective cryptoanalysts of the system is based on advances in computational number theory and leads to a predicted need for a greater key length.

d. Digital Signature of CCITT X.509

CCITT X.509 specifies DS for the signed tokens in the directory and is not a standard for DS in general. The mechanism of the CCITT X.509 digital signature [Ref. 44] is shown in Figure 3. The D and E block are the mechanism of decryption and encryption, respectively. The h block is a hash function.

(1) Information is signed by appending an enciphering summary of the information to it. The summary is produced by means of a one-way function, while the enciphering is carried out using the secret key of the signer.

(2) The recipient of signed information verifies the DS by:

- applying the one-way hash function to the information;

- comparing the result with that obtained by deciphering the DS using the public key of the signer.

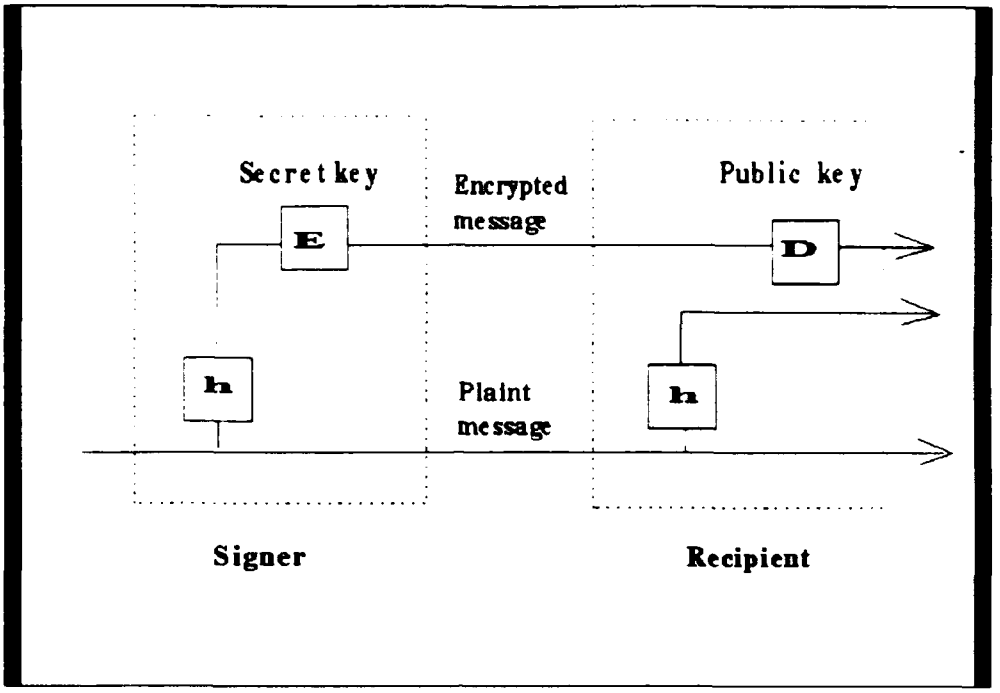

Figure 3. The CCITT X.509 Digital Signature

(3) This authentication framework does not mandate a single one-way hash function for use in signing. It is intended that the framework will be applicable to any suitable hash function, and will thus support changes to the methods used as a result of future advances in cryptography, mathematical techniques or computational capabilities. Two users wishing to authenticate must support the same hash function for the authentication to be performed correctly. Thus, within the context of a set of related applications, the choice of a single function will maximize the community of users able to authenticate and communicate securely. The signed information includes the hashing algorithm and the encryption algorithm used in computing the DS.

(4) The encipherment of data items described in CCITT X.509 are written by the ASN.1 MACRO for implementation, which include the data type resulting from applying a DS to the given datatype.

C. RECOMMENDATIONS OF USE DS ON EDI

A variety of security procedures exist that may be implemented to protect against the threats listed in Chapter IV. Unfortunately, some of these are impractical in a commercial environment. For example, to protect interchanges from interception, it is possible to use dedicated private cables or fiber optic cables. Actually, many companies still transmit information on public telephone line or even microwaves. These are very easy to intercept without detection. Currently, the only practical method of protection is the use of cryptographic techniques.

After analyzing the threats and vulnerabilities of EDI implementation, three of the security services listed in Chapter IV are relevant to EDI interchange security. These are encryption, authentication and key management. Chapter III stated that the state-of-art of digital signatures realizes the security tools needed to allow for secure EDI. The survey showed that almost all of the security mechanisms used for authentication, encryption, and key management involve the obsolete conventional private key cryptosystem.

