12 Information Security Management

Opening Vignette: The World Trade Center Disaster and Recovery Planning

On September 11, 2001, two airplanes flew into the World Trade Center (WTC) killing more than 2000 people. All WTC offices were destroyed and many nearby buildings were badly damaged and immediately evacuated. Beyond the loss of human lives[1], major services were disrupted. Phone lines along the east cost of the United States were jammed due to the obvious increased phone activity. In addition, major telecommunication providers such as AT&T and Verizon lost major portions of services because their major switching centers and computer systems were located in the areas near to the WTC. This impacted several major clients, including Lufthansa Airlines which lost telephone services for its sales offices in downtown Manhattan. Lufthansa had chosen AT&T as its primary and Verizon as a backup provider. With both facilities impacted, Lufthansa was left without a telephone service for almost a week.

Many companies that relied on the Internet to conduct business were not severely impacted. In fact, the Internet became a viable alternate vehicle for communications in that disastrous week. In my own office in New Jersey that afternoon, we could not get the news from TVs (no TVs were available in the offices), so we all visited the news sites from CNN, FOX and others to understand what was going on. Internet telephony and email became the primary source of communicating with family and friends to let them know that we were OK.

Merrill Lynch had over 9,000 employees working at the WTC and the nearby World Financial Center. Most were unharmed and were relocated to other places of work quickly and successfully. Merrill resumed its business later in the same day and did not suffer as much as others. The main reason was that it had redundant telecommunications capabilities and a good disaster recovery plan. Merrill had rehearsed the plan four months earlier, so it was better prepared for a disaster than others. The plan included priorities for business activities, so that high priority activities could be brought online quicker. It also included detailed procedures for restoring critical applications with procedures that included necessary technologies, personnel, and facilities for a quick restoration in case of a disaster. Logistics were also in place for transportation of personnel and equipment with provisions for housing and feeding employees for up to 8 weeks. This disaster recovery plan went into action within minutes after the incident and Merrill was operational later that day.

[1] I was in the area shortly afterwards and have not been able to forget it.

Source: Laudon and Laudon, "Management Information Systems", 8th Edition, Prentice Hall, 2003.

Contents

-

12.1 Introduction

Increased reliance of enterprises on applications and the IT infrastructure (networks, computing platforms, middleware services) to support these applications is creating new security and intrusion threats. Comprehensive security architectures are needed that protect the corporate IT and physical assets by employing the latest security technologies to respond to external factors and organizational requirements. This chapter concentrates on the big picture (the key players and how they interrelate with each other to address corporate security issues). The chapter starts with a few short case studies to illustrate the internal security threats in Section 12.2 and Section 12.3 defines basic security terms and views and gives an overview of security at several levels – from networks to applications.

Sections 12.4 through 12.15 are the core of this chapter – it develops an architectural view that shows how the various corporate assets can be protected by using a combination of technical and organizational approaches. Figure 12-1 presents the following architectural components of such a view and is the foundation of this chapter.

- **Security requirements** to protect the IT assets are established as the first step in building a security architecture. These requirements must be able to take into account the external factors (realities of life at national or international levels) that drive security initiatives in

enterprises as well as organizational requirements that are different for different organizations. See Section 12.4

- **A management approach** is needed to develop organizational policies, roles, responsibilities, and training programs before choosing the security technologies (this issue is ignored by several academics). This approach must also include how the risks of security exposures will be managed and how the company can survive different types of attacks. Section 12.5 reviews these considerations.
- **A set of security techniques** are employed to satisfy the identified requirements subject to the overall management approach. The security techniques and approaches involve cryptographic techniques such as symmetric and asymmetric key cryptography, digital certificates and public key infrastructures (PKI) that apply at different levels of a system (from networks to applications). Research developments in highly adaptive systems that reconfigure themselves to respond to assault levels and the impact of standards bodies also need to be considered. Sections 12.6 through 12.15 discuss these technologies.
- **A methodology** ties the different technical and organizational pieces together into a series of procedural steps. This methodology, discussed in Section 12.8, explains steps such as how to identify IT and other assets (buildings, people) to be protected, how to conduct risk analysis through attack trees, and develop appropriate countermeasures to mitigate risks.

The chapter concludes by discussing a detailed case study that shows how the security methodology can be used to secure a financial institution (Section 12.9). Additional short case studies are also provided to illustrate different aspects of IT security (Section 12.10). The emphasis of these case studies is not on technologies but is instead on management approaches.

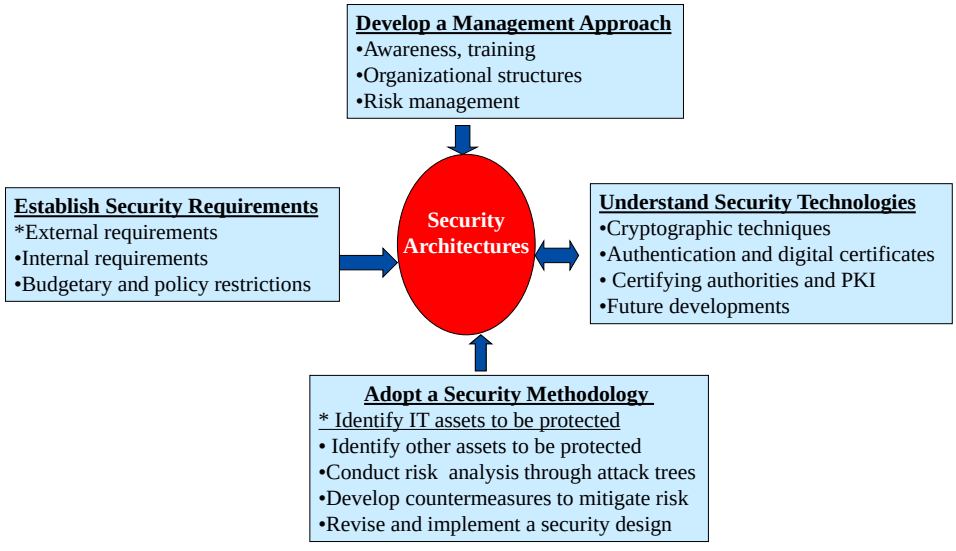


Figure 12-1: Security Architectures

Key Points

- It is important to develop an architectural view that shows how the various corporate assets can be protected by using a combination of technical and organizational approaches
- Security requirements must be able to take into account the external factors (national and international) as well as specific organizational requirements.
- A sound security management approach is needed with appropriate organizational structures, policies, roles, responsibilities, training programs and risk management.
- Several security technologies are used to secure systems. These include:
- User logon and password is one of the oldest and still most commonly used technologies.
- Encryption is another technology that has been used for a number of years to mask the messages so that the adversaries cannot see/modify the messages. Encryption is generally discussed in terms of two approaches: a) Private key (symmetric) encryption schemes in which the same algorithm is used by the sender to encrypt the message and the receiver to decrypt it, and b) Public key (asymmetric) system in which the encryption algorithm E and the decryption algorithm D are different - hence the name "asymmetric".
- Digital signature is used to authenticate the source of a message. It is essentially the same as a public key system except that the order in which the keys are applied is reversed.
- Message digesting is used to make sure that a certain message was not changed along the way between the sender and the receiver.
- A digital certificate binds an entity's identification to its public key and is issued by the Certification Authority.
- A systematic methodology is needed to tie the different technical and organizational pieces together into a series of procedural steps.
- Encryption can be used as an authentication mechanism.

The Agenda
- Security Requirements and Management
- Basic Cryptography
- Authentication, Authorization and PKI
- Security Methodology

12.2 Short Case Studies – Protecting the IT Assets from Internal Attacks [2]

The following cases are concerned with securing the environment within a company, i.e., how to secure the systems from damage by the insiders, not the outsiders.

12.2.1 Global Retailer

Company: Global retailer X (because the company has a policy of not talking to the press, the CIO agreed to share all the following information on condition of anonymity).

Place: primary headquarters of the company outside New York City.

Description of the security problem: Hundreds of computers were infected with a stealth virus not recognized by the latest antivirus software. Later, the virus was identified as ***Demiurg***, the stealth virus that spreads through Microsoft Excel spreadsheets. When a user opens an infected spreadsheet, the virus infects the Windows Kernel32.dll file, a fundamental part of the operating system. When the computer is rebooted with the infected Kernel32.dll file, the virus spreads to executables and batch files, corrupting so many files that the computer eventually stops working.

What was at risk: the entire company network, in this case the existence of the company. Total estimated damage of lost sales was more than $250,000.

Problem categorization: Management issue. This case illustrates the importance and key role of CIO in organizing the staff and solving the problem. However, it shows also that little was done before the accident to prevent a virus intrusion into the system. The only precaution was that the company was running the latest version of McAfee's antivirus software. The company did not seem to have a strict policy regarding opening files and e-mails from the unknown respondents.

Solution approach: Time in cases like this one is the most important factor. Because IT staff became alert when the significant damage was already done, they tried to shut down the network as quickly as possible: through intercom orders, by posting flyers and by physically delivering the news. It took less than two hours to have the network shut down (more than 400 PC users; also, no remote access for mobile users, no connection to offices in other countries, and no communication with stores, which could still ring sales and process credit card transactions but could not look up customer data or inventory at other locations).

The FBI was notified.

---

[2] These case studies were collected by Christopher Freiler and Viktoryia Petrashova, students at Fordham Graduate School of Business

During the following 4 days the virus was identified. Then, with the help of a simple program, it was prevented from being active. Finally the system was disinfected. McAfee was notified during the process to provide any help in the problem's solution.

12.2.2 FBI

Company: Federal Bureau of Investigation (FBI).

Place: Alexandria, Virginia.

Description of the security problem: A career FBI agent with significant experience and access to FBI IT systems was charged with spying for Russia since 1985, in what FBI Director Louis Freeh has called the worst case of insider espionage in bureau history. The accused person, an expert in counterintelligence methods at the FBI, was assigned to the New York Field Office's intelligence division in 1979 to help establish the FBI's automated counterintelligence database in that office. Investigators characterized him as having a "high degree of computer technology expertise." Although the accused was arrested while dropping off classified paper documents for his Russian handlers, he made extensive use of computer media, such as encrypted floppy disks, removable storage devices and a Palm II handheld computer, to communicate with Russian intelligence officers, according to the affidavit. In fact, he provided as many as 26 encrypted floppy disks during the course of his espionage activities.

What was at risk: national defense system.

Problem categorization: network and software security. A person or a group responsible for the security must be able to identify and understand the network and system intrusions. Also, artificial intelligence-enabled security software that use profiles can tip administrators off to "anomalous activity" on the network.

Solution approach: We can only guess about what exactly has been done by FBI security to find the threat. What is known for sure is that the FBI keeps records on what sites every employee is using and also on his/her everyday activities; when it's needed, that would lead to a special investigation. Also, in addition to these regulations, the order has been issued that a special panel be formed to review all FBI processes and systems and to study the issue of insider abuse.

12.2.3 Verizon

Company: Verizon Communications (a telecommunications company with subsidiaries that provide local telephone services in the region stretching from Maine to Virginia. At the time of the case, the company's name was GTE Corp., before the merger with Bell Atlantic Corp.

Place: Tampa, Florida.

Description of the security problem: An employee used his ability to gain access to GTE's secure computers at about 3 a.m. on May 15, 2001. Once he had access, he began to erase data contained in the computers and entered a command that prevented anyone from stopping the destruction process. Other IT workers at the GTE facility could not stop the self-destruction of the material once it had started.

What was at risk: customer database and confidential information that would be difficult (and costly) to recover. Total estimated damage: $200,000.

Problem categorization: Procedures. The accused had access to a secure area, and he, according to the record, severely abused those privileges. Therefore, the problem in this case can be identified as the problem of managing people within the security structure. It is also important to assign the right level of responsibilities, and enforcing them so that the lowest possible damage is incurred when used against the company. The challenge is to keep everybody happy simultaneously.

Solution approach: Obviously, there should be back up for the database information. Procedures can be implemented to secure information in the sensitive databases from deleting/transporting by requiring signoffs from multiple people. The deleted information can also be logged as part of the deletion procedure (most database managers support such logs) and stored somewhere for later use, if needed.

12.2.4 Ethical and Organizational Issues in these Cases

All three cases involve some ethical and organizational issues. In the Global Retailer virus case, the CIO questioned the intent of bringing the virus into the company's network (i.e., was it a competitor). In the FBI case, the following was said by the FBI Director: "At the end of the day, all of our systems probably need to be looked at and maybe improved. But at the end of the day, what we rely upon is honest people." In the Verizon case, the ethical issue is obvious -- what do you do when you cannot trust your employees.

The main point is that to develop security systems, companies should decide on the following organizational issues, in addition to the technologies: how to prevent staff from wrongdoing through clear guidelines, how to control the access to a system to minimize possible damage, and how to survive attacks from internal as well as external adversaries.

12.3 Definitions and Concepts

12.3.1 Overview

Basically, security involves a study of vulnerabilities of a system, identification of threats that could exploit the vulnerabilities, and introduction of circumventions (countermeasures) to handle the threats. Vulnerabilities, threats, and circumventions are usually discussed in terms of the following **(PIA4):**

- ***Privacy (P)*:** assure confidentiality of information (i.e., no one other than the authorized people can see the information) when transmitting it over a network or storing it in a insecure place.
- ***Integrity (I):*** assure retention of information (i.e., no unauthorized modification) during transmission or storage.
- ***Authentication(A):*** identify for certain who is communicating with you (i.e., make sure that you are who you are)

- ***Authorization/*Access control (A):** determine what access rights that person has (i.e., can you only read given information or can you also update, delete, add information).
- ***Accountability (A):*** assure that you can tell who did what when and convince yourself that the system keeps its security promises. A major issue is *Non-repudiation (NR)* -- the ability to provide proof of the origin or delivery of data. NR protects the sender against a false denial by the recipient that the data has been received. It also protects the recipient against false denial by the sender that the data has been sent. In other words, a receiver cannot say that he/she never received the data or the sender cannot say that he/she never sent any data.
- ***Availability (A):*** assure that the system being secured can be accessed and used by the users. Too much security can make a system too cumbersome and non-usable.

These attributes of security can be used to define the level of security (protection) exercised on different corporate resources. The following table shows an example.

Table 12-1: Example of a Security Profile

Resources	Privacy	Integrity	Authentication	Authorization	Account-ability	Non-repudiation
Customer Database	Yes	Yes	Yes	Yes	No	No
Payment System	Yes	Yes	Yes	Yes	Yes	Yes
Web Advertising	No	Yes	No	Yes	No	No
Web Server	Yes	Yes	Yes	Yes	No	No
Corporate Network	Yes	Yes	Yes	Yes	No	No

12.3.2 Evolution of Security – The Government View

Many of the ideas about security approaches are originating from the government work through research conducted through DARPA (Defense Advanced Research Project Agency). According to the DARPA OASIS (www.oasis.org) program, security approaches have evolved through several phases of research:

1GS (First Generation Security): In this phase, the emphasis is on protection of the assets. Detection of security breaches is after the fact. Thus the main emphasis is on recovering from the attacks (i.e., information assurance)

2GS (Second Generation Security): In this phase, the emphasis is on detection of the breaches before they can do any damage (or as quickly as possible).

3GS (Third Generation Security): This phase involves research on building systems that can tolerate (survive) attacks and reconfigure themselves to adjust to the level of attack. These survivable systems are the main thrust of current and future research in security.

As work proceeds through various stages of security, we need to consider the following situations:

Hacking versus Assaults: Hackers are basically "ankle biters" and irritants who can do damage to you while assaults are much more dangerous because the aim of assault is destruction. This could be thought of as pickpockets versus armed robbers. Higher level of protection is naturally needed against assaults.

Intrusion tolerance versus Security: Security generally means "protection" from malicious entities. However, intrusions may occur due to malicious or natural events. For example, a system failure due to a hardware problem or an attack on the system intrudes in your ability to do the work. Basically, intrusion is any undesirable/unauthorized activity that exposes, prevents and/or subverts a legitimate operation. ***Intrusion tolerance*** involves a combination of security and fault tolerance (Intrusion tolerance = security + fault tolerance). DARPA research has indicated that security and fault tolerance approaches contradict each other. For example, fault tolerance is achieved through replication, however, security is achieved through reducing replication (more copies are harder to protect). See Section 12.4.3 for a discussion on how to resolve the tradeoffs.

Information assurance versus Security: Security concentrates on protection while information assurance (IA) deals with how to recover from security breaches. ***Information assurance*** is concerned with combining security with recovery, i.e., you not only protect the assets but also recover from the attacks. IA thus includes security plus backup/recovery, disaster recovery, and contingency planning. Generally speaking, assurance is the "ground for confidence that an entity meets its security objectives," where a security objective is defined as "a statement of intent to counter identified threats and/or satisfy identified organization security policies and assumptions."[3]

12.3.3 Security at Various Layers – A Quick Glance

You need to define and enforce the security policies that are consistent across all elements of applications, middleware services, and networks. These, and other aspects of security, are supported at various levels (network, middleware, application) by using a wide range of technologies (see Figure 12-2). Security is needed at these different levels since security at each level fulfills different requirements. Let us briefly review the security at various levels (details will be given in the next chapter).

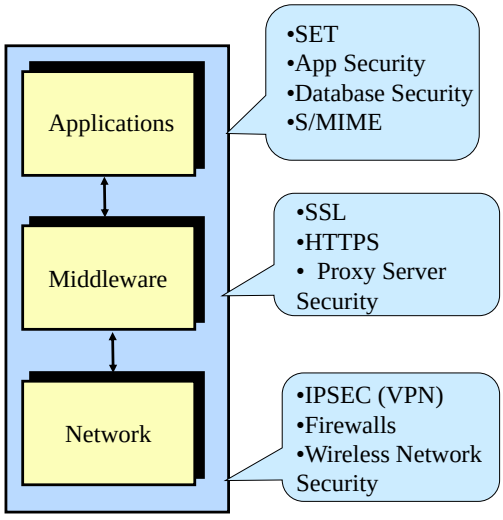


Figure 12-2: Levels of Security

[3] Based on definitions by System Assurance Methodology (SAM) developed as part of the DARPA research.

Network security protects communication and transactions data. "Firewalls" and "gateways" are erected to protect and regulate traffic. In addition, the network traffic can be encrypted at packet level (IPSec) or at the transport level (SSL- Secure Socket Layer). Computing platforms include operating system and database manager security. Middleware can also imbed some security features and S-MIME secures email. A variety of security approaches exist at the application level, in which case authorization controls are used within applications to regulate access to specific data, and cryptographic infrastructures are built to strongly authenticate users and provide confidentiality. Examples of application level security is provided by database managers, Java security, and SET (Secure Electronic Transactions). In particular, applications themselves provide access control and strong user authentication.

Security must be considered at all levels. Securing a higher layer while keeping lower layers unsecured makes the system vulnerable to intrusions from the lower layers In general, lack of security at a certain layer might compromise the overall system even if other layers are secured. Consider, for instance, a system where the application data is secure, but is transmitted over an insecure network. In this case, the overall security of the application could be suspect. Specifically, application security protects application data (e.g. database security mechanisms allow the data to be stored on the hosts in a protected manner) and system resources (e.g. Java Security) while SSL protects data while being transferred on the network.

12.3.4 Security Design – Sneak Preview

A security design approach is needed to include various security issues at different levels. Basically, it is important that the business logic of a Web application runs on a server and not on the client. The Web application server can be used to integrate access to resources (databases, etc.), which provides greater security of the resources. In addition, you should structure your application by using network filters ("firewalls"). A good design protects the Web server (providing presentation services) behind an outer firewall, and the remaining servers (supporting business logic) behind a second, inner firewall. This structure, shown in Figure 12-3, is known as a demilitarized zone, or DMZ. In most cases, a Web server sits alone in the DMZ, handling requests from the Web and passing them along to the secure intranet network and some marketing related applications can also reside on the web server. The applications and internal business systems behind the inner firewall contain all the private business logic and data of the application. In addition, you can gain performance benefits by caching frequently requested data inside the DMZ rather than retrieving it from back-end systems each time it is requested. However, machines in the DMZ are known to be at higher risk. In addition to DMZ, you also need to consider security of clients. For example, mobile devices typically need another level of security before they can enter the DMZ.

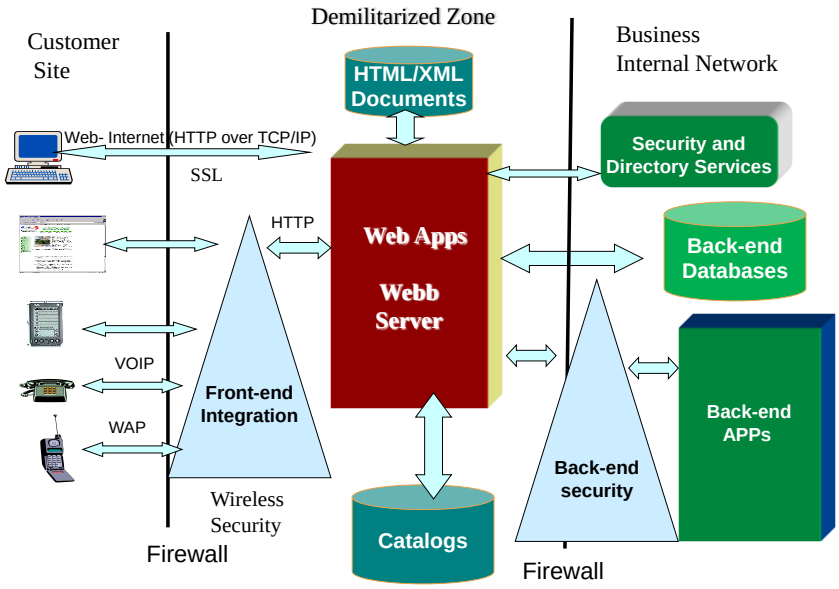


Figure 12-3: Security Design Overview

12.4 Establishing Security Requirements

To protect IT and physical assets, security requirements are established as the first step in building a security architecture. These requirements must include external as well as internal factors.

12.4.1 External Factors that Drive Security Requirements

Several national and international factors drive security initiatives in enterprises. For example, government regulations and consumer/customer attitudes towards security change due to major national or international events (e.g., the September 11, 2001 attack on the World Trade Center and recent attacks in Paris in 2016). These factors, not within the control of organizations, include:

- National and international emphasis such as homeland security
- Eminent or possible intruder/attacker threats on certain days
- Privacy and confidentiality laws and legal requirements imposed by the government (e.g., the heavy regulations in the healthcare industry)
- Consumer/customer attitudes towards security and privacy
- Threat models ("ankle biters versus national enemies") and protection against hacking versus assaults based on general industry observations (banks are more prone to attacks by thieves and military sites are more prone to espionage attacks)

12.4.2 Organizational Requirements

Many security requirements are specific to specific industry segments and organizations within the industry segments. For example, the airline industry has different security requirements than banks and large international banks have different security requirements than small local banks (if any left!). The main idea is: how does security support your organizational goals? The main business driver for business security is the growing reliance of businesses on IT. The organizational requirements should:

- Clearly note the business drivers for security.
- Classify assets to be protected according to relative importance (not everything needs to be protected at the same level).
- Establish realistic survivability and intrusion tolerance requirements.
- Consider tradeoffs between QoS requirements and survivability/intrusion requirements.
- Keep budgetary and policy restrictions in mind because solving all problems can be expensive. The main challenge is: how do you balance costs versus benefits?
- Consider cultural situations while managing security in a global and multi-cultural environment.

12.4.3 Protection Policies

An important aspect of security analysis is establishment of "protection policies" that specify the level of security and safety needed by the selected technologies. Protection policies can be specified by users or system administrators. For our purpose, we specify protection in terms of two dimensions: system security and system availability (see Figure 12-4). The basic idea is that a highly protected system is highly secure and 100% available, i.e., it is intrusion tolerant. Intrusion may be intentional or unintentional and involves a combination of security and fault tolerance. There are several types of intrusions such as the following:

- Nothing modified, i.e., the intruder only looks but does not change anything
- Denial of service, i.e., the intrusion disallows service to legitimate users
- Modification to the system so that it behaves differently than intended
- Damage (permanent or temporary) to the system so that it is not recoverable
- Introduction of viruses

The protection policy chosen against possible intrusion threats can be represented as a tuple: (S, A) where S represents the security level chosen and A the availability. The security S is provided at the following levels:

- Level 0: no security specified
- Level 1: Authorization and authentication of principals
- Level 2: Auditing and encryption (Privacy)
- Level 3: Non-repudiation and delegation

Note that the security levels are inclusive (i.e., if you choose level 2 security it implies levels 0, 1, and 2). Availability A can be represented in terms of replications (more replications increase system availability):

- Level 0: No replication (i.e., only one copy of the resource is used).
- Level 1: Replication is used to increase availability. The resource is replicated for a fail-safe operation.
- Level 2: FRS (Fragmentation, Redundancy, Scattering) is used. FRS schemes split a resource (e.g., a catalog is broken into 4 fragments), replicate it, and scatter it around the network to achieve high availability and intrusion tolerance Details about FRS can be found elsewhere [Deswarte 1988, Deswarte 1991].

Figure 12-4 shows how the security and availability levels can be mapped to protected systems. A protection policy can be chosen for each component of a system. For example, network access can have a different protection policy than a database. You can choose, for example, that the a wireless network needs a protection policy (security level 2, availability level 1) while a database only needs a protection policy (security level 1, availability level 0).

Obviously, there are costs (time, efforts) associated with enforcing a given protection policy. Some protection policies are easier to enforce than others because COTS solutions are available. The policies involving FRS (Fragmentation, Replication, Scattering) require considerable effort because FRS schemes are still being developed.

It should be noted that the security and availability levels are being suggested here as a basis for discussion and analysis. Different security and availability levels can be introduced, if needed. The cells of Figure 12-4 can be used to represent the security requirements in terms of the type of protection level needed for different objects in the system. For example, you can decide that the customer database needs a security level of 2 and an availability level of 1. Thus this figure can be used as a "scatter chart" that represents security requirements.

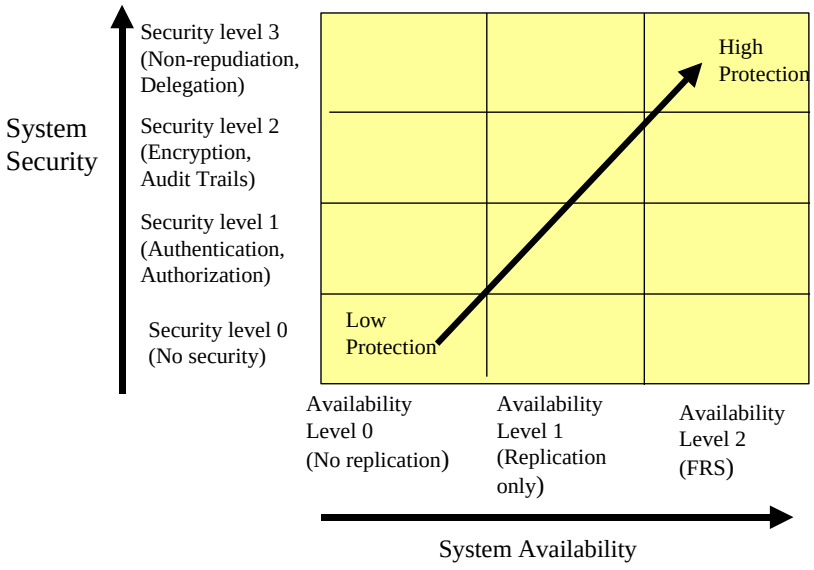


Figure 12-4: Possible Security Policies

12.5 Management Issues in Security

12.5.1 Overview

IT security is not strictly a technical problem. A wide range of management approaches are needed to develop organizational policies, roles, and training programs before choosing the security technologies. The main responsibility of managers is to develop an approach that satisfies the security requirements established above by:

- Developing organizational roles and responsibilities
- Instituting security awareness and training programs
- Understanding the differences between old versus new organizational models and adjusting security approaches accordingly
- Viewing security as a risk management task with financial and legal tradeoffs
- Establishing appropriate security trust models

This section presents a quick summary of these management issues. Many articles on different aspects of security management can be found in the handbook edited by Tipton, H. and Kraus, M., "Information Security Management Handbook", 4th and 5th Edition, Auerbach, 2000.

Results of the security management process are usually represented in several documents with tables and charts. The following table shows an example of the main resources (databases, computers, networks) to be protected, the level of protection needed (based on requirements), the person who will be responsible for the protection of the resource, and the training/awareness needed. Other columns can be added to this table.

Table 12-2: Sample Security Management Decision Table

	Level of protection Needed	Person Responsibility	Training/awareness Needed
Resource1			
Resource2			
Resource 3			

12.5.2 Organizing for Security: Roles and Responsibilities

A key element of a security management program is to put someone in charge of security. In most cases, information system security has not been a separate job but instead a job function that is attached to other jobs. For example, it is common to find database administrators as in charge of database security, network administrators responsible for network security, and IT managers responsible for IT security. It is better to appoint:

- An information security coordinator for each business unit (BU) to coordinate the security activities of the BU.

- A Chief Security Officer for overall coordination at the enterprise level.

It is also important to separate duties, i.e., a single individual should not have complete control of everything. In addition, rotation of responsibilities to keep a fresh perspective is important. It is not a good idea to keep one person in one position too long. Rotation of responsibilities has some plusses and minuses. On the plus side, it keeps interest up, makes more people aware of the security issues as they are rotated through various responsibilities, and assures that security people do not bypass procedures for friends. For example, I have noticed that security guards in buildings do not check IDs diligently once they know that the person works there. It is possible for a person to have a revoked ID but still go in and out of buildings because he/she is a familiar face. Changing security guards periodically takes care of this problem. The main minus of job rotation is that it may reduce job efficiency as new people need to be trained and are going through the learning curve.

12.5.3 Security Awareness

Security awareness is a vital aspect of security management. Even the best crafted policies and procedures fail if no one is aware of them. In most cases, these policies and procedures are documented in thick security documents, and very few people actually read the security documents. Thus a good security program must include the following steps (see [Peltier 2000]):

- Develop the security policies and procedures. Security policies include what needs to be secured at what level, and what is the penalty for breach. You also need to establish procedures and guidelines to enforce the policies and select technologies to enable the above.
- Make the people fully aware of the policies and procedures. This awareness should encompass employees, managers, as well as the customers.
- Organize security awareness days and drills. These events, if done correctly, can be very useful in explaining the importance of security and the roles played by policies, procedures, and staff in maintaining a vigilant security system.
- Institute random checks (after hours) to see if security policies are being followed. In an organization that I worked in, they checked to see if material marked proprietary and confidential was left openly in rooms and also if the desktops were properly "locked" (the screen saver required an ID and password to access the system). The next day, we had a green or red sticker on our doors to indicate if our areas were properly secure.
- Publish security alert newsletters to inform people about some real examples of security breaches and the actions taken. Once again, in my own experience, the security newsletter was published every quarter or so, was a one-pager, and had some interesting (some humorous) examples of security breaches or attempts. The newsletter indicated very clearly the actions taken by the company -- in many cases termination of employment.

12.5.4 Security Training

Security training is an important element of a security program. To identify training needs, it is a good idea to document key security threats and then assess current levels of awareness about the threats. The "gap" between what is needed versus what is known should drive the training program.

It is essential to get a corporate buy-in before embarking on an ambitious security training program. Let us face it -- while security is important, it is not very interesting to be trained on security and then to read the wonderful security documents. I have slept through several of them. Some type of buy-in and corporate incentive is typically needed to get the general organization population on security. Here are some ways to conduct security training somewhat successfully:

- A required course (lecture or computer aided) that the employees have to take as a condition of continued employment.
- Provide incentives (e.g., part of performance appraisal) for security training. This method works quite well and I have used it several times as a manager. Employees who do not take a required training course in, for example, business ethics or security have something said about it in their yearly performance review.
- Self-awareness tests/questionnaires that are sent to employees who have plenty of time to do this.

12.5.5 The Role of Organizational Computing Models

The organizational computing models greatly impact the security policies and procedures. Traditional computing models are:

- Host-dependent (centralized): mainframes serve as the hosts of major applications and databases.
- Hierarchical: the control and authority flows from top (central) sites to lower sites.
- Closed: only known users can use the systems and the managers can find out who is using the systems and when new users are added.
- Point-to-point: the connections between the various computers (mainframes, workstations, desktops) are well defined because leased point-to-point lines interconnect the computers.
- Homogeneous: the computers and the software are largely supplied by one supplier (for example, IBM and/or Microsoft).

Modern enterprise computing models, on the other hand are:

- Decentralized and distributed: the data as well as programs are widely distributed among a variety of computers.
- Flat: there is very little top down control. Traffic flows between computing devices between central controls.
- Open: the current Internet-based systems are quite open and it is very difficult to know who is communicating with whom.
- Broadcast: On most current networks, instead of the point-to-point connections between multiple users, messages are broadcasted over shared media.

- Heterogeneous: the current systems are very heterogeneous with hardware and software from a multitude of suppliers.