As we learned in Chapter III, digital signatures based on public key cryptosystems provide authentication for verification and encryption for privacy and integrity. Security standards like ANSI X9 series are still using DES to encrypt messages and keys and IVs for distribution. As key management is the natural weakness of private key cryptosystems, DES cannot provide enough security for the interchange of EDI information. However, after adopting the public key based DS mechanisms, key distribution will no longer be a problem in the interchange of EDI information. The ANSI X12 standard series using DES to generate MAC may be replaced by DS to get rid of the key management weakness problems to increase interchange security. If the message digest mechanism is combined with DS, then the problems in message filtering and padding before transmission to the network will be solved (details were surveyed in Chapter III and IV). It will need a lot of work to revise the security concept of those standards from the conventional private key cryptosystem to the modern public key cryptosystem. Futhermore, interchange security cannot promise the complete security of EDI implementation.

D. SECURITY MANAGEMENT MODEL OF EDI

1. Introduction

The responsibility of management is to safeguard the assets of the organization. EDI's job is to transmit transactions affecting the assets of all trading partners. When an organization decides to implement EDI security standards to minimize EDI vulnerabilities and threats, additional management considerations must be included. Also, when addressing overall security in an EDI system as a network distributed system, it is necessary to integrate computer system security with communication security measures to protect the EDI information. Neither one can provide complete protection of critical EDI information in a distributed environment without integration. For instance, using an access control service to restrict users access to an EDI application software within a system or onto a network provides a higher level of security. Trusted computer system mechanisms are needed to ensure the enforcement of security controls and that correct operation of the security measures are maintained. Secure protocol standards are vital to the successful operation of security measures. Security mechanisms, like encryption algorithms, form an essential part of the overall solution. Harmonious integration of such security features forms the key ingredient in the development of an overall secure EDI system.

This security plan of building a EDI security model may be added to the implementation steps described in Chapter II. The security plan of the organization must be before the agreement with trading partners is completed.

2. Managerial Characteristics of EDI

The following characteristics of EDI might influence decisions in building the EDI security management model:

a. There are at least two interconnected trading partners that transmit documents on the networks. In some cases, there are many interconnected systems or networks transmitting sensitive and critical EDI documents.

b. There are many organizations involved in the development and acquisition of components (big hubs with a huge numbers of spokes).

c. The EDI platform uses a multilevel network. Computers or network nodes process different levels of classified information while users have different clearances and need-to-know restrictions.

d. The EDI software packages and EDI platforms are operated by people.

e. There is no overall security policy for EDI implementation. Those characteristics are the concerns of organizations who want to implement security on an EDI system.

3. Building EDI Security Management Model

For EDI environments with the characteristics listed above, the management process may be extremely complicated and resources intensive. A security model for the organization is built to protect business resources. Before we build a model, a determination of the EDI environment boundaries, a review of EDI implemented documents and identification of the overall EDI components should be delineated.

a. Set EDI Security Goals

Setting EDI security goals is not difficult for the organization which is already involved in EDP. The requirements of physical security, personal security, and data security are known to them. Adding EDI applications to their processes does not generally require much change, especially, if they already have telecommunication applications. Organizations without the experience of computer security need to set their security goals before they decide to implement EDI to protect their own assets. The security goals are as follows:

(1) To provide protection against threats to EDI.

(2) EDI document security must operate at the application level, which means end-to-end protection.

(3) EDI platforms have to follow the network multi-security levels.

(4) All security services should be based on existing and proven security standards.

(5) Do not change the existing EDI structure.

(6) Set the computer security model, if it doesn't have to use an EDP.

b. Determine Security Policy

We will determine the EDI system security policy in at least seven areas:

(1) *EDI System Security Objectives*. Identify primary and secondary security objectives. The primary objectives may include confidentiality, integrity, and availability.

(2) *Roles and Responsibilities*. Establish specific duties and tasks for all personnel with various parts of EDI system. This includes:

- Base level communication-computer security officers, and base computer security managers
- Network managers, and network security officers
- Computer system managers, and computer system security officers
- EDI software work team managers, and EDI software security officers
- Auditing managers
- Functional Business Managers
- Business managers and technical support from key trading partners
- EDI consultant and vendor security support representatives

(3) *Life Cycle Management*. Identify life cycle phases for EDI system implementation additions, deletions, and modifications, and especially for new EDI systems being developed to interface with their trading partners. Emphasize the need to continually monitor the EDI system to ensure that security measures remain effective after system changes.

(4) *EDI System Security Measure*. In addressing specific policies and procedures regarding:

- Physical security, which involves protection and survivability for personnel and equipment: protection against intentional human threats such as theft, sabotage, and espionage; and environment security.

- Procedural security, which protects against unintentional human threats such as accidental blunders, improper maintenance, etc.

- Personnel security involves clearance and need- to-know.

- Information security, which includes procedures for handling classified and sensitive unclassified information; magnetic remains; fraud, waste, and abuse, etc. and especially, business information, such as the bargaining price of the product or the volume of the orders.

- Communication security protects secure and sensitive EDI documents communication.

- Emanations security to prevent exploitation of electronic signal radiations.

- Operations security to identify, control, and protect evidence of the planning and execution of sensitive activities.

- Trusted systems security, which includes trusted computing base (TCB) classes, modes of operation, identification and authentication, discretionary and mandatory access control, object reuse, audit trails, labels, trusted paths, and documentation requirements.

- Hardware security, which involves nonvolatile strong media, peripheral security, maintenance activities, periods processing, and firmware.

- Software security, which involves evaluated and non-evaluated products; user-developed software; public domain software; software development, testing, and debugging; security software; job control language; configuration management; trusted software, data base management systems; maintenance activities; and malicious logic.

- Integrity measures such as device identification, message management, protocols, and integrity checks.

- Other security considerations such as interface policies and resources allocation policies.

(5) *Contingency and Emergency Plans*. Establish criteria for developing and testing contingency and emergency plans, especially policies for making and storing backups of the EDI systems and its background database.

(6) *Education, Training, and Awareness*. Identify policies and procedures for all aspects of security training, including initial and recurring training for all personnel, specialized training requirements for personnel in key security positions.

(7) *Incident and Vulnerability Reporting*. Establish policies and procedures for identifying and reporting incidents and vulnerabilities. Policy development must focus on specifically tailoring DoD guidance to the network. The security guide book uses abstract terms such as subjects, objects, groups of subjects, need-to-know, security labels, discretionary access controls, and mandatory access controls. The policy should describe what these abstract terms mean in the context of the EDI implementation procedure. The policy should address specific issues such as the range of security labels which the network must accommodate, specific objects having security labels, how security labels are determined for objects, and the various EDI participants, industries, and actions which are subject to discretionary or mandatory access controls.

c. Determining Priorities for EDI Security

A systematic approach should be utilized to determine appropriate EDI environment security measures. EDI security should not be addressed by one individual or a single organization. It is important to know the concerns and the needs of all of the organizations which are involving in the business of using EDI. The following steps provide a significant approach for implementation of a secure EDI.

(1) *Define EDI Configurations*. The first step in determining priorities for EDI security is to define all aspects and assets of EDI. The goal of this step is twofold: the first is to have a detailed EDI configuration that indicates interoperated hardware, the major

EDI software applications used, significant information processed on the EDI system, and how EDI documents flow through the platform; the second goal of this step is to evaluate each part of the EDI system. Assets may include any piece of hardware, software, application, data, etc. Assets then become areas of the EDI system that need to protected. Configurations include:

- EDI Software Packages. These include EDI software, EDI platform network management software, EDI background database and its management systems, and some EDI implementation project management systems, etc.

- EDI Platform Services. These include data transmission, the ability to link E-mail, Fax and EDI, the ability of international communication, the ability of integration and other platform related services.

- EDI System Environment Hardwares. These include the bar code system, computers, networks, and other computer system related peripheral machines.

After the EDI configuration is completed, and the assets are determined and valued, there should be a reasonably correct view of what the implementation of EDI system consists of, and what areas of the EDI environment need protection.

(2) *Determine Risk*. The goal of determining the risk is to determine the level of current security for the EDI system. To begin the process of determining the risk, consider the threats to the EDI system and the possible vulnerabilities of the EDI system that could be exploited by those threats. Risk analysis may uncover some vulnerabilities that can be corrected by improving EDI management and operational controls. These improved controls will usually reduce the risk of the threats by some degree until such time that more thorough improvements may be planned and implemented. Attention should be paid to existing EDI system security controls to determine if adequate protections being provided to prevent vulnerabilities. These controls may be technical, procedural, etc. As specific threats and related vulnerabilities are identified, a risk value needs to be associated with the threat. The risk can be evaluated by some functions, such as probabilities of the threats occurring and calculating the expected value of those threats.

To ensure that all identified risks and vulnerabilities are addressed, construct a list by prioritizing the threats based on its risk. This list of potential threats, vulnerabilities and related risks will allow an assessment of the current security situation for the EDI system. Areas where there is adequate protection do not surface as contributing to the risk of the EDI.