Naturally, it is easier to manage the security in the traditional environments than the modern Internet-based environments. In reality, most current organizations have a mixture of traditional mainframe-based and newer systems. The main challenge is to decide how to deal with these mixtures. The following guidelines may be used (see [Murray 2000]):

- Single user name space and single user logon should be used to hide the multitudes of systems that the users need to access (see [Vacca 2000]).
- Use strong authentication to assure that the unknown users in the open environment are who they should be.
- Keep the business service or application (user visible entities) as high level points of control to keep focus on what is important.
- Use firewall to localize issues and to segment the activities for control as much as possible.
- Protect keys and security infrastructure as diligently as the resources themselves.

12.5.6 Risk Management and Attack Trees

Security approach can be viewed as risk management. i.e., manage the risks associated with threats to the system. The basic idea of risk assessment is:

- What could happen (threat event)?
- If it happened, how bad could it be (threat impact)?
- How often could it happen (threat frequency, annualized)?
- How certain are the answers to the first three questions?

Based on assessment, a risk mitigation approach needs to be developed. This entails:

- Are the risks acceptable?
- What can be done?
- How much will it cost (annualized expense)?
- Is it cost effective?
- Can risk be transferred (e.g., buy insurance policy)?

The effort needed to mitigate risks should be proportional to risk assessment. You do not want to spend millions of dollars to protect an asset that is worth two thousand dollars. It is a good idea to estimate a total expected loss that is given by:

Total Expected Loss (TEL) = L1xF1 + L2xF2 + …. LnxFn

where:

L1, L2,,, Ln are losses expected with event 1 to n

F1, F2,,,, Fn are frequencies per year

TEL is typically annualized, I.e., if you loose $1M for an event that can happen once every 10 years, then annual TEL = $100K. The loss can be tangible (quantitative) or intangible (qualitative). The loss L for an event can be measured in terms of asset value (A) and exposure (E) per event, i.e. L = AxE. Consider, for example, a restaurant of value $10M that is being insured for fire. There is a 50% chance that a fire in the kitchen will burn the restaurant. Thus L = 10x0.5 = $5M.

A great deal of literature exists on qualitative versus quantitative aspects of risk analysis. Elements that need to be estimated include:

- Asset value
- Threat frequency
- Threat frequency exposure
- Safeguard cost
- Safeguard effectiveness

Initial attempts at quantifying everything have not succeeded. Qualitative measures of Low, Medium, and High are commonly used. The main problem is that these measures can be highly subjective (your low may mean my high). A possible solution is to assign ranges, i.e., L < 10K and H > 100K.

Attack trees, introduced in the SAM methodology and discussed later in Section 12.8, are a convenient way to explore potential attacks and thoroughly examine the "attack space". An attack tree is simply a tree that is similar to a logical decision tree used to perform a systematic analysis of the attack space in terms of what is under attack, where the attack could happen, when the attack could take place and how the attack could happen. An important objective of the tree is to provide a heuristic for systematically considering attacks. To illustrate the key points, let us develop an attack tree against a physical safe (see Figure 12-5). The goal of attackers is to open the safe. To open the safe, attackers have several options: they can pick the lock, learn the combination, cut open the safe, or install the safe improperly so that they can easily open it later. Now you can assign values -- I (impossible) and P (possible) in this figure -- to the leaf nodes to indicate what needs to be considered next. You can now pursue the nodes that are possible for further evaluation. Let us now evaluate "learn the combination" node and break it into two activities: find the combination written down or get the combination from the safe owner through eavesdropping or other means. Each node becomes a sub-goal, and children of that node are ways to achieve that sub-goal.

In the attack trees, there are AND nodes and OR nodes (everything that is not an AND node is an OR node). OR nodes are alternatives while AND nodes represent steps toward achieving the goal. For example, to eavesdrop on someone for the safe combination, attackers have to eavesdrop on the conversation AND get safe owners to say the combination. Instead of "possible" and "impossible" values to the nodes, you can assign other values (easy, difficult, very difficult). Values such as expensive versus inexpensive, intrusive versus nonintrusive, legal versus illegal, special equipment required versus no special equipment can be used. Assigning "expensive" and "not expensive" to nodes can help in analyzing if the asset is worth protecting, For example, if the asset is worth $10,000 and it takes $30,000 to steal it and $100,000 to protect it, then a decision has to be made about protection.

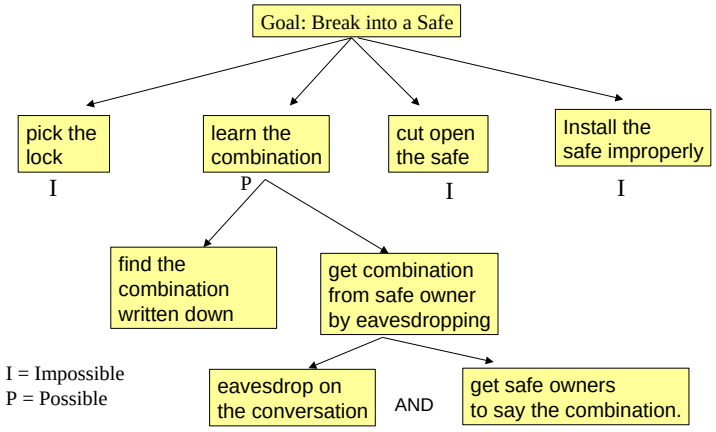


Figure 12-5: Sample Attack Tree for Opening a Physical Safe (Shneier 1999)

An important objective of the tree is to provide a heuristic for systematically considering attacks. When fully developed a tree should aim for completeness. However, since there are many possible attacks, it may not be feasible to put them into a tree. The leaves of attack trees are indicators of attacks. Each attack should be characterized in terms of technical and non-technical capabilities that are required, access needed to the system, risk assumed, and objective of the attack (see [Schneier 1999] for additional discussion): In large systems, it is not possible to draw large attack trees as graphs. In such cases, it is possible to describe the tree in terms of document sections and subsections. The following is an example of an attack tree for denial of service represented as a word document. In this case, the tree is defined in terms of what, where, when and how. We have not assigned "possible" or "impossible" values to the nodes.

S1. What. Service Assurance of System Threatened – Denial-of-Service Attack on System

S1.1. Where. Individual Component or Subsystem

S1.1.1.Where. End system

S1.1.1.1. When. During network operation

S1.1.1.1.1. How. Passive attack

S1.1.1.1.1.1. How. Eavesdropping

S1.1.1.1.2. How. Active attack

S1.1.1.1.2.1. How. Illegal logon or system entry as user or root to cause denial of service

S1.1.1.1.2.2. How. Dial-port flooding

S1.1.1.1.2.3. How. Shutdown

S1.1.1.2. When. During development …

S1.1.2. Where. Router

S1.1.2.1. When. During network operation

S1.1.2.1.1. How. Passive attack

S1.1.2.1.1.1. How. Eavesdropping (preparing for DoS attacks)

S1.1.2.1.2. How. Active attack

S1.1.2.1.2.1. How. Attacks on routing protocols

S1.1.2.2. When. During development …

12.5.7 Security Trust Models

In small companies, it is easy to know who you are communicating with. However, in large environments, the players communicate with several parties in C2B, B2B, B2E, and other configurations over the Internet. In many cases, you are communicating with people outside of your corporate environment, including some you have never met, such as vendors, customers, clients, associates, and so on. Establishing a line of trust in such a setting is difficult. Companies follow a range of *trust model*, such as the following, which dictate how users will exchange security credentials with each other.

- **Direct Trust.** Direct trust is the simplest trust model. In this model, a user trusts the partner because he or she knows the partner directly. This model is simple but does not extend to a large number of users in the Internet environments.
- **Hierarchical Trust.** In a hierarchical system, there are one or more third parties that are trusted by the users. This type of trust "tree" is used in many distributed systems.
- **Web of Trust.** A web of trust encompasses both of the other models and is a cumulative trust model. In other words, you trust my opinion that others are good and honest people only if you consider me to be a trusted person. Systems such as PGP (Pretty Good Privacy) use this model.

Time to Take a Break

✓ • Security Requirements and Management
• Basic Cryptography
• Authentication, Authorization and PKI
• Security Methodology

Suggested Review Questions Before Proceeding

- What are the key security concepts?
- What are different choices in securing systems at network, middleware, and application levels?
- What are the tradeoffs between security and availability?
- What type of requirements drive a security solution?
- What are the main management issues in developing a sound security system?
- What is FRS and how is it related to security?

- How is risk management related to security?
- How can attack trees be used in risk management?

12.6 Basic Cryptography

12.6.1 Cryptography Overview

Cryptography has been used for a number of years to mask the messages so that the interveners could not see or modify the messages. In fact, cryptography is a Greek word that means "secret message" and was used by Julius Caesar to mask messages he sent to his generals. Cryptography over the years has become a science of using mathematics to encrypt and decrypt data. It enables you to store sensitive information in files or transmit it across a network so that it cannot be read by anyone except the intended recipient.

The main ingredients of cryptography, as illustrated in Figure 12-6, are:

- ***Plain text,*** also known as clear text, is the original message or data that the sender wants to send.
- ***Encryption algorithm,*** or ***cipher***, performs various transformations and substitutions on the plaintext, i.e., scrambles the plaintext. These algorithms *are* mathematical functions used in the encryption and decryption process.
- ***Key***, used by the encryption algorithm, to scramble (encrypt) the plaintext. The exact transformations and substitutions depend on the key. A *key can be a* word, number, or a phrase. The same plaintext encrypts to different ciphertext with different keys.
- ***Ciphertext*** is the scrambled message produced as an output. This message depends on the encryption algorithm and the key.
- ***Decryption algorithm*** unscrambles the ciphertext and is effectively a reverse of the encryption algorithm. The decryption algorithm may use the same or a different key for decryption. As we will see later, this is the difference between symmetric and asymmetric encryption.

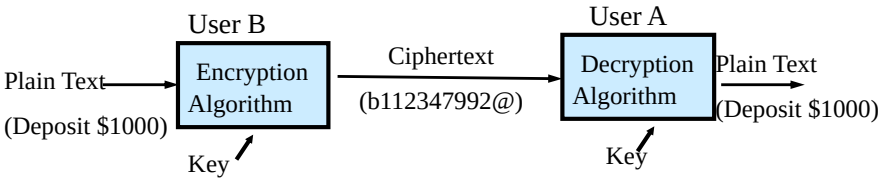


Figure 12-6: Cryptography Components

To illustrate the key ideas, let us look at Caesar Cipher -- an old but well known example of cryptography. Caesar Cipher was used by Julius Caesar to send encrypted messages to his generals. The main idea was to replace each input letter with the third letter to the right as shown below:

a b c d e f g h i j k l m n o p q r s t (input - plaintext)

d e f g h i j k l m n o p q r s t u v w (output- cipher)

In this case the key is 3 and the encryption algorithm is "shift right". To decrypt, the receiver reverses it (i.e., shifts the ciphertext 3 letters to the left). For example, the plaintext "come back" would produce a ciphertext of "frph edfn". No wonder the enemies of Caesar were so confused!

While cryptography is used to secure data cryptanalysis is used to analyze and break secure communication. Cryptanalysts, also called attackers, use a combination of analytical reasoning, mathematical tools, pattern finding, patience, determination, and luck. Cryptology includes both cryptography and cryptanalysis.

Cryptography can be strong or weak depending on the time and resources needed to recover the plaintext from a ciphertext. Strong cryptography produces a ciphertext that is virtually impossible to decipher. The main idea is to make this task so difficult that given today's computing power and available time, it is not possible to decipher the result of very strong cryptography for thousands of years. This does not mean that an extremely determined cryptanalyst does not get lucky or just uses other means to access secret keys (greed and money still work!), In addition, newer computing systems with massively parallel processors could have some impact on difficulties in breaking strong cryptography. Basically, the security of encrypted data is entirely dependent on the strength of the cryptographic algorithm, length of the key, and the secrecy of the key.

Due to ecommerce, encryption/decryption has become a major area of active work. In addition, the events of September 2001, have heightened the need for security. In most cases, data is transformed into an encrypted message. The encrypted message is then transmitted and decrypted on the other side by using the same key. This type of cryptography, known as Conventional or Symmetric Encryption, is discussed below. Encryption/decryption can be performed by hardware and/or software. Modern computing systems have the ability to implement very sophisticated encryption/decryption techniques. The same encryption can be used on all data in a system or encryption keys can be more "personalized". For example, instead of using the same encryption/decryption key on all data from all stations in a network, each user can use its own encryption/decryption key. A user can have his or her own encryption card which is inserted into a workstation before the user logs on. This card encrypts the data before sending it across the network. The encrypted data can be read only by those users or programs with access to an appropriate key. The sidebar "STEGANOGRAPHY: An approach to Hide Secret Information in Image Files" shows an interesting example of encryption. Encryption techniques generally fall into two broad categories: symmetric key and asymmetric key.

STEGANOGRAPHY: An approach to Hide Secret Information in Image Files

Steganography is a popular technique used in cryptography – it conceals information to be transmitted in image files, audio clips, video clips, or other formats. This is how it works for a message M, let us say 135, that needs to be transmitted over a network:

- Message M is encrypted into an image file, let us say picture of a room
- The "encryption" key used is to indicate the first digit by the number of tables, second digit by the number of couches, and the third digit by the number of chairs in the picture.

- The encrypted message M* shows a picture of a room with one table, 3 couches, and 5 chairs.
- The encryption key is sent to the receiver separately (e.g., a phone call)
- M* is sent to the receiver who uses the encryption key to decrypt the message.

The stenography schemes are automated to handle very sophisticated scenarios. Thus digital steganography takes care of encryption where the information is embedded in an image file by a sending program and the recipient program decrypts it to extract the information in the images. Obviously there are many ways to be encrypt and decrypt image files by using many complex algorithms.

Source: Fridrich, J., "Steganography in Digital Media: Principles, Algorithms, and Applications", Cambridge University Press, 2009

12.6.2 Symmetric Key Encryption – The Conventional Approach

12.6.2.1 Overview

In a symmetric key encryption scheme, the *same key* is used by the sender to encrypt the message and the receiver to decrypt it. For example, as shown in Figure 12-7, the same key (e) is used to encrypt as well as decrypt. To use this cryptographic system, you encrypt the plaintext by using receiver's secret key e. Thus when the receiver receives the ciphertext, it can be decrypted by using e. The main issue in this widely used encryption scheme is that the senders and receivers have to agree on a secret key e. The main problem, as we will see, is that this scheme does not scale well to large number of users because all senders and receivers have to agree on secret keys before transmission. In addition, how secure can this key be if it has to be exchanged between many senders and receivers?

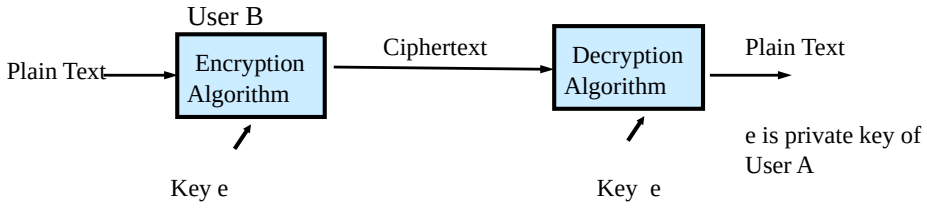


Figure 12-7: Symmetric (Secret) Key Encryption -- the Conventional Approach

The idea of symmetric key encryption is very old -- that is why it is known as the conventional encryption. For example, the Caesar's cipher mentioned previously is an early example of symmetric key encryption. Many old classical approaches were variants of Caesar's cipher where:

- Key can be n instead of 3
- The shift can be left or right

- The alphabets can be replaced with numbers and special characters
- The shifts and the directions can change periodically (e.g., the first 10 characters are shifted by 3 letters to the right and the next 10 letters are shifted 6 letters to the left, etc.).

The main limitation of the old character based cryptography is that they are not very strong -- you can shift characters left and right just so long. You also cannot perform mathematical functions on characters (you cannot add or multiply alphabetic characters). Modern cryptographic techniques have changed all this by using translation of plaintext into bits instead of characters. The characters are translated into bits by computer systems by using ASCII or EBCDIC codes anyway, so no extra effort is needed here. Once everything is in bits, now random numbers can be generated for keys and then "applied" to the plaintext bits by using a variety of mathematical functions.

A variety of symmetric key encryption algorithms have been introduced over the years. DES (Data Encryption Standard) is the most common algorithm (64 bit key). In several cases, DES is applied multiple times for stronger crytpography. TripleDES is an example. Other variants of DES such as DESX, GDES, and RDEX have also been developed. Additional examples of private key encryptions are RCx, IDEA (International Data Encryption Algorithm), and Blowfish. See [Stallings 2001] for details of these and other algorithms).