(3) *Select Security Services and Security Mechanisms*. This step examines security services and mechanisms to reduce defined risks. Security services are the sum of the mechanisms, procedures, that are implemented on the EDI system to provide protection. This step is broken into four tasks.

- Task-1: Consider the security services provided in the EDI security architecture.

- Task-2: After the needed security services are determined, consider the list of security mechanisms for each service.

- Task-3: The decision to use a certain mechanism will largely depend on the cost of the mechanism. The cost estimation of each mechanism should be made at this task.

- Task-4: Update the list of threats and mechanisms each time an examination of the implementation of the EDI system occurs.

(4) *Develop Priorities of EDI Security*. The goal of this step is to produce a prioritized list of security services and mechanisms that should be implemented to reduce perceived EDI implementation risk and to protect the EDI system adequately. The process involves the comparison of various threats and the influenced risk. Sometimes, an evaluation model for threats should be built including the cost-effective factors to the organization. After priorities have been set by the EDI implementing project team, the next step is implemented.

(5) *Implement and Test Security Mechanisms*. Just as the mechanisms that constitute the priorities for EDI system security were chosen using a systematic approach, so should the implementation of those mechanisms proceed in the same manner. The goal of this phase is to ensure that the security mechanisms are implemented correctly, are

compatible with other EDI functionalities and security mechanisms, and that the security mechanisms meet the requirements of providing adequate security. This step begins by developing a plan to implement the mechanisms. This plan should consider factors such as the timeliness required to reduce risk, available funding, etc. A testing schedule should show how each mechanism interacts with other mechanisms. The expected results of the interaction should be detailed. It should be recognized that not only it is important that the mechanism perform functionally as expected and provide expected protections, but that the mechanism does not contribute to the risk of the EDI system through a conflict with some other mechanisms or functionalities. After all mechanisms are implemented, tested and found acceptable, the list of priorities for EDI system security should be reexamined. All these phases need to be repeated again until the end of the life cycle.

d. Construct Security Architecture.

Constructing the security architecture, requires the construction of a general architecture of EDI, the subset security architecture, and identification of security architecture issues.

(1) *Construct the General EDI Architecture.* From high-level perspectives, it is possible to construct many different general models of EDI system.

- Mission architecture. Divide the primary mission into mission components, then determine which centers and information processing centers support each component. Then, determine which elements of the EDI system support the centers to accomplish their missions. If the EDI system has a command and control structure, identify the flow of mission command and control from the highest agencies to the lowest agencies contributing to the EDI system.

- Communication-computer architecture. Identify EDI data flow from individual sources, to processing entities such as correlation centers, to the EDI system users.

(2) *Construct the Security Architecture*. The security architecture must be very detailed. It must identify general security facts, assess the EDI system's ability to meet primary security objectives, and describe protection mechanisms used by the EDI system. General security facts about the entire EDI system must be identified.

- Highest and lowest classifications processed, or types of sensitive unclassified material being processed.

- Minimum and maximum user clearances and restrictions.

- All security modes of operation used through the EDI system. Second, assess how the EDI system achieves these three security objectives:

- How does the EDI system protect secure data from unauthorized disclosure (confidentiality)?

- How does the EDI system ensure system integrity (the ability to function unimpaired, free from deliberate or inadvertent unauthorized manipulation) and data integrity (i.e., does data correctly represents information, and do authorized users and EDI system software processors handle and manipulate the data properly)?

- How does the EDI system provide assurance of service? Specifically identify and describe mechanisms which protect against common threats and vulnerabilities to the EDI system. Be sure to cover all the disciplines of physical security, procedural security, personnel security, information security, communication security, trusted system security, hardware and software security, integrity mechanisms, and other security measures.

(3) *Identify and document security architecture*. Identify and document security architectureissues uncovered during network investigation and analysis. Make recommendations to resolve these issues.

e. Management Activities

Except for the cryptography technology, management activity is the most important part of the security management model. The structure of security management

programs may vary substantially from one organization to another. We need to examine all the required security services and mechanisms that can help us solve the possible security vulnerabilities and threats as described in Chapter IV. Five management activities that address these services follows:

(1) *Authentication Management.* [Ref.45]

- Associating authentication information, for example, passwords, keys, identities, and tokens, to system entities.

- Updating, modifying and revoking authentication information.

- Assisting in the authentication verification process.

- Choosing the authentication mechanisms to be employed.

- Interaction with other security services, such as access control.

(2) *Access Control Management.*

- Establishing and associating access control information to system entities, such as access control lists to files, roles and capabilities of users.

- Updating, modifying and revoking access control information and rules.

- Enforcing access control rules.

- Denying or granting access based on specific access control rules.

- Interaction with other security services.

(3) *Key Management.* Key management issues have been stated in Chapter III and IV. The specific key management standard for EDI (ANSI X12.42) is based on ANSI X9.17. This standard contains a large number of options, many of which are not applicable to an EDI environment. This management will be dependent on the type of encryption mechanisms employed and the associated key distribution protocols. The activities involve as following:

- Generation of suitable keys at intervals commensurate with the level of security required.

- Determination in accordance with access control requirements of which entities should receive a copy of each key.

- Distribution of the keys in a secure manner.

- Decisions involving updating of keys, deletion or removal of keys and when they become obsolete.

(4) *Audit Management.*

- The definition and selection of security relevant events to be logged and/or collected.

- The enabling and disabling of audit trail logging of selected events.

- The analysis of audit trails.

- The preparation of security audit reports.

(5) *Risk Management.* [Ref. 46]

- Identify assets.

- Determine vulnerabilities.

- Estimate likelihood of exploitation.

- Compute expected annual cost.

- Survey applicable controls and their costs.

- Project annual saving of control.

f. Management Considerations

Implementing this security management model may require managerial considerations as follows [Ref. 47]:

(1) *Separation of Control.* This consideration is based on a concept that is usually employed for a variety of reasons. From a practical point of view, it is helpful to decompose any problem or activity into a set of independent tasks which can be assigned to individuals that are trained to handle them. When automation is desired, separation of control will apply to limit the responsibility of action to well defined areas so that easily understood rules of action may be programmed into a machine to effect the desired results in small increments. This process also provides multiple check points to evaluate each node of the procedure, especially, in automation systems.

Since this work addresses the problem of security in an EDI environment, separation of control is valued most for its ability to isolate the responsibility for action. Thus, it is possible to construct systems where errors can be quickly isolated to particular individuals. To be most effective, it is essential that those working with the system understand how easily errors can be attributed to their actions. In order to foster an environment of cooperation, it must be communicated that their minor errors will not be held against them. Separation of control is therefore necessary in a secure EDI system. The responsible manager will design and implement EDI systems independently of any internal application. A security officer is required who is not under the authority of anyone in computer operations or programming, and the internal auditor must be familiar with the entire of EDI system and its sequences.

(2) *Internal Controls*. The objectives of internal control are to protect the EDI environment to assure that all transactions are properly processed. Generally, internal controls include general, administration, and application controls. [Ref. 52] General and administrative controls concern proper implementation, maintenance, and operating procedures. These controls provide the standards and guidelines for a variety of applications. Application controls are concerned about the accuracy of data, validity of data, appropriate maintenance, and that process results fulfill demands of management. Simply speaking, the application controls are data-oriented and specific task-oriented, and general controls are procedure-oriented. The structure of internal controls is very influential to the process of auditing.

(3) *Auditing*. Auditing serves an organization's security by making the detection of security lapses more likely. The probability of detection should encourage an organizations' employees to enforce security procedures. While a certain familiarity with the technology is important, normal auditing concerns remain the same. Control problems are likely to surface from a lack of a systematic flow of work, poorly written procedures, untrained staff, no penalty for noncompliance with organizational policies, and lack of testing and quality assurance.

(4) *Access Control.* It is important that responsibility for security at any given time and place be assigned to one particular security officer who is responsible for all employee access at the transmission site. The National Computer Security Center (NCSC) has been established in the NSA to evaluate the quality of trust of specific commercial operating systems.

(5) *Records Retention.* For the threats of repudiation, it is essential to maintain the security of the signatures and the data received as long as there is any threat of repudiation by the receiver. For any security environment, it is important to maintain business records so long as questions can be raised concerning the procedures used. When signatures expire, they should be achieved if they are used to provide confidentiality or integrity for stored data (which means encryption or authentication are used). This is especially important for encrypted stored data, for without the signature, the data will be useless.

(6) *Security Entities.* Security entities are the processes that perform the functions of encryption and authentication as defined in the ANSI X12 standards. It is important that the physical and logical location of these entities be selected to satisfy the user's business requirements. Since the security entities are responsible for all of the secret information shared between trading partners, the security entities must be physically secured from the threat of penetration by the outside agents. Logically, the security entities must be placed in the processing chain so that the data is protected early enough to meet the security needs, but not prior to the physical generation of all the data requiring protected. In both views, it is important to create a security perimeter around the sensitive data internal to any business partner so that it will be transparent that the current business systems need protection. The security entity acts as a gateway to the protected interior of the perimeter and the outside where the interchange will require the protection of the ANSI X12 security standards.