12.6.2.2 Modern Cryptography Foundations

The basic idea of modern cryptography is that all data (plain text, cipher text) is in bits instead of alphanumeric characters. A random number K is generated to serve as a key. For encryption, complex functions are applied to the key and the message bits (i.e., added, subtracted, shifted, "orred"[4], "exclusive orred"[5]) to produce the ciphertext (see Figure 12-8).

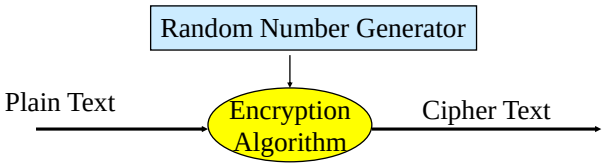


Figure 12-8: Modern Cryptography

Modern cryptography uses two types of cryptographic algorithms: stream ciphers that operate on continuous bits and block ciphers that operate on blocks of data.

Stream ciphers accept a random number as bit streams and perform operations between these bits and the bits of the plaintext. The operations can be shift rights, shift lefts, additions, or other bit operations. These ciphers are usually implemented in hardware devices.

Block ciphers are usually implemented in software and are much more popular than stream cryptos. This scheme operates on blocks of text – usually in multiples of 8 bits (character length). DES (Data Encryption Standard) developed by NSA (National Security Administration) is the best known example of this type of encryption. The key, also known as cipher variable, is 64 bits long. The encryption algorithm works as follows (see Figure 12-9):

[4] Or operation is a binary arithmetic operation which indicates that the resulting bit is on if any of the input bits is on.

[5] Exclusive or is a binary arithmetic operation which indicates that the resulting bit is on if the input bits are dissimilar

Plain text is divided into 64 bit blocks (T1, T2,,,,). These blocks are processed in several iterations and produce corresponding ciphertext blocks (C1, C2,,,).

Each block is divided into 2 parts: L_0 and R_0 (left and right bits)

The two halves are processed as follows for I=0,1,…, 15 by using a complex function f:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} + f(R_{i-1}, K_i)$$

Each block (T1, T2, …) is processed several times and new R is computed from L by using f to generate the ciphertext blocks C1, C2, ….).

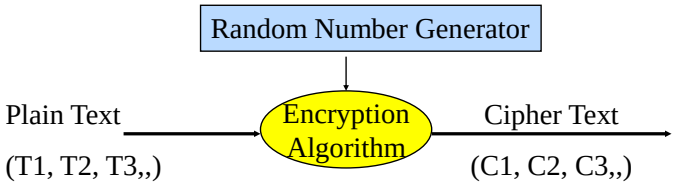


Figure 12-9: Block Crypto Algorithms

12.6.2.3 Issues with Symmetric (Secret) Key Cryptography

While private key encryption is usually very fast and efficient, the problem is with key management. In other words, since the sender and receiver have to agree on the same key, sending the key from one side to the other might compromise it. Basically, If someone finds out the key then he/she can decrypt the ciphertetxt. So how can the sender and receiver know about the key safely? Naturally, it is not a good idea to send the key plus the ciphertext in the same message -- they should be separate messages. For security, the key can be generated by software or hardware. A typical technology used in many corporations is secure card. A secure card is given to each employee. The card has a chip which generates a random number R every minute. The number generated is unique to the employee. To access a corporate system, the employee types in his/her employee ID plus R -- this becomes the key K. On the receiving system, a similar routine generates R for the employee. Messages sent are encrypted by using K that the user types and are decrypted by using the key K generated by the system. If these keys do not match, the message cannot be decrypted.

What do hackers do? Typical hacker practice consists of the following steps:

- Trap the messages between senders and receivers.
- Try to understand the messages. In many cases, the messages are not encrypted and are in cleartext. This is hacker delight.
- If messages are encrypted, try to guess the key. Sophisticated unscramblers can be devised that run forever and try to guess the keys by using different combinations.

The main idea of strong encryption is to make sure that the key must be very difficult to guess. For example, the secure ID does generate keys that are quite difficult to guess. Key lengths and sophisticated key processing improves security but adds significant overhead.

12.6.2.4 Public Key (Asymmetric) Cryptography for Better Security

In a public key (asymmetric) system, the encryption Key e and the decryption key d are different - hence the name "asymmetric" (see

Figure 12-10). Each user has a pair of keys, a private key d that he keeps secret and a public key e that he publishes. When Pat needs to send a message to Joe, she encrypts the message with Joe's public key e_j. This encrypted message can only be decrypted with Joe's private key d_j. Therefore if this encrypted message is delivered to a user Sam who does not know d_j, then Sam cannot decrypt it. Even when the message is broadcasted over the Internet, only Joe can decrypt the message because he is the only user who knows d_j. The main point of public key systems is that the decryption key is completely private and is not transmitted over the network. While public key systems solve the problem of key management, they are usually significantly slower than private key systems. The RSA (Rivest, Shamir and Adleman) algorithm, developed in 1976 is by far the most widely used public key encryption algorithm. Additional details about Public Key (Asymmetric) Cryptography can be found in Appendix A (Section 12.14).

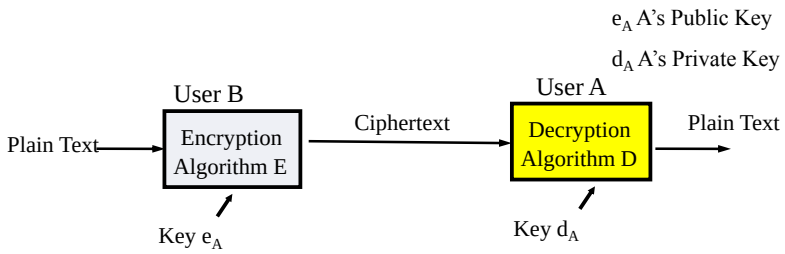


Figure 12-10: Public (Asymmetric) Key Encryption

12.6.3 Core Security Technologies – A Quick Overview

A large number of security technologies have been introduced in the past several years. To keep our focus on management aspects in this chapter, we have moved these details to Appendix A (Section 12.14) and Appendix B (Section 12.15) at the end of this chapter. A quick overview of these technologies is presented her as an executive summary.

User logon and password is one of the oldest and still most commonly used technologies. In this case, a system keeps track of who can access that system. This technology enables the use of existing systems with minimal disruption to existing infrastructure and applications.

Encryption is another old technology that has been used for a number of years to mask the messages so that the interveners cannot see/modify the messages. Due to ecommerce, encryption/decryption has become a major area of active work. In the simplest case, data is transformed by a key into an encrypted message. The encrypted message is then transmitted and decrypted on the other side by using the same key. Encryption/decryption can be performed by hardware and/or software. Modern computing systems have the ability to implement very sophisticated encryption/decryption techniques. The same encryption can be used on all data in a system or encryption keys can be more "personalized". For example, instead of using the same encryption/decryption key on all data from all stations in a network, each station or user can use its own encryption/decryption key. A user can have his or her own encryption card which is inserted into a workstation before the user logs on. This card encrypts the data before sending it across the network. The encrypted data can be read only by those users or programs with access to the same encryption key. Encryption is generally discussed in two different formats:

Private Key: In a private key (symmetric) encryption scheme, the same key is used by the sender to encrypt the message and the receiver to decrypt it. While private key encryption is usually very fast and efficient, the problem is with key management. In other words, since the sender and receiver have to agree on the same key, sending the key from one side to the other might compromise it.

Public key: In a public key (asymmetric) system, the encryption Key E and the decryption key D are different - hence the name "asymmetric". Each user has a pair of keys, a private key D that he keeps secret and a public key E that he publishes. When a sender Bob needs to send a message to a user Joe, he encrypts the message with Joe's public key E(J) . This encrypted message can only be decrypted with Joe's private key D(J). Therefore if this encrypted message is delivered to a user Pat who does not know D(J), then Pat cannot decrypt it. Thus when Joe receives the message, he can decrypt the message by using D(J) and read the message. Notice that in this key the decryption key is private and not transmitted over the network. While public key systems solve the problem of key management, they are usually significantly slower than private key systems. The RSA (Rivest, Shamir and Adleman) algorithm, developed in 1976 is by far the most wildly used public key encryption algorithm.

Digital signature is used to authenticate the source of a message. It is essentially the same as a public key system except that the order in which the keys are applied is reversed. A sender “signs” the message by applying his private key to it. The sender sends the message and the signature to the receiver. The receiver checks the signature by applying the sender’s public key to it. If the receiver gets the original message back, he is sure that the message was signed by the sender’s private key, and therefore, was sent by the receiver himself. In essence, a digital signature is a block of data created by applying a cryptographic signing algorithm to some data using the signer's private key. Digital signatures may be used to authenticate the source of the message and to assure message recipients that no one has tampered with a message since the time it was sent by the signer.

Message Digesting is used to make sure that a certain message was not changed along the way between the sender and the receiver. A message digest algorithm produces a fingerprint of the message, by applying a hashing function to it. The receiver can check for the integrity of the message by reapplying the hash function and comparing with the original fingerprint. The hash functions used in these schemes are such that the fingerprint changes dramatically if a single bit of the message changes.

A Digital Certificate binds an entity's identification to its public key and is issued by the Certification Authority. Digital certificates, based on the X.509v3 standard, enable Internet applications and other users to verify the identity of an entity. Unfortunately, certificates produced by one product may not interoperate with other products because X.509 does not define the formats of the certificate entries and other necessary provisions. PKIX, the X.509 standard by IETF, defines the contents of public key certificates and is intended to resolve these interoperation issues

Time to Take a Break

- ✓ Security Requirements and Management
- ✓ Basic Cryptography
- Authentication, Authorization and PKI
- Security Methodology

Suggested Review Questions Before Proceeding

Note: Some of these questions are related to Appendix A

- What is cryptography and what are its main ingredients?
- What are the key differences between the symmetric and asymmetric key cryptography?
- List the main security technologies and discuss how are they related to cryptography?
- What are digital signatures and how do they differ from message digests?
- How do digital envelopes combine symmetric and asymmetric keys?
- Explain the man in the middle problem through an example

12.7 Authentication and Authorization

12.7.1 Authentication

In a digital enterprise, you need to authenticate the consumers who buy your products or services, employees who access internal systems from remote locations via the public Internet, or business partners who are tightly integrated into your supply chain and ERP systems.

For authentication, a large number of systems employ **user ID and password** as a basis for authentication. Due to known problems with IDs and passwords (i.e., hackers guessing the passwords), some applications choose to make use of one-time passwords. However, the use of such one-time passwords often requires the deployment of token cards -- an expensive and labor intensive effort. This is why software-based solutions are more popular. Most systems enforce authentication by developing a ***session key*** that establishes the identity of partners at session start and is used throughout a session. But then this session key needs to be encrypted. Should a private or public key system be used? Given the advantages and disadvantages of these approaches (private key is efficient but not very secure and public keys are not efficient but secure), in practice, a public key system is

used to exchange the session key between the two sides. Then this key is used in a private key system only for that session. Many current systems, such as SSL (Secure Socket Layer) use this technique.

12.7.1.1 Passwords versus Passphrases.

A *password* is a unique string of characters that a user types in as an identification code. A *passphrase* is a longer version of a password, and in theory, a more secure one. Passphrases are typically composed of multiple words, thus they are more secure against standard *dictionary attacks*, where the attacker tries all the words in the dictionary in an attempt to determine the password. For security purposes, passphrases should be relatively long and contain a mixture of upper and lowercase letters, numeric and punctuation characters. Some packages, such as PGP, use a passphrase to encrypt the user private key on your machine instead of passwords.

12.7.1.2 Using Cryptography for Authentication

Cryptography is used mainly for confidentiality and privacy by encrypting messages. However, public key cryptography provides a method for authentication also through *digital signatures*. A digital signature, discussed in Section 0, enables the recipient of information to verify the authenticity of the information's origin. Thus, public key digital signatures provide *authentication* and digital signature technology can be used to authenticate the source of a message instead of, or in addition to, the traditional ID and password.

12.7.1.3 Using Kerberos for Authentication

Kerberos (http://ww.mit.edu/kerberos/) is a cryptographic authentication scheme developed at MIT. It uses a third-party authentication server to grant cryptographic "tokens" that authenticate users to a given service. Kerberos is used quite heavily for user authentication because it supports, in addition to the venerable user-ID and password authentication: additional authentication schemes such as certificate-based public key systems: asymmetric-key cryptography; smart cards; and token cards. We will re-visit Kerberos later.

12.7.2 Authorization and Access Control

Authorization is concerned with assuring that only authorized users can access a particular system privilege. Authorization relies heavily on access control -- the process of checking whether an authenticated user's privileges permit the execution of a particular operation on a particular protected resource. For example, can Alice withdraw money from account zc-11-35? The access control is typically enforced through access control lists (ACLs) that may look something like the following:

User name	Resource Name
Joe	Payroll
Sam	Accounting
Tim	Inventory control

Scalability of ACLs is a major issue because modern applications may scale to dozens or hundreds of Web servers and potentially tens of millions of end users. The administration of ACLs can be very complex if they must be configured on each Web server system. Authorization to back-end data or subsystems must be handled as well, including systems that have existing authorization mechanisms.

In addition, authorization to other key e-business resources such as objects and message queues must be incorporated.

Due to the complexity of managing ACLs, many applications provide access control on their own because it is not always possible to provide intra-application access control using Kerberos or public-key schemes. Some products have been released that make use of the Distributed Computing Environment (DCE) access control policies. These products, such as HP's Praesidium, make use of the fine-grained access control capabilities of DCE and link them to the deployment of Kerberos within a system. Other products such as the Tivoli Secureway Policy Director provides a centralized authorization service that is the point for administering access controls for Web servers, Web applications servers, firewalls, EJBs (Enterprise Java Beans), and other systems.

12.7.3 Accountability and Assurance

A system needs to log all attempts to access corporate resources to ensure that the system is secure. This logging can also facilitate management decisions by allowing analysis of use patterns. A comprehensive, distributed logging and audit facility for Internet-based applications is needed.

In essence, an e-business must provide assurance that the infrastructure and application resources, including systems, networks, and data, are protected with regard to confidentiality and integrity. This includes protecting the enterprise network and systems from various forms of attacks, and also requires that the communications between the consumer or business partner and the application is secure and confidential. A solution architect can choose from the set of mechanisms discussed so far to satisfy the specific security requirements for the solution. Two additional considerations are:

Intrusion detection - These services emphasize early detection of intrusions. Should a DMZ, extranet, or any internal system be compromised, you need to detect that fact early, and take necessary actions to prevent the launching of a further attack into the private network.

Virus detection - Computer viruses can enter your systems in a variety of ways: via e-mail attachments, from software installs, from files brought by employees from home, etc. They can quickly proliferate from system to system, user to user and cause damage to data, applications and networks. Viruses must be quickly identified and isolated, and damage must be promptly repaired.

12.7.4 Certifying Authorities and PKI

A major problem with the public key cryptosystem is that it works well as long as you know the public key of the recipient. Basically, you must be vigilant to ensure that you are encrypting to the correct recipient's key. How do you find out the correct public key of a recipient? If you freely exchange keys via public servers, *man-in-the-middle* attacks as discussed above are a potential threat. It is vital that you are assured that the public key to which you are encrypting data is in fact the public key of the intended recipient and not a forgery. In small settings, you could simply encrypt only to those keys which have been physically handed to you, let us say, on an external drive. But how do you exchange information with people you have never met; how can you tell that you have the correct key? Although a variety of approaches can be developed, the most practical approach is that of using a trusted third party, called ***"certifying authorities"***. Certifying authorities, along with several other support mechanisms to ensure a strong trusted environment, are known as the ***Public Key Infrastructure (PKI).*** See Appendix B (Section 12.15) for more details.

Time to Take a Break

- ✓ Security Requirements and Management
- ✓ Basic Cryptography
- ✓ Authentication, Authorization and PKI
- Security Methodology

Suggested Review Questions Before Proceeding

- What are the differences between authentication and authorization?
- What exactly is PKI and what does it consist of?
- What are digital certificates and how are they related to certifying authorities?
- What are the key players in PKI and how are they interrelated to each other?

12.8 Putting The Pieces Together – A Methodology

12.8.1 Overview

Several methodologies, formal as well as informal, have been reported in the literature for security and information assurance. An example is the Red Team methodology used in several defense oriented projects. This methodology shows how a Red Team, a team of trained security professionals, can audit the security of a system. It also provides a useful and broad characterization of possible attack types. Another example is the work of Volkmar Lotz, "Threat Scenarios as a Means to Formally Develop Secure Systems" (LICS 1146, 1996; also a Munich Ph.D. thesis). However, Lotz's method is a formal method that is based on streams of messages communicating over channels. Threat scenarios are given abstract characterizations as streams, and they interact with streams abstractly representing system behavior. The following discussion is largely based on the System Assurance Methodology (SAM). The objectives of SAM are :

- Identify typical significant threats and characterize in a systematic way likely attacks to IA systems relative to their missions and designs.
- Characterize adversaries by their motivation, objectives, resources, tolerance for risk, and required access to targeted systems, i.e., develop a theory of adversarial behavior in terms of the attacks they are likely to mount.

- Characterize countermeasures systematically by the burdens they place on systems in development and in operation and by the effects they have on attack characteristics and the resources required by adversaries to exploit vulnerabilities and execute attacks.
- Characterize systematically threats, likely attacks, and countermeasures over time during the evolution of IA systems.
- Characterize gaps and needed remedies in the IA Program as the result of finding specific threats, likely attacks, and countermeasures.
- Determine measurable positive changes in IA systems as they evolve.
- Assist in the strengthening of systems through design changes to counter given threats and likely attacks.
- Provide help in determining the direction and progress of the IA Program.
- Provide a means for strategic planning for IA systems.

12.8.2 Description of the Methodology

Figure 12-11 shows a simplified view of a security methodology based on SAM. The methodology starts with a clearly stated mission statement and a system design (or architecture) for the system under consideration. The main steps of the methodology are:

- Build a model of the system that includes design/architecture of the system under consideration. The description of the system may be formal or informal.
- Develop a management approach that starts with identifying risks and establishing requirements. Based on this, corporate training for awareness is developed and roles/responsibilities are identified. We discussed the management approach in Section 12.5
- Do detailed risk analysis by developing attack trees that highlight possible attacks and measures of vulnerabilities. This also includes a model of attacker behavior with some idea of adversary objective, likelihood of the adversary having requisite capabilities, system access, and tolerating risk of detection. Attack trees, described in Section 12.8.3 can be the foundation of risk assessment.
- Develop countermeasures (risk mitigation approaches) to survive the attacks. The effectiveness of countermeasures is gauged according to cost, performance, functionality, and ease of use. The countermeasures, discussed in Section 12.8.4, use the security technologies we have reviewed in previous sections.
- Update the system on the basis of the likely attacks and countermeasures. The desired result of the system update is a stronger system together with some assurance evidence. This sounds obvious, but in some cases, the updated system is weaker because the updates are too complicated and leave several security holes. See section 12.8.5.
- Go back and reiterate the steps with more in-depth analysis or new attackers, different systems, different attack trees, etc.

The iterative process can become more formal as it proceeds. In the initial iterations, the process can be informal brainstorming sessions for systematically validating a security design for a system with respect to an adversary and likely vulnerabilities. In the later iterations, the process can become a security testbed with automated aids.

The main idea is to build survivable systems that can tolerate attacks by working through various attack trees systematically [Schneier 1999]. Success of this process is measurable by the overall advance of the system and its resistance to, or ability to deal with, further attacks that might not have

been considered explicitly. In other words, after a few iterations, the system should have been updated to the point where it can tolerate and survive almost anything (famous last words!).

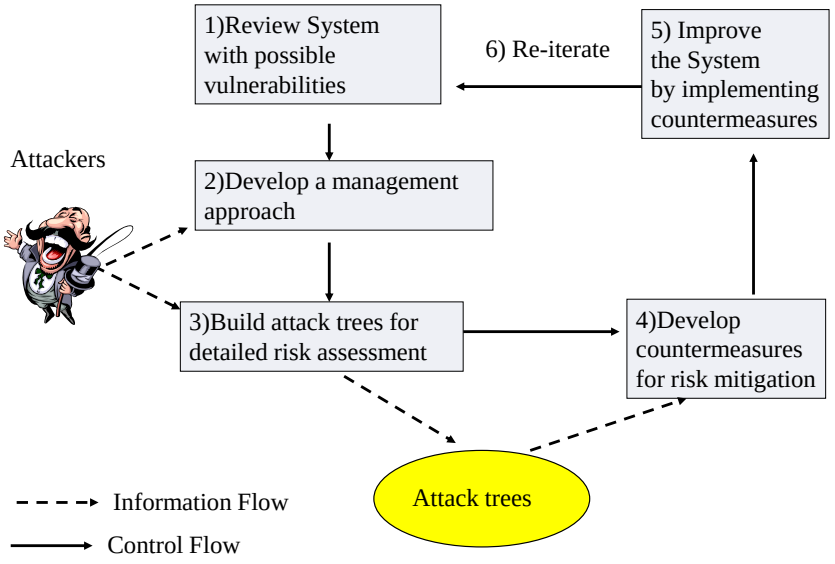


Figure 12-11: A Simple Methodology

12.8.3 Detailed Risk Analysis Through Attack Trees

Attack tree, introduced in the SAM methodology, is a convenient way to explore potential attacks and thoroughly examine the "attack space". An attack tree is simply a tree that is similar to a logical decision tree used to perform a systematic analysis of the attack space. The attack tree may be represented through a graph or some other means such as the extended outline mode of Microsoft Word. Attack trees are built by considering the "what," "where," "when," and "how" of attacking the system. For "what," an attacker can try to compromise system and data integrity, data confidentiality, or system availability. For "where", an attacker could attempt to do this inside a firewall (an internal attacker), at the firewall that separates the internal system from the public Internet, or on the public Internet. For "when", an attacker could mount the attack at any point in the lifecycle of the system, during system design and development, during system operation, or after the system has exceeded its useful life and is being discarded. The "how" of an attack deals with the mechanism used to execute the attack, such as eavesdropping.

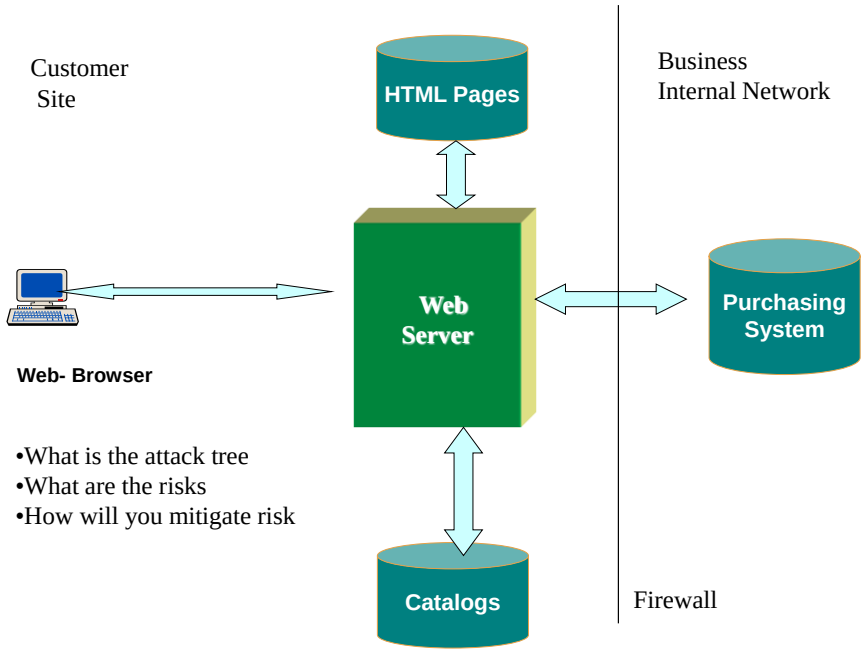


Figure 12-12: Online Purchasing System

Let us consider the online purchasing system shown in Figure 12-12. Attack trees can be built for each security concern: privacy, integrity, and availability. The following is a simple attack tree or piece of an attack tree for an online purchasing system:

C What: Confidentiality of data in the Purchasing System

C1 Where: Inside the Purchasing firewall (the when and how for later analysis)

C2 Where: Between Purchasing System and the Internet

C2.1 When: During system development

C2.1.1 How: Inadvertent human attack

C2.1.1.1 How: Coding error in purchasing access software

C2.1.2 How: Deliberate human attack

C2.1.2.1 How: Malicious code inserted in purchasing software

C2.2 When: During system operation

C2.2.1 How: Inadvertent human attack

C2.2.1.1 How: Sensitive information included in query[6]

C2.2.2 How: Deliberate human attack

C2.2.2.1 How: Sensitive information included in query

C2.2.3 How: Deliberate software attack

[6] The DARPA Information Assurance program is specifically focussed on deliberate attacks. However, an IA analyst may need to consider inadvertent "attacks," such as operator error, when designing a system.

C2.2.3.1 How: Sensitive information included in query

C2.2.3.2 How: Malicious software returned to purchasing with query results

C3 Where: On the Internet

In building attack trees, an analyst uses knowledge of the mission and knowledge of threats to determine which branches of a tree to explore in depth and which branches not to pursue or to prune early in the analysis. Naturally, judgement and knowledge are critical to the overall assurance of the system. Pruning a tree branch early can result in missing a critically important attack that develops deep in the tree. However, because attack trees would be extremely large for most systems, it is imperative that the trees be pruned as early as possible to keep the analysis manageable. The leaves of attack trees are indicators of attacks. Each attack characterizes the risk and should help in determining countermeasures. The table below lists all the attacks from the tree above.

Table 12-3: Confidentiality Attack Tree

Attack	Description
C2.1.1.1	Sensitive information is inadvertently released from Web server to the Internet due to a software coding error by a person during development of the purchasing system
C2.1.2.1	Sensitive information is released from purchasing to the Internet malicious software inserted in the purchasing system by a person during system development
C2.2.1.1	Purchasing system administrator inadvertently includes sensitive information in a query sent to the Internet
C2.2.2.1	Purchasing system administrator deliberately includes sensitive information in a query sent to the Internet
C2.2.3.1	Malicious software in the Business Internal Network includes sensitive information in software-generated queries sent to the Internet
C2.2.3.2	A response to a purchasing query returns malicious software in addition to the requested data. The malicious software installs itself in the Business Internal Network and leaks sensitive information to the Internet

In addition to confidentiality, there may be concerns about other security properties. Similar trees can be built for availability and integrity.

The attack tree shown above has been heavily pruned during construction. Prudence is needed before pruning because once pruned, the pruned attacks are excluded from future analysis. An important objective of the tree is to provide a heuristic for systematically considering attacks. A fully developed tree should have a complete listing of attacks but may be too big. Partial construction or pruning should be done carefully because the determination of which branches not to follow is based on a

conjecture of adversary behavior. It is quite difficult to guess adversary behavior -- they may act "rationally" according to commonly accepted "rational standards" or may act rashly or "out of character" in various cases or extreme conditions. It is quite possible adversaries may do the unexpected when it leads to the desired objective. It is best to be as comprehensive as possible in building an attack tree and prune those leaves that have the least likelihood and/or impact.

12.8.4 Development of Countermeasures and Risk Mitigation

The security designers develop countermeasures from their knowledge of the possible attacks and the security technologies and approaches that can help you to survive the attacks. There can be many countermeasures for a given attack, so a countermeasure is chosen based on cost, functionality, performance measures, ease of use, and effectiveness in dealing with the corresponding attack. For comparisons among countermeasures, it is desirable to choose the countermeasures that increase the costs and capabilities needed by the adversary and also increase the risk of being detected. The comparative analysis should lead to the most appropriate countermeasures to be implemented for risk mitigation. A major concern is to cohesively design the great many countermeasures that could cover the many likely attacks. The main challenge is: how will the countermeasures fit with each other and also with other functions of the system, while maintaining the mission of the system and keeping within reasonable developmental and operational costs? For a unified design approach, there will be a considerable number of value judgments about the countermeasures and how they fit into the overall system. For example, different cryptographic techniques can be used as countermeasures but need to be evaluated against cost and performance issues (e.g., symmetric versus asymmetric, key length impact, etc.).

12.8.5 Updating System Designs

The chosen countermeasures, when added to the current system design lead to an updated system design. This is also a creative rather than an algorithmic process. Retaining the mission and the main functions of the system should be key factors in updating the system design. There may be compromises so that some degradation of some functions of the upgraded system is acceptable. For example, strong encryption can degrade the performance of a system. The updated system will need to be assessed to determine whether it is acceptable or not. The assessment will include several factors such as preservation of the primary mission and confinement of implementation and operational costs of the redesigned system. If the updated system is not acceptable, then the information assurance process should be re-applied. Being unacceptable means that the system does not satisfy the functional, performance and security requirements. Even when the updated system is acceptable, it is extremely unlikely that it will be a perfect system that is completely impervious to all attacks.

12.8.6 Conclusions

Methodologies, such as the one discussed above, are guided steps that the user can follow systematically. The main advantage of this methodology is that it combines a management approach with technical solutions. It is also risk driven and the use of attack trees keeps us close to the central problems of securing systems against the attacks of malicious adversaries. The methodology is also iterative. It would be a good idea to automate some parts of the process. For example, the attack trees can be built from a system diagram. However, it is not clear how to automate the process of constructing countermeasures and picking the right ones that lead to a highly survivable system.

12.9 Chapter Case Study: Security for an Investment Company (GRQ)

12.9.1 Overview

Get Rich Quick (GRQ) is an investment firm that has been bought by XYZCorp. With partners in the US and Europe, GRQ wants its customers to access and update their account information and use some of the firm's financial analysis tools via the Internet. The goal of the company security system is to reduce the cost of customer service while ensuring customer and company data are secure *and* recoverable. While there are many design areas at play in this company, the focus here is on security. In particular, the objective is to develop a management approach by using the methodology described in the previous section. The approach should include:

- Security requirements
- Security risks
- Organizational structure, awareness policies, roles and responsibilities
- Risk analysis and key security technologies as countermeasures

12.9.2 System Conceptual Model

Let us start with a conceptual model of GRQ shown in Figure 12-13. The GRQ corporate web site consists of a user interface that connects to an Accounts Balance Program (ABP) that allows customers to view, update, and modify account information; a customer database that contains information about customers; an investment database that contains investment data; and other typical corporate applications and databases for payroll, accounts payable/receivable, etc. A corporate network will operate in the building, connected to the public Internet. A firewall protects the internal corporate resources. This simple model will be sufficient to get us started.

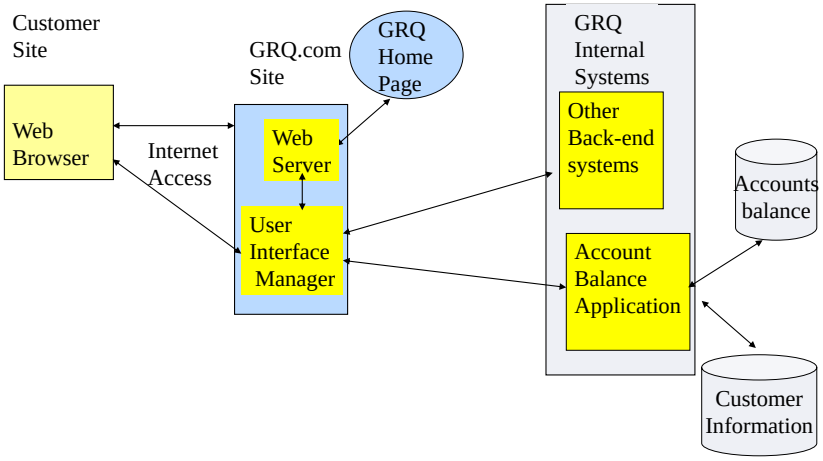


Figure 12-13: GRQ System Conceptual View

12.9.3 High Level Security Risks

With the implementation of a web portal for account management, GRQ will be potentially exposing itself to several security risks. Moving the customers away from telephone support and to the Internet has the obvious benefits (24x7 service, more support with fewer staff), however it introduces several risks, at high level, such as the following:

- Denial of service can happen due to any number of reasons. For example, network outages and network flooding, viruses, hackers and physical equipment problems could all deny users the ability to conduct business with GRQ.
- A web interface will open the company up to assaults from outside parties looking to disrupt GRQ's business.
- Access to individual accounts will increase the risk of manipulation of account data by customers.
- Unauthorized users may access information on customers, accounts and research information.
- Attacks using falsified authentication by hackers
- Middle-Man assaults by hackers to access information from one of the participants.
- Should any of these events occur, GRQ will risk damage to its internal systems and also could damage its reputation, resulting in significant loss of business.