(7) *Interoperability.* Interoperability is one of the success factors of EDI. It is also a very important factor in security management. Two entities may successfully

communicate only if their implementations are interoperable. They need to communicate using protocols and procedures which are compatible with one another. Also, if the encryption and decryption are schemes different, they will never work together. ANSI X12.42 and ANSI X12.58 provide a number of options for the user to tailor the cryptographic processes to a particular application. Security managers need to make application decisions, and state the procedures in the agreement.

(8)*Version Control.* Version control is a part the of interoperability problem which deals with two party support of the same version. It is important because the recovery policy must be very specific when errors occur. Originally ANSI X12 specifications called for support of the currently standardized version plus the most recent version. This may still be adequate for the active EDI transaction database, but given the abilities of the current database system, it hardly seems necessary to limit the capability of implementation. Each version is only a set of tables and procedures that the computer is to follow. Perhaps individual organizations should consider the possibility of supporting prior versions, especially in those organizations where a wide ranging or loosely knit community will be interchanging data.

(9) *Security Network.* Chapter 3 discussed techniques of cryptosystem for securing the information. Two techniques may be provided for network security. One technique to secure the network is end-to-end data encryption, which encodes information and places it in an electronic "envelope". That envelope cannot be forced open until it reaches its destination. In fact, end-to-end data encryption is sufficiently secure to be used for electronic fund transfers and other highly sensitive electronic transactions. Another encryption technique involves installing RSA's public key technology. However, encryption doesn't completely prevent unauthorized users from entering parts of the system where they don't belong. To prevent such trespassing, many security experts recommend the use of a trusted server. How the trusted system implements such controls depends on whether it uses the techniques of discretionary access control (DAC) or mandatory access control (MAC). With DAC, individual data owners can decide who can share their

information. The head of accounting, for example, could choose to let the customer service chief see the files in the department's directory. MAC, on the other hand, lets network administrators impose security controls on their own. It does this by allowing the administrator to assign security levels for every user, and security levels for every accessible file. Users can be assigned security rights based on such criteria as rank, position, written agreement, and the results of background investigations. Files can be assigned classifications such as top-secret, confidential, or public. Where both are in place, MAC has precedence over DAC. This means that the same customer service chief might be allowed to read only confidential or public data, but not files classified as top-secret. Any time he attempts to read that top-secret accounting data, the trusted server will automatically shut down and alert the network administrator of the attempt. Security of network is essential to the EDI implementation.

VI. CONCLUSION

Though EDI is not a new technology, it is becoming necessary for business transactions. EDI applications can only automate some transactions, as a successful company-wide implementation of EDI technology involves detailed planning. Good business planning requires compatible and well-managed training and support. When reviewing the EDI needs of a business, the EDI system manager must recommend the protocols and software that are best suited to the corporate mission. An overview and references of how to plan the implementation of EDI were discussed in Chapter II.

Users are seeking support from vendors when combining EDI and MHS technologies on a single network. Such a move offers users better security and tracking than is available with existing EDI services. Large companies, interested in cutting costs of maintaining separate networks for electronic business documents and electronic mail, also want a single standards-based messaging network. The X.400 messaging protocol and its X.435 EDI component may meet these demands. The use of OSI framework still requires the support of translation software, transport services, and value-added network gateways. Without an OSI operating environment, one may use the TCP/IP platform to run its EDI applications and these products do exist.

The future of EDI lies in the direction of work group computing and the possible large volume EDI processes. More realistic costs would be exposed in an integration impact assessment that analyzes expected payback, the number of trading partners, their level of preparedness, and the need for integration switches to other applications.

Unfortunately, the security of EDI has not been solved, although it is a well known issue. Chapter III surveyed digital signatures and in Chapter IV, we discussed security threats and their solutions. Chapter V points out the weaknesses of EDI security. The ANSI X12.58 EDI Cryptographic Security Structure Standard and X12.42 Cryptographic Service Messages provide the most important security concepts, methods of encryption and authentication for EDI systems. Security standards are associated with EDI standards,

though they are not mandatory. All EDI standards recommending DES are known to have difficulties in key management. Key management problems may be solved by DS, based on public key cryptography.

Encrypting an entire document is a slow process when tight security is maintained. If one uses RSA for signing a document by encrypting the message digest we will get shorter processing time than if encrypting the entire EDI document. A compromise between these two extremes is to use a private key system, such as DES, to encrypt the document and then continue the digest signing by RSA. The best method will depend on the requirements of the business transaction.