12.9.4 Security Requirements

The GRQ Web customer interface will result in an unlimited number of potential interactions from an unlimited number of potential sources. The security requirements must not simply protect information from the wrong hands. They must also assure that the system survives successful attacks and keeps providing the services. This security plan increases the scope of protection to include a course of action in the event of a breach of security or failure of a certain level of service. Thus Information Assurance will be employed in order to accomplish both objectives. Specific security measures to be implemented must provide the following:

- Authentication and validation of internal and external users, i.e., unauthorized access to GRQ's corporate information systems should be prohibited.
- A simplified user interface for authentication to keep high user satisfaction.
- Rigorous access control so that only authorized users can access and modify the account information.
- Quick detection and denial of service to unverified users.
- Ensure that users are provided with 24x7 access.
- The system must be recoverable quickly, i.e. backup and recovery procedures must be in place.

- High availability of systems should be guaranteed through use of techniques such as Fragmentation, Redundancy, and Scattering (FRS).

By providing users with high availability, the need for large customer service resources will be reduced, lowering costs and achieving one of the major goals of the initiative.

12.9.5 Management Approach

A sound management approach is needed by the organization. Integral parts of the management approach are the organizational structure of the security team, the awareness policies distributed to the employees and the different roles and responsibilities of each team member.

12.9.5.1 Organizational Structure

One of the key elements of the successful implementation of the security program is to define the organizational structure. The organizational structure will ensure the effectiveness of GRQ security. The GRQ system administrator is responsible for creating and maintaining the security model. Different organizational units within the GRQ corporation will support the Account Balance Program (ABP) shown in Figure 12-13. A corporate Chief Security Officer (CSO) is responsible for the security and the integrity of all GRQ systems. It is the CSO's job to watch out for the Account Balance Program. The CIO will be more hands off and will work with the CSO and other managers who manage the GRQ resources such as the network, databases, and applications.

As the organization grows, an information security coordinator in every business unit within GRQ may be very effective (initially security coordination may be just a function assigned to an area manager). While CSO would be responsible for the overall implementation and running of the security program, the information security coordinators provide the day-to-day decisions within each business unit, and then report to the CSO. Also, the supporting level personnel would bear responsibility for implementing the overall security awareness program.

12.9.5.2 Security Awareness Approach

GRQ's CSO will have to raise company awareness of the security, especially of the ABP. This may be done through required courses or internal marketing within the firm via emails and written memos. The firm should clearly inform the employees, through the training, on what they may access and the penalty for any security breach. The training, procedures and guidelines will help GRQ to be able to maintain the desired level of security. Specifically, the following approach is suggested before the implementation of security measures and technologies:

- The security plan should be presented to all GRQ existing employees, as well as to the newcomers on a regular basis.
- All current training needs of GRQ employees should be identified, and presented to the CIO by the CSO. If possible, every business unit security coordinator should contribute to this report.
- The security plan should be presented, after approval, on a regular basis, such as twice a year, to ensure that all newcomers as well as existing employees are aware of the plan.

- Several companies can provide security training. Examples of the companies are the Computer Security Institute and Learning Tree, Inc.

Special control measures will be needed to keep the information security awareness plan alive and effective. These could include, among others, unexpected "security audits" of security procedures, after hour visits to IT department employee offices to check if the required security procedures are being performed; and regular (twice a year) organization of "Security Alert Days" with different security issues in agenda, such as viruses, etc.

12.9.5.3 Roles and Responsibilities

The roles and responsibilities of the different team members need to be established. Of course, the CSO is responsible for maintaining the overall coordination of the GRQ's security. It is expected that the CSO will disperse the responsibilities among various employees. This is to ensure that no single person has too much control over the system. This benefits the firm in that it better maintains security, and allows for protection from vulnerability if that person were to leave the job. The best way for GRQ to establish this is to rotate the responsibilities among the different team members.

The level of security in every business unit will be implemented according to the roles and responsibilities of user segmentation. The factors to be included are: level of awareness needed versus achieved; job category and specific job function; familiarity with systems; and areas of expertise.

12.9.6 Detailed Risk Analysis and Countermeasures

The main idea of detailed risk analysis is not only to identify the threats but also to develop countermeasures to combat them. The main objectives of this analysis are:

To identify potential threats and describe them as real possible attacks on different GRQ systems in different business units.

To describe the possible behavior of the systems under attack to identify the gaps in defense and to develop the defense strategy.

To determine the time lapse between the attack and recovery; frequency of the attacks; possible places of the attacks, etc.

To evaluate the effectiveness of the defense measures and to assist in the strengthening of the systems.

To connect the management approach to an implementation.

To determine the possible attacks on the system, we first review the GRQ system design and then develop "attack trees" that are built by considering "what", "where", "when", and "how" of attacking the system. For purpose of illustration, we will use the conceptual view presented in Figure 12-13. Before developing an attack tree, let us look at the what, where, when, and how at high level.

What: Parts of the architecture could be at risk for security breaches?

Web Server and network. Denial of service, hacked entry with intent to do harm or gain information

Institutional Databases: This includes all user personal and financial information as well as corporate information and directory database

Routers and gateways: could be passageways into the Extranet and Intranet

Where: Will these breaches or attempts take place?

Hackers may pinpoint individual systems or components

Localized to the web servers or network routers or gateways

Internal espionage and eavesdropping

When: Would a possible attack take place and with what frequency?

During normal operations

After hours; batching or back-up

While in development, rollout or switch over

How: Will these risks be assessed and mitigated

Assessed by a total loss basis

Multiple redundancies will help secure system

Attack trees can be built for each security concern: privacy, integrity, and availability. An example of building the privacy attack tree for GRQ confidentiality of data is presented below (note that this is very similar to the attack tree developed for an online purchasing system in Section 12.8.3):

C What: Confidentiality of data in the GRQ System

C1 Where: Inside the GRQ firewall

C2 Where: Between GRQ Web Server and the Internet

C2.1 When: During system development

C2.1.1 How: Inadvertent human attack

C2.2 When: During system operation

C2.2.1 How: Inadvertent human attack

C2.2.1.1 How: Sensitive information included in query

C2.2.2 How: Deliberate human attack

C2.2.2.1 How: Sensitive information included in query

C2.2.3 How: Deliberate software attack

C2.2.3.1 How: Tapping the line, especially in case of a

wireless network

C3 Where: On the Internet

This tree focuses on confidentiality attacks against the interface between GRQ server and the Internet. The attacks are the last leaves of the tree (C2.1.1, C2.2.1.1, C2.2.2.1, C2.2.3.1).

In addition to confidentiality, there may be concerns about other security properties. The tree below explores some aspects of service assurance due to denial of service on the GRQ system:

S1. What. Service Assurance of System Threatened – Denial-of-Service Attack on System

S1.1. Where. Individual Component or Subsystem

S1.1.1.Where. End system

S1.1.1.1. When. During network operation

S1.1.1.1.1. How. Passive attack

S1.1.1.1.1.1. How. Eavesdropping

S1.1.1.1.2. How. Active attack

S1.1.1.1.2.1. How. Illegal logon or system entry as user or root to cause denial of service

S1.1.1.1.2.2. How. Dial-port flooding

S1.1.1.1.2.3. How. Shutdown

S1.1.1.2. When. During development …

S1.1.2. Where. Router

S1.1.2.1. When. During network operation

S1.1.2.1.1. How. Passive attack

S1.1.2.1.1.1. How. Eavesdropping (preparing for DoS attacks)

S1.1.2.1.2. How, Active attack

S1.1.2.1.2.1. How. Attacks on routing protocols

S1.1.2.2. When. During development …

The attack tree below shows some aspects of availability:

A What: Availability of GRQ System access to the Internet

A1 Where: Inside the GRQ firewall

A2 Where: Between GRQ Server and the Internet

A2.1 When: During system development

A2.1.1 How: Insert time-bombs in GRQ network or flood the network

A2.2 When: During system operation

A2.2.1 How: Disrupt critical network services such as DNS

A2.2.2 How: Flood Web access ports

A2.2.3 How: Introduce malicious code into GRQ

A3 Where: On the Internet

Both the confidentiality tree and the availability tree shown above have been heavily pruned during construction. For example, the "where" branch of the availability tree that examines attacks occurring inside the GRQ firewall is ignored. This may or may not be a prudent decision. By pruning this branch early in the analysis and excluding any such attacks, the remainder of the analysis will not consider this option and countermeasures will not be chosen against this type of attack.

12.9.7 Choosing Enabling Security Technologies as Countermeasures

GRQ will establish some key attributes of the security architecture for the Accounts Balance Program (ABP) to address the risks identified above (high level as well as low level) through attack trees. The main elements of the countermeasures are use of encryption, Virtual Private Networks (VPNs), firewalls, authorization, and general attempts to minimize the points of access to critical databases and applications. All external users, customers, remote employees, global partners and suppliers, have to receive certificates in order to access the corporate information. Specifically, the countermeasures should include the following:

Modern cryptographic techniques should be employed by GRQ for the encryption/ decryption of its data between customers. Specifically, GRQ should use the symmetric key encryption. For asymmetric encryption, digital signatures and integrity through message digests, GRQ may also want to explore using Public Key Infrastructure (PKI) as a means for encrypting/decrypting data from its applications. Packages such as PGP or Kerberos may be employed.

The platform for GRQ's Extranet system has to be highly controlled. This means that only GRQ's data center personnel has physical access to GRQ's application server & network equipment. If a business partner owns a piece of equipment it is to be shared between both organizations.

Firewalls will be used to localize the different areas of the firm's security architecture. The firewall will separate the corporate web site from the ABP, the customer database as well as the investment databases.

Secure network connectivity has to be provided using a dedicated line or Virtual Private Network (VPN). VPNs provide encryption and authentication features over public networks for secure communications. Before a GRQ VPN device communicates with another it first must establish a password. Authentication systems like GRQ's are based on digital certificates that are more secure than a password-based authentication. GRQ would want to ensure that its network is semi-private, meaning that only business partners have access to the network. GRQ would want a network effectively capable of detecting intrusions.

Access to applications should be private. Users must be authenticated & authorized to perform operations depending on their rights. Application users have to be uniquely identified using adequate authentication techniques. Accountability can be accomplished by identifying and authenticating users of the system & subsequently tracking actions on the system to the user who initiated them.

There should be single user name and logon. Once a person is logged on to the GRQ system, they will not have to log on again.

GRQ will use authentication and authorization. The GRQ security architecture must enforce user accountability. At the network level, it is very difficult to achieve this due to the proxy servers, application gateway, firewalls, and address translation.

The users' authorization to the network will be stored in a web server/directory server. The GRQ users' rights to what files and directories they can access, will all be located and accessed from this server. This is sufficient for static web content security. The application will provide further authentication and authorization once the network access has been granted to the GRQ employee.

Authorization rights have to adhere to the least-privilege principal. This principal is the practice of restricting a user's access (DBMS updates or remote administration), or type of access (read, write, execute, delete) to the minimum level necessary to perform a job. It is a conservative approach to granting user rights.

GRQ should also take Non-repudiation (NR) into consideration. Non-repudiation is the ability to provide proof of the origin or delivery of data. NR protects both senders & recipients in a data interchange. A receiver (GRQ) cannot say that he/she never received the data and the sender (customer) cannot say that he/ she never sent any data.

For quality of Service, GRQ should ensure availability, latency, bandwidth & response time of its Extranet system. Quality of service should not be compromised in any security system. Servers should be physically secured & back-up power sources should be available.

12.10 Additional Case Studies and Examples of Security

The following case studies were collected by the students of my class on "Security and Information Assurance", taught at the Fordham Graduate School of Business. The names of the contributors are Tracy Brown, Rebeca Cates, Michael Collins, John Coyle, Michael Fazio, Christopher Freiler, John Geraghty, Rita Ghei, Kevin Kline, Mary McNally, Tammie Min, Thomas McGinley, John Morris, Pankaj Navathe, Mitch Rothman, and Viktoryia Petrashova.

12.10.1 Standard Chartered Bank, Americas Front Office: Remote Access Disaster Recovery/Business Continuity Plan

12.10.1.1 Context & Background

Standard Chartered is a British emerging markets bank with a focus on Asia, Africa & the Middle East. Its wholesale banking unit, known as Corporate & Institutional is managed globally from Singapore. A large portion of the bank's Asian business is driven from the New York (Americas) office. From New York, Standard Chartered helps Fortune 500 customers like Coca Cola, Wal-Mart & Disney finance their operations in Asia.

The bank relies upon a Lotus Notes based system for email communication and for housing the bulk of the bank's credit & revenue databases.

Before September 11, 2001, it was very difficult to use the email application of Lotus Notes successfully from a remote location. Remote access to critical databases was virtually impossible.

After September 11, 2001, Standard Chartered Bank lost its Americas office in 7 World Trade Center. Everyone survived fortunately, but just about every computer & server was lost. While the operations department had a disaster recovery plan established to continue business in a limited capacity after the disaster, the Americas front office (approximately 80 people) were left sharing 2 computers at the small disaster recovery site in Jersey City.

It took several months for the front office to get back to business as usual. A decision was made by local management to purchase laptops for every critical front office employee to keep at home. The laptop strategy would distribute risk geographically & was considered to be safer (people wouldn't

have to leave home) in the event of another tragedy. In addition, people who need to work from home would now be able to gain access to bank systems from home.

The strategy, while very compelling, leaves some unanswered security questions. How can the bank insure that its valuable information is secure with the new remote strategy?

12.10.1.2 Security Problem

Credit limits are proposed and approved via a Lotus Notes based system called Fasttrack. Digital signatures authenticate that the person proposing/approving limits are who they claim to be. What, if anything needs to be modified with the current system as we move toward remote access?

Connectivity: What are the security risks of transferring confidential bank data via phone lines (dial-up connection) versus DSL or Cable Modem?

12.10.1.3 What is at risk?

The bank's confidential customer data is at risk. It could be detrimental if one of the customers or competitors were able to access this information. Also, there is a concern with regard to rights/credit approval authorities getting into the wrong hands (e.g., they wouldn't want someone with access to the Regional Credit Officer's laptop to be able to approve a $ 100 million loan arbitrarily just because he or she had physical access to his laptop at home).

12.10.1.4 Problem Categorization

Management: Who can access what systems and databases remotely?

Network: What concerns are there with regard to accessing the bank's LAN/GWAN network remotely?

Web: A soft copy of the bank's Disaster Recovery Plan will be kept online along with employee contact information on a password protected website. There are a number of ASP's that the front office uses: Moodys.com, etc. Are there any remote access issues there? Will everyone access the bank's systems via the web or via a dial up connection?

Database: Should there be any limits with regard to database remote access (printing limitations, read only)?

12.10.1.5 Solution Approach - What should be done?

Security Awareness: Security policies need to be established & implemented. Procedures & guidelines must be put in place to enforce the policies. Bank managers & employees need to be told about these policies.

One person from each business unit has to be put in charge of his or her team's security management program. Checks need to be made on the program periodically.

After key threats are identified, a training program should be implemented to teach users of the new remote access system how they can protect the bank from becoming vulnerable to newly identified threats associated with remote access.

Standard Chartered (Americas) needs to develop a security architecture that satisfies the requirements in terms of organizational policies, awareness program & enabling technologies.

The bank is looking at a remote networking application called Shiva that offers users access to bank systems via the internet. The Shiva system is expected to ensure security of bank systems through the use of single user logons & strong authentication. Shiva also enables administrators to centrally manage the access rights of remote users.

12.10.2 CNN Denial of Service Attack

The cable network CNN was victimized by a denial of service attack (a denial of service attack is a situation where hackers flood a system with numerous useless messages that tie up the system resulting in a possible loss of service). The CNN website (www.cnn.com) was shut down for approximately two hours as a result of hackers who flooded the servers thus rendering the website inaccessible to the general public. At the time, these website attacks were considered to be the most severe to date. These incidents influenced the need for greater study into the field of information assurance and security.

In terms of CNN and media concerns, the attack proved that media companies were also susceptible to electronic sabotage. In essence, these attacks threatened the competitive advantage that news websites possess over other forms of news media -- the ability to break news at all times of the day. If CNN's website could be put out of commission for extended periods of time, people would no longer rely on cnn.com to report news in a timely manner. This issue was of paramount importance to CNN. Also related to CNN and denial of service was the loss of advertising revenue on the site. If viewers are kept off the site, advertisers can demand a refund or credit on their account. CNN will then be required to make-good on the advertising space that was lost due to the site shutdown.

Another concern that CNN and other news websites must have considered after the assault was that if hackers could keep people out of their websites, could the hackers also break in and revise or delete news copy? This is an even graver concern for media organizations. CNN would lose a tremendous amount of credibility if people could access their website and change news stories. For this reason, CNN must implement tighter security around their website to safeguard against hackers that can disrupt and discredit their hard work.

Let us look at denial of service somewhat closely. Unlike most other types of "hacks", a denial of service attack will not usually result in the theft of information or other specific security losses. Instead, a denial of service attack results in either loss of service or extremely slow response time on a website. This can be extremely damaging because it can frustrate consumers, hurt the reputation of a website in terms of reliability and if the site is revenue generating it can potentially result in the loss of sales.

Specifically what happens is that a constant stream of requests is sent to a target with the intention of overloading it. The information is sent to the target in small packets of data called "pings" which are used as a signal between two computers. The attacker sending the "pings" lies about their real address so the target computer is unable to return the ping and make any connection. The result of this is an enormous amount of junk traffic floods the computer beyond its capacity causing the website to become unavailable.

The denial of service attack that hit CNN Interactive on February 8, 2000 had serious impact on the site and overall caused a lot of risk to the organization in terms of ensuring their customer base that there site is secure and reliable. CNN Interactive provides content that needs to be very accurate and as up to date as possible in order to meet the needs of their customers. On this day the content that was served was very inconsistent, out of date and overall the site was extremely slow. In all the hackers affected the news site operations for nearly two hours. The situation could have been a lot

worse for CNN had it lasted longer than two hours. When sites like CNN are inaccessible it means a loss of both integrity and revenue for the firm.

In the instance of a company heavily dependent on its website for revenue and survival, how management responds to a DoS attack may well determine if the company can survive. Since the network is by definition overloaded by a DoS attack, management must rapidly recognize that the attack has begun and either eliminate the superfluous packets choking their network lifeline or reroute the traffic to a backup network. The databases that companies use should remain unaffected by the DoS attack except inasmuch as the people who need the information within the database will be unable to access or use it. Unless there is something more nefarious attached to the DoS packets, once the traffic is sorted out, cleaning up the database is relatively straightforward albeit time consuming and therefore expensive. Applications accessed through the network affected by the DoS attack will be slowed if not outright crashed by the overwhelming network traffic. If the applications are critical to running company operations, especially in virtual companies such as Sun Microsystems, the entire company, not just the website, will come to a halt.

Though the concept of the DoS attack is straightforward, it seems that there are only three possible strategies to prevent DoS attacks in the future. The first strategy involves using address filtering at edge routers to detect and prevent large numbers of fake packets from entering the network. However, such a fix is expensive and thus far has not been widely used due to fears of its longevity. Second, UUNET and Cisco are working on reverse path forwarding protocols to be sure packets are coming from appropriate networks. Unfortunately, though RPF will be cheaper, it is still on the drawing board. The third and only truly foolproof manner of preventing DoS attacks is to beat the hackers to the punch—by shutting down your own website and handling all transactions in person. But this is not acceptable.

12.10.3 Analysis of IT Security in Pharmaceutical Trials

Pharmaceutical companies invest hundreds of millions of dollars each year developing and testing drugs for introduction to the public. Complex scientific innovations are the basis for countless studies and hopefully introduction of a new drug in the marketplace.

The lifecycle of a newly innovated drug includes screening of a potential testing population, drug testing (including such things as dosing, blood screening, urine screening – many of which are bar coded), recording of adverse affects from the drug, checking the data for errors, preparing reports of results, and performing statistical analysis to determine the final results of the tests. To add more complexity to the pharmaceutical studies, several studies on differing populations must occur to assure that effects of the drugs on people of different ages, gender, & ethnicity are positive.

In this life cycle, the integrity of the studies' data is compromised because multitudes of hands touch the data before a "final" product, called an NDA, can be submitted to the FDA for approval. In addition to the risk of data integrity, pharmaceutical companies also have the risk of having their ideas "pirated" by other companies looking to profit off of someone else's intellectual property.

Because of the extraordinary legal, regulatory and financial exposure of the global pharmaceutical manufacturer to both U.S. and European Union "Security, Access and Control " (SAC) regulations, no component of the global clinical trials application is more important than the security model.

In the United States, companies spend on average $240 million on clinical trials as a drug passes through the Food and Drug Administration (FDA) for final approval. Pharmaceutical companies have billions of dollars at risk if they cannot correctly match a pharmaceutical product to a target patient population as quickly as possible. Lost earnings and revenue by pharmaceutical companies can also occur when a very small segment of a target patient population experiences side effects from an otherwise effective pharmaceutical.

Healthcare SAC standards in the U.S. are defined by the interaction of state laws with the Health Insurance Portability and Accountability Act (HIPAA). The Act implies that computerized healthcare information systems must have certain features and functions that address the proposed requirements of HIPAA. These requirements are enterprise-wide. That is, above the level of each individual clinical trial application. These requirements must be satisfied by the pharmaceutical manufacturer at the enterprise-level. For example, a single global authentication system for each user should be employed by the pharmaceutical manufacturer for all clinical trial applications that access a central collection of data.

The functions required by clinical trials applications to ensure integrity, security, reliability of the application and its respective data are listed below. These functions in unison create a secure application supporting privacy and security policies that can be applied to an individual or a single transaction:

Personal Security, Authentication, Role-based Activity, Context-based Activity, Security Policies, Authorization, Ownership, Integrity, Auditable Activity, Security of Data, Message Integrity

The problems pharmaceutical companies face in light of this idea of a global ICH (International Conference on Harmonization) compliant clinical trial architecture can be categorized as follows:

Management – Grouping users into roles; establishing unique IDs & passwords for all users within a role; setting security policies based on a user's role; authorizing and creating an audit log as dictated by HIPAA; adherence to security standards set by state/country legal authorities

Application/Data – Determine what applications/data can be accessed and which transactions executed according to user ID/role; show history of access and modifications according to user ID/role; protection of user's data (including work in progress) until data is ready for "general" access; protection of data integrity from unauthorized access (internal and external), modifications, and other SAC threats (including hackers and viruses)

Network/Web – Ability to protect network messages, data, and applications while transmitting between clinical trial sites; protect data and applications from outsiders (hackers, viruses, etc.)

The solution to expediting the market launch of new drugs is to replace complex multiple submissions of new drug approval with a single folder submission. A single folder submission will save a significant amount of time and resources which will facilitate the approval and the launch of new drugs. The single folder submission will follow the approved Common Technical Document (CTD) guidelines as well as meet ICH guidelines on efficacy, quality and safety. In addition to the submission's conformance to the CTD and ICH guidelines, the submission must be available online and continuously updated.

The technologies that would best support the single folder submission includes "global telecommunications infrastructure, high reliability databases and pervasive access to real-time clinical and ICH dossier data" made up of "best of breed" components. The EMC "E-Infostructure" is the recommended model architecture. Such an architecture combines physical, functional, connectivity, and security layers of hardware and software. The Enterprise Storage Network, ESN, and the

Database Management System(s) will make up the storage management layer and will be used to store all information pertaining to the single folder submissions. Oracle Parallel Server or IBM DB2 will support databases for the central repository. Computational, analytical and application-oriented servers which will be used for analyzing information will be made up of highly scalable central processors with high-availability operating systems. Secondary applications will be supported by processors such as Intel based processors.

As for connectivity, the enterprise storage management software and EMC Symmetrix will integrate all operating systems and database storage into a uniform central repository. The ESN will also serve as the connectivity layer. It will use the information connections, Connectrix and Celerra to connect multiple primary and secondary operating systems to enterprise application and data management platforms. Devices such as a mainframe or a PDA will be able to obtain information through wide and local area access as well as wireless access. TimeFinder will refresh the data warehouse without disruption and EMC's SRDF can protect clinical trial and other critical data to prevent any delay in speed to market and can also be used for disaster recovery and information mobility. There is continuous and secure availability of reporting, decision support, web access and availability of the databases. Each of these capabilities can be replicated to enhance reliability and availability of the operational systems.

12.10.4 Example: Security in Healthcare

A general goal of healthcare information systems is to provide open but secure access to information. The main problem is that many systems are old (one of the oldest in the industry) and not well designed. The healthcare professionals are not sure how to deal with the Internet -- the idea of having healthcare information over the web terrifies some folks. To further complicate matters, the healthcare industry is heavily regulated and new privacy laws are being introduced regularly. Customer distrust of the healthcare providers is also quite high.

An interesting survey was conducted by the Consumer's Union in 1999. The survey found the following information:

Willingness to share. Largely unwilling, only to researchers (50%)

Perceived threat. largest: hackers (70%)

Effective Safeguards. Punishments and fines were perceived as the most effective. Requiring specific permission from the owner and using technology for better security were also deemed effective.

The Key points of this survey are that the consumers do not trust health plans or providers, consumers do not trust computers, and that the consumers will compromise quality of healthcare for privacy.

This raises some interesting questions such as the following: can the opinions of consumers be turned around, can the increased use of the Internet serve as an example (many consumers feel comfortable giving credit card info over the Internet), and can improved technical solutions win consumer confidence?

12.11 Chapter Summary

Security architectures are needed to protect the corporate IT and physical assets by employing the latest security technologies to respond to external factors and organizational requirements. This chapter has introduced basic security terms and views and given an overview of security at several levels. A short overview of management issues is presented for completeness. A wide range of security technologies ranging from public/private key encryptions to digital certificates and ACLs are currently available to address the authentication, protection, authorization, and accountability aspects of security. Table 12-4 shows a mapping of various security technologies to security needs (e.g., which technologies address which needs). A methodology was introduced to tie the different technical and organizational pieces together into a series of procedural steps and a detailed case study was discussed to show how this methodology can be used to secure a financial institution.

Table 12-4: Security Considerations - Mapping Technologies to Needs

Technologies	Privacy	Authentication and Authorization	Integrity	Accountability (Non-repudiation)
Encryption	X	X		
Password protection	X	X		
Digital signatures		X		
Message Digest			X	
Digital certificates	X	X		
ACLs		X		
Audit trails				X

12.12 Review Questions and Exercises

1) What are the key ingredients of a security management approach? Which ones are absolutely essential and which ones are nice to have?

2) List some generic security requirements that apply to the current breed of eBusiness applications.

3) What are the tradeoffs between symmetric and asymmetric cryptography? When will you use what?

4) Use the brief RSA description given in this chapter (sidebar "RSA -- A Brief Description") and work through it by using the following:

a) Two prime numbers (p=7, q=13) and for message text = 12

b) Two prime numbers (p=7, q=17) and for message text = 8

5) What is the most common scenario that combines symmetric as well as asymmetric cryptography? Give an example of a security package that uses this scenario.

6) Choose a security package, install it, and conduct some very simple tests to understand how the basic cryptography works. Possible candidates are PGP (www.pgpi.com) and SSL (discussed in the next chapter).

7) How can attack trees be used to identify risks, analyze them, and then mitigate them? Give an example.

12.13 Main References

Aron, M., "Better Security Needed for B2B", Australasian Business Intelligence Nov 4, 2002

Deswarte, Y., Fabre, J.-C., Fray, J., Powell, D. and Ranea, P., "SATURNE: A distributed computing system which tolerates faults and intrusions", Workshop on future trends of distributed computing systems in 1990's, 1988 pp329-338.

Deswarte, Y., Blain, L. and Fabre, J.-C., "Intrusion Tolerance in distributed systems", IEEE Symposium on Research in Security and Privacy 1991, pp110-121.

Dickenson, G., "Wyndham Worldwide v. Federal Trade Commission : The Developing Parameters of the FTC's Data Security Requirements", JOURNAL OF INTERNET LAW, volume 19, number 6, 9-19., 2015

Ellis. J.H., "The Possibility of Secure Non-Secret Digital Encryption", CESG Report, January 1970.

Gollman, D., "Computer Security", Wiley, 2011

Gove, R., "Fundamentals of Cryptography and Encryption", published in [Tipton 2010].

Howard, John D. "An Analysis of Security Incidents on the Internet 1989 – 1995", Carnegie Mellon University, 1998.

Kienzle, D., "Practical Computer Security Analysis", University of Virginia., January 1998.

Murray, W., "Enterprise Security Architectures", published in [Tipton 2010].

Oppliger, R., "Security Technologies for the World Wide Web", Artech, 2000.

Ozier, W., "Risk Analysis and Assessment", published in [Tipton 2010].

Paar, C and Pelzl, J., "Understanding Cryptography: A Textbook for Students and Practitioners", Springer, 2010

Peltier, T., "Security Awareness Program", published in [Tipton 2010].

Schneier, B., "Attack trees", Dr. Dobb's Journal, Dec.,1999.

Sood, S., "A combined approach to ensure data security in cloud computing", Journal of Network and computer applications, 2012.

Stallings, W., "Network Security Essentials", Prentice Hall, 2000.

Stein, L., "Web security: A step-by-step Reference Guide" Addison Wesley, 1998.

Tipton, H. and Kraus, M., editors, "Information Security Management Handbook", 6th Edition, Auerbach, 2010.

Vacca, J., "Single Sign-on for the Enterprise", published in [Tipton 2010].

Verisign White papers on e-Commerce Security (www.verisign.com).

12.14 Appendix A: More on Cryptography

12.14.1 Asymmetric Key Cryptography – A Closer Look

12.14.1.1 Main Features of Asymmetric Cryptography

The concept of public key cryptography was introduced by Whitfield Diffie and Martin Hollman in 1975. However, there is now evidence that the British Secret Service invented this technique a few years before Diffie and Hollman [Ellis 1970]. The reason why this was not known by Diffie and Hollman is that this technique was kept a military secret with no thought towards commercialization.

In an asymmetric (public) key system every user has two keys: a public key e that everyone knows (it is used for encryption) and private key d that only the receiver knows (it is used for decryption). Even if someone knows the key used for encryption, this key cannot be used for decryption. This scheme is very suitable for the Internet because anyone can send a message, but only the intended user can decrypt and read the message. Given:

M: Message (plain text)

E: algorithm uses public key (e) - everyone knows it (used for encryption)

D: algorithm uses private key (d) - only receiver knows (used for decryption)

The following relationships hold

1. D(E(M))= M

2. E(D(M)) = M

3. Given E, it is not possible to determine D

4. Given D, it is not possible to determine E

Each person (A, B,,) has two keys (e_A , d_A), (e_B , d_B). When the sender B wants to send a message M to A, it first encrypts M by using e_A, and sends it to A. After receiving the encrypted message, A decrypts it by using d_B This process uses principle 1. As we will see later, digital signature uses principle 2.

The main idea is that you publish your public key to the world while keeping your private key secret. Anyone with a copy of your public key can then encrypt information that only you can read. Even people you have never met. It is computationally infeasible to deduce the private key from the public key. Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the information. RSA (named for its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman) is the best known public key cryptosystem. Other examples of public-key cryptosystems are Elgamal (named for its inventor, Taher Elgamal), Diffie-Hollman (named for its inventors), and DSA, the Digital Signature Algorithm (invented by David Kravitz).

12.14.1.2 Symmetric Versus Asymmetric Cryptography

Conventional symmetric encryption has several benefits. It is very fast and is especially useful for encrypting data that does not need to be sent over a network. Thus you can encrypt a sensitive file and only the people knowing the private key can read it. However, the private key must be shared between senders and receivers if the data is to be sent and shared by multiple users. As the population of users grows, some (dishonest) users can tell others about the private key or send the encryption card to an accomplice. Remotely located users must trust a courier or some other secure communication medium to prevent the disclosure of the secret key during transmission. Anyone who intercepts the key in transit can later read, modify, and forge all information encrypted or authenticated with that key. The persistent problem with conventional encryption is *key distribution:* how do you get the key to the recipient without someone intercepting it?

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. The need for expensive secure channels and key distribution is eliminated. These factors limited private key systems to government and large banks. Public key encryption provides strong cryptography to the general public masses.

While public key systems solve the problem of key management, they are usually significantly slower than private key systems. The RSA (Rivest, Shamir and Adleman) algorithm, described briefly in the following sidebar, is between one hundred to one thousand times slower than DES [Stein 1998]. Due to this, the public key encryption is not typically used on the entire message. Instead, just the key is encrypted by using RSA. This compromise, known as digital envelope, is discussed later in Section 12.14.4.

Security Key Considerations: Performance Versus Protection

A key is a value that is used by a cryptographic algorithm to produce a ciphertext. In modern cryptography, keys are very large numbers, usually measured in bits. The larger the key (bits in key), the more secure the ciphertext. However, the algorithms used for symmetric versus asymmetric cryptography are very different and thus a key size of n bits in symmetric crypto is not equivalent to n bit key in asymmetric crypto. For example, a conventional 80-bit key has the equivalent strength of a 1024-bit public key.

Although the public and private keys of asymmetric systems are mathematically related, it is very

difficult to derive the private key given only the public key; however, deriving the private key is always possible given enough time and computing power. Thus it is very important to pick keys of the right size so that they are large enough to be secure but small enough to be applied quickly. Larger keys are cryptographically more secure for a longer period of time because they are harder to guess. This does depend of course on the determination of attackers and the fast computer systems of the future. For a long time, a 56-bit symmetric key was considered extremely safe but now larger key sizes (128 bit) are being preferred.