The provision of DS mechanisms does not guarantee a secure EDI system. In addition, the security management of EDI may be addressed by building an EDI security management model. The security management model presented in Chapter V sets security goals, determines security policies and priorities, and constructs security architecture. Management activities such as key management, access control management, audit management, and risk management are also important considerations when implementing the security of an EDI system.

LIST OF REFERENCES

1. Krivda, C. D., "Electrifying Space," *MIDRANGE Systems*, V. 5, No. 18, p. 25., September 29, 1992.

2. Ibid, p. 24.

3. McKendrick, E. J., "A Pioneer of EDI," *MIDRANGE Systems*, V. 5, No. 14, p. 18, July 1992.

4. Zorfass, P. and Michel, C., "White Paper of EDI," *Computer World*, V. 26, No. 40, October 5, 1992.

5. Shell, N., "Bar Code Break Out," *Datamation*, V. 38, No. 7, p. 71, April 1, 1992.

6. Cashin, J., "Business Transation Take Electronic Route," *Software Magazine*, V. 11, No. 15, pp. 81-87, December 1991.

7. International Telecommunication Union, CCITT, *Recommendation Summary: MHS-EDI Messaging System*, ITU, 1991.

8. Hinge, K. C., "Electronic Data Interchange," AMA Membership Publication Division, p. 12, 1988.

9. Ibid, p. 20.

10. McKendrick, E. J., "A Pioneer of EDI," *MIDRANGE Systems*, V. 5, No. 14, p. 19, July 1992.

11. Zorfass, P. and Michel C., "White Paper of EDI," *Computer World*, V. 26, No. 40, October 5, 1992.

12. Rivest, R. L., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public Key Cryptosystem," *Communication of the ACM*, V. 21, No. 2, p. 120, February 1978.

13. NIST, "NIST Proposed Digital Signature Standard," *Communication of the ACM*, V. 35, No. 7, p. 36, July 1992.

14. Lipton, S. M. and Matyas, M. S., "Making the Digital Signature Legal and Safeguarded," *Data Communication*, V. 7, No. 2, p. 253, February 1978.

15. Diffie, W. and Hellman, M., "New Directions in Cryptography," *IEEE Press*, V. 22, 1976.

16. Lipton, S. M. and Matyas, M. S., "Making the Digital Signature Legal and Safeguarded," *Data Communication*, V. 7, No. 2, pp. 230-243, February 1978.

17. Diffie, W., "The First Ten Years of Public Key Cryptography," *Proceedings of IEEE*, V. 76, No. 5, pp. 133-135, 1988.

18. El Gamal, T., "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Press*, V. 31, 1985.

19. Schnorr, C. P., *Efficient Identification and Signatures for Smart Cards*, Springer, Germany, 1990.

20. Goldwasser, S., Micali, S., and Rivest, R. L., "A Paradoxical Solution to the Signature Problem," *Foundation Computer Science*, October 1984.

21. Rabin, M. O., *Digital Signature and Public Key Functions as Intractable as Fractorization*, MIT Lab for Computer Science Technical Report, Boston, Massachusetts, January 1979.

22. Fahn, C., *Answers to Frequently Asked Questions About Today's Cryptography*, RSA Laboratories Report, p. 6, September 14, 1992.

23. Ibid, p. 38.

24. Beth, T., Frisch, M., and Simmons, G. J., "Public Key Cryptography," *E.I.S.S. Press*, p. 35, July 1991.

25. Fahn, C., *Answers to Frequently Asked Questions About Today's Cryptography*, RSA Laboratories Report, p. 38, September 14, 1992.

26. Ibid, p. 36.

27. RSA Data Security Inc., *Product Report*, p. 5, 1991.

28. Ibid, p. 7.

29. Rivest, R. L., "The MD4 Message Digest Algorithm," paper obtained from Internet public network, February 17, 1990.

30. Mitchell, C. J., Piper, F., and Wild, P., *Digital Signature*, IEEE Press, p. 261, 1992.

31. Ibid, p. 263.

32. Ibid, p. 270.

33. Ibid, pp. 273-274.

34. NIST, "NIST Proposed Digital Signature Standard," *Communication of the ACM*, V. 35, No. 7, pp. 36-37, July 1992.

35. International Communication Union, CCITT, *Recommendation X.402: Message Handling System Overall Architecture*, ITU, p. 141, 1988.

36. Rusell, R. and Gangemi, G. T., *Computer Security Basic*, O'Reilly & Association, Inc., pp. 204-205, December 1991.

37. Data Interchange Standard Association, *Control Standards and Transaction Set Tables*, V. 1, pp. 70-72, DISA, February 1991.

38. International Communication Union, CCITT, *Recommendation X.400: Message Handling System and Service Overview*, ITU, pp. 28-29, 1988.

39. International Communication Union, CCITT, *Recommendation X.402: Message Handling System Overall Architecture*, ITU, pp. 96-102, 1988.

40. International Communication Union, CCITT, *Recommendation X.435: Message Handling System -Electronic Data Interchange Messaging System*, ITU, pp. 112-114, March 1991.

41. International Communication Union, CCITT, *Recommendation X.509: The Directory-Authentication Framework*, ITU, p. 55, 1988.

42. Ibid, pp. 64-67.

43. Ibid, p. 77.

44. Ibid, p. 62.

45. Varadharajan, V., "A Security Reference Model for Distributed Object System and Its Application," *NIST/NCSC*, p. 602, October 1992.

46. Pfleeger, C. P., *Security in Computing*, Prentice Hall, p. 87, 1989.

47. Jones, T. C., *Secure EDI*, Data Interchange Standard Association, 1991.

GLOSSARY OF TERMS

ANSI. American National Standard Institute.

ANSI ASC. Accredited Standards Committee of American National Standard Institute.

ASN.1. Abstract Syntax Notation One.

AU. Access unit.

BAUDOT. 5-Bit per character information coding scheme.

CA. Certification authority.

CCITT. The international telegraph and telephone consultative committee.

CBC. Cipher block chaining.

CFB-1. 1-Bit cipher feedback.

CFB-8. 8-Bit cipher feedback.

CRL. Certificate revocation list.

CSM. Cryptographic service message.

DAC. Discretionary access control.

DAP. Directory access protocol.

DES. Data Encryption Standard.

DS. Digital signature.

DSA. Digital signature algorithm.

DSP. Directory system protocol.

DSS. Digital Signature Standard.

ECB. Electronic code book.

EDI. Electronic data interchange.

EDIFACT. Electronic Data Interchange for Administration, Commerce and Transfer.

EDIM. Electronic data interchange message.

EDIN. Electronic data interchange notifications.

EFT. Electronic funds transfer.

FG. Function group.

FIPS. Federal information processing stanc

ISDN. Integrated services digital network.

ISO. International Standards Organization.

ITS. Initial text sequence.

IV. Initialization vector.

JIT. Just in time.

KD. Automatically distributed data key.

KDC. Key distribution center.

KK. Automatically distributed key encrypting key.

KKM. Manually distributed key encrypting key.

KMF. Key management facility.

KTC. Key translation center.

MD. Message digest.

MHS. Message handling system.

MS. Message store.

MTA. Message transfer agent.

MTS. Message transfer system.

NCSC. National Computer Security Center.

NIST. National Institute of Standards and Technology.

NSA. National Security Agent.

OSI. Open system interconnection.

PEM. Privacy enhancement mail.

PKCS. Public Key Cryptography Standard.

POS. Point of sale.

QR. Quick response.

RF. Radio frequency.

RSA. An algorithm used for creating digital signatures invented by R. Rivest, A. Shamir.

and L.Ademan and named after them.MIT professors

SHS. Secure Hash Standard.

SxE. ANSI X12 security trailer segment

SxS. ANSI X12 security header segment.

SWIFT. Society for Worldwide Interbank Financial Telecommunication.

TCB. Trusted computing base.

TCP/IP. Transmission Control Protocol/Internet Protocol.

TDCC. Transportation Data Coordinating Committee.

TDS. Tamper detection system.

TRM. Tamper-Resistant module.

TS. Transaction Set.

UA. User Agent.

UCC. Uniform Commercial Code.

UCS. Uniform Code Standard.

UN/ECE. Economic Commission for Europe of United Nation.

VAN. Value added network.

WINS. Warehouse Information Network Standard.

INITIAL DISTRIBUTION LIST

		No. of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Chairman, Code AS Administrative Sciences Department Naval Postgraduate School Monterey, California 93943-5000	2
4.	Professor Jon T. Butler, Code EC/Bu Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
5.	Professor Roger Evered, Code AS/Ev Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor Chyan Yang, Code EC/Ya Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
7.	Navy Information Center P.O. 90151-7 Taipei, Taiwan, Republic of China	1
8.	Computer Education Center National Defense Management College Chung Ho, Taipei, Taiwan, Republic of China	1

9. Library 1
National Defense Management College
Chung Ho, Taipei, Taiwan, Republic of China

10. Library 1
Naval Academy
Tsuoin, Kaushung, Taiwan, Republic of China

11. Hua-Fu Pao 2
#61 Yun-Hua St. SHIM-DAN City
Taipei, Taiwan, Republic of China 231