Keys are stored in encrypted form. Systems such as PGP (Pretty Good Privacy) store the keys in two files, called *keyrings*, on your hard disk. The public keyring is used to store the public keys of your recipients and the private keyring is used to store your private keys. These keyrings are important -- if you lose your private keyring, it is time to close your tent and go home because you cannot decrypt any information encrypted to keys on that ring.

Protection of the keys that in turn are used to protect the assets is extremely important. Private keys and shared secrets, once acquired, must be protected. End-to-end security must include consideration of the security of the end user device. Private keys stored on a personal computer disk file may be stolen via access to the file system or outright theft of the device. Security can be enhanced by the use of smart cards. Another approach is to use a security chip embedded in end user systems. In addition, server-side hardware devices can provide tamper resistant key storage as well as assistance for encrypting and decrypting messages and public/private key operations, etc. that require heavy computational load.

Sharing of private keys, although not a recommended practice, is necessary at times. For example, Corporate Signing Keys are private keys used by a company to sign legal documents or press releases. In such a case, it may be better for multiple members of the company to have access to the private key. It may be worthwhile to split the key, called ***"key splitting***", among multiple people in such a way that multiple people must present a piece of the key in order to reconstitute it to a usable condition. We have all seen examples of key splitting in movies where each person knows only part of a secret code. If a secure network connection is used during the reconstitution process, the partial key holders need not be physically present in order to rejoin the key.

12.14.2 Digital Signatures for Authentication

A major appeal of public key cryptography is that it provides a method for employing *digital signatures.* A digital signature serves the same purpose as a handwritten signature -- it enables the recipient of information to verify the authenticity of the information's origin, and also verify that the information is intact. Thus, public key digital signatures provide *authentication* and data *integrity*. A digital signature also provides *non-repudiation,* i.e., it prevents the sender from claiming that he or she did not actually send the information. These features are as important to cryptography as privacy.

Digital signature technology is used to authenticate the source of a message. It is essentially the same as a public key system except that the order in which the keys are applied is reversed. A sender "signs" the message by applying his private key to it. The sender sends the message and the signature to the receiver. The receiver checks the signature by applying the sender's public key to it. If the receiver gets the original message back, he is sure that the message was signed by the sender's private key, and therefore, was sent by the sender himself. In essence, a digital signature is a block of data created by applying a cryptographic signing algorithm to some data using the signer's private key.

Digital signatures may be used to authenticate the source of the message and to assure message recipients that no one has tampered with a message since the time it was sent by the signer

Let us consider a scenario where Joe wants to send his signature to Pat as shown in Figure 12-14. Joe creates a short message ("I am Joe") as a signature S and encrypts his signature with his private key (encrypts by using d_J). The created ciphertext $d_j(S)$ is sent to Pat. On arrival, Pat decrypts it by using Joe's public key e_J. This means that the signature is Joe's signature because the ciphertext can only be decrypted if sent from the right person. This uses Principle 2 as discussed in Section 12.14, i.e., E(D(M)) = M. The main idea is that the receiver knows that the message could not be decrypted unless encrypted by D.

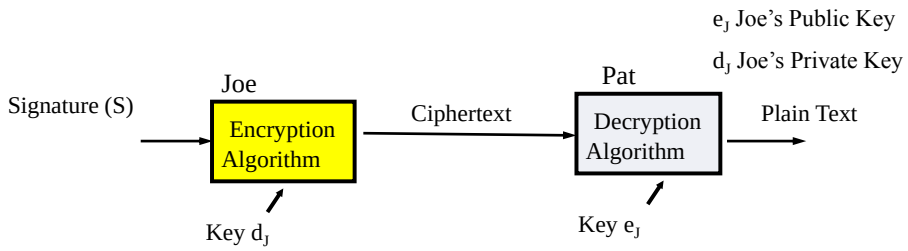


Figure 12-14: Digital Signature

Let us now consider a more complicated, albeit more practical scenario. Let us assume that Joe's lawyer sends a document (Joe's will W "Pat will inherit my estate") and asks Joe to sign it and send it to Pat. Let us assume that Joe has the private-public key pair (d_J, e_J) and Pat has the private-public key pair (d_P, e_P) Once again, Joe creates a short message ("I am Joe") as a signature (S) and encrypts his signature with his private key d_J. Joe then attaches the encrypted signature to the will and creates a message M where:

$$M = W + d_J(S)$$

To send M securely to Pat, Joe encrypts the whole message M by using Pat's public key. In other words, Joe encrypts M by using Pat's public key and generates $e_P(M)$. Pat receives this cipher and decrypts it by using Pat's private key e_P, i.e., it obtains M by using D(E(M)) = M. But the message M = W + D_j (S), i.e., it has the will plus the signature. The will is in plaintext now but the signature is still encrypted. The ciphertext $d_j(S)$ is decrypted by Pat by using Joe's public key e_j. This means that the signature is Joe's signature because the ciphertext can only be decrypted if sent from the right person.

In real life, digital signature protocols are more complicated than this to make sure that the signatures cannot be forged, modified, or attached to different documents. Consider, for example, if a digitally signed document was sent but a malicious individual detached the signature from the document and attached it to another document or modified the document in some way. In other words, it is extremely important to make sure that the message is not modified. This is accomplished through message digests presented in the next section. Another way to assure the identity of the remote user in prolonged conversations is to issue a "challenge" (a random number) that is sent to the remote user. The remote user encrypts the challenge with her private key and sends it back to you. Now, if you can decrypt the challenge by using the remote users' public key, then you can continue. If not, you should disconnect (and perhaps, send some impolite message to the intruder!).

12.14.3 Message Digest for Maintaining Integrity of Information

Both symmetric and asymmetric key cryptography provide some built-in integrity checking to make sure that the information is not modified in transit. Modified messages do not decrypt correctly. However, this is not a strong integrity check because messages are typically encrypted in small blocks of text, It is possible for a portion of an encrypted message to be deleted or duplicated without any problems. In addition, it is possible to take someone's signature from one document and attach it to another document without causing a decryption problem. In many secure communications, it is extremely important that the slightest change in a document cause the receiving process to detect the change and cause an error. In many cases, integrity of information is more important than privacy. Encryption schemes handle privacy but do not guarantee integrity. This is extremely important in wireless systems because wireless communications are very prone to errors due to rain, thunderstorms, and God's other creation known as hackers.

Message digesting is used to make sure that a certain message was not changed along the way between the sender and the receiver. A message digest algorithm f(), as shown in Figure 12-15, produces a fingerprint of the message, by applying a hashing function to it. The receiver can check for the integrity of the message by reapplying the hash function and comparing it with the original fingerprint. The hash functions used in these schemes are such that the fingerprint changes dramatically if a single bit of the message changes. Two messages cannot generate the same hash.

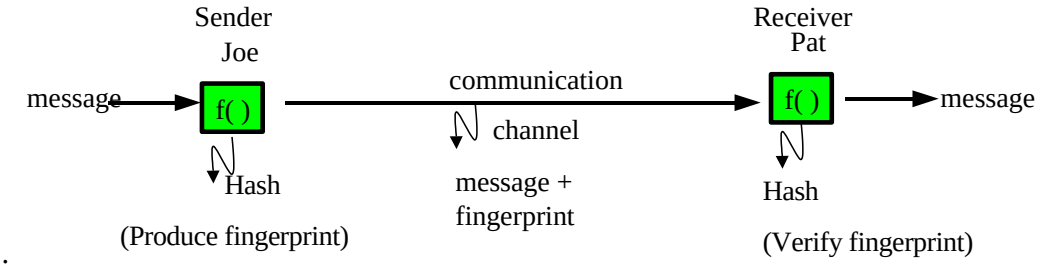


Figure 12-15: Message Digest

The message digest algorithms produce the fingerprint of a message using hashing schemes to verify its integrity in a manner that is similar to encryption. However, no decryption is required. This is why it is also called one way encryption. The process works as follows:

- Sender creates a hash h1 and sends it with message
- Receiver gets the message and creates a hash h2
- If h1 = h2, then everything is OK, else loss of data

A one-way hash function takes variable-length input, a message of even thousands or millions of bits, and produces a fixed-length output; say, 128-bits. The hash function ensures that, if the information is changed in any way, an entirely different output value is produced.

Several systems such as PGP (Pretty Good Privacy) use a cryptographically strong hash function on the plaintext. This generates a fixed-length data item known as a *message digest.* Once again, any change to the information results in a totally different digest. Let us take the example of PGP a little further. After creating the digest, PGP uses the digest and the private key to create the "signature." PGP transmits the signature and the plaintext together. Upon receipt of the message, the recipient uses

PGP to recompute the digest, thus verifying the signature. By using a secure hash function, you cannot take someone's signature from one document and attach it to another, or alter a signed message in any way. The slightest change in a signed document will cause the digital signature verification process to fail.

Many hashing algorithms are in use (no surprise!). Most popular algorithms are a) MD5, the most widely used message digest function -- it produces a 128-bit hash, and b) SHA, a secure hash algorithm developed by NIST for a digital signature standard - it uses a 160-bit hash.

12.14.4 Digital Envelopes – Combining Symmetric and Asymmetric Key Systems

Public key systems are quite secure but are quite slow due to their complexity. For example, the RSA algorithm is very complex because it performs intricate operations on large prime numbers. Performance of RSA is very slow (can be up to 1000 times slower than private key algorithms [Stein 1998]). The performance degrades with message size, thus RSA is not suitable for large documents.

A good compromise is the digital envelope that uses a mixture of symmetric and asymmetric encryption. Digital envelopes use public key only to encrypt the keys of a symmetric system instead of the entire message. The operation is similar to digital signature:

- Generate a secret key (called a "session key" because it is discarded after the communication session is done).
- Encrypt the message by using the session key and *symmetric algorithm*(e.g., DES)
- Encrypt the session key with the receiver's public key. This becomes the **digital envelope**.
- Send the digital envelope plus the encrypted message to the receiver.

Digital envelopes combine symmetric and asymmetric key systems and are the most commonly used encryption technique at present. They are used widely in many commercially available security systems such as PGP, SSL, SET, and others.

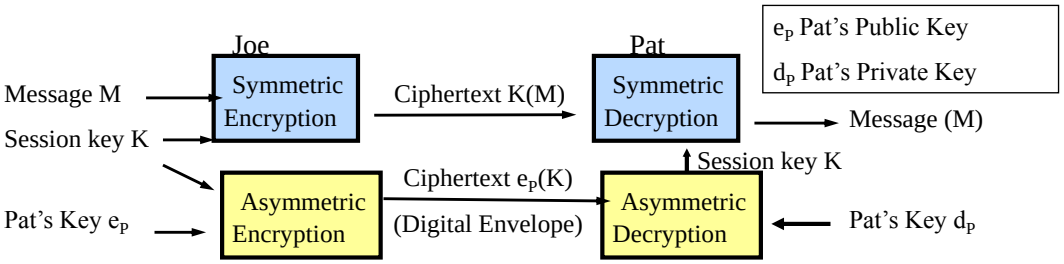


Figure 12-16: Digital Envelope

Figure 12-16 illustrates this process using our friends Joe and Pat. In essence, two encryption processes take place on the sender side as well as on the receiver side. On the sender (Joe) side, the session key K is used to encrypt the message M by using symmetric encryption. The key K is also encrypted by using Pat's public key and a digital envelope is created that contains the session key. On the receiver (Pat) side, first the digital envelope is opened using Pat's private key. This produces the session key K that is then used to decrypt the message M.

Diffie-Hollman: Encryption Without Authentication

Public key cryptography usually requires authentication. However, in some cases, it may be desirable for two parties to communicate with each other securely but with anonymity. An algorithm known as Diffie-Hollman allows two parties to negotiate a session key without ever sending the key itself across the network. This algorithm works by having the two parties pick a partial key independently. Then they exchange enough information with each other so that each can independently build a session key but not enough so that an eavesdropper can reconstruct the session key. This key is now used in a symmetric key fashion to encrypt and exchange documents. When the exchange is done, the key is discarded.

Diffie-Hollman suffers from a problem known as man in the middle. In this case, a malicious individual ("man in the middle") C could intercept the traffic between the two parties (A and B) and masquerade as B to A and vice versa. See the following section for a discussion of this problem. This type of attack is more realistic in wireless networks.

12.14.5 Man in the Middle – Security Weakness of Public Key Systems

Public key systems, in general, provide very strong protection for information storage and transmission. However, there is a well known problem with public key systems called "man in the middle" that should be mentioned here. This problem is associated with a special class of security scenario that involves encryption without authentication known as Diffie-Hollman (see the sidebar " Diffie-Hollman: Encryption Without Authentication"). However, it is serious enough, especially in wireless networks, that it should be noted generally. The main problem is that a malicious individual ("man in the middle") could intercept the traffic between a sender and receiver and then send his own public key instead of the public key of the receiver and later decrypt the information by using his own private key. Let us go through some details using Figure 12-17.

Let us assume that B is the sender and A is the receiver, but C is a malicious man in the middle. Since we are using public key cryptography, each person (A, B, C) has two keys (e_A , d_A), (e_B , d_B), (e_C , d_C) where according to our convention, E is the public and D is the private key. The following scenario is plausible:

B wants to send message M to A, so B needs A's public key e_A

C is dishonest (man in the middle) and intercepts this message

C sends its own public key e_c to B

B encrypts the message and sends it out

C Intercepts the encrypted message and decrypts it by using his own private key d_C

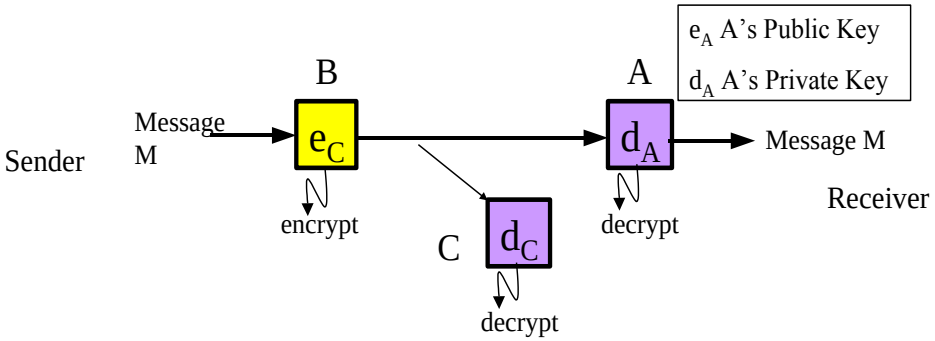


Figure 12-17: Man in the Middle - C is the Malicious person

To protect against this type of problem, the public key must also be protected and mechanisms must exist to verify that if you ask for A's public key, you in fact are getting A's public key. This is done through Certificate Authorities (CAs) that maintain a user's public key as a certificate. CAs are part of Public Key Infrastructure (PKI) that is discussed in the next section.

Key Point: How Symmetric and Asymmetric Systems are Really Being Used in Practice

Most current systems use a mixture of symmetric and asymmetric key cryptography. The reason is simple: Asymmetric key systems are quite secure but are quite slow due to their complexity but the symmetric key systems are the other way around (weak in security but strong in performance). Ideally, you would like to use the RSA algorithm or other asymmetric key algorithm on the entire plaintext but they are very very slow (can be up to 1000 times slower than symmetric). In addition, the performance of RSA degrades with message size, thus RSA is not suitable for large documents.

A good compromise is the **digital envelope** approach that uses a mixture of symmetric and asymmetric encryption. Digital envelopes use public key only to encrypt the *keys* of a symmetric system instead of the entire message. The operation works like this

- Generate a secret key by using a random number generator (called a "session key" because it is discarded after the communication session is done)
- Encrypt the message by using the session key and *symmetric algorithm*(e.g., DES)
- Encrypt the session key with the receiver's public key. This creates a digital envelope.
- Send the digital envelope plus the encrypted message to the receiver

This approach is used widely in systems such as PGP, SSL, SET, and others.

12.15 APPENDIX B: Certifying Authorities and the Public Key Infrastructure (PKI)

12.15.1 Overview

As stated previously, a major problem with the public key cryptosystem is that it works well as long as you know the public key of the recipient. Basically, you must be vigilant to ensure that you are encrypting to the correct recipient's key. To avoid *man-in-the-middle* attacks as discussed above, trusted third parties, called ***"certifying authorities"*** are used. Certifying authorities, along with several other support mechanisms to ensure a strong trusted environment, are known as the ***Public Key Infrastructure (PKI).*** Simply stated, *PKI is not one technology but a family of technologies that are based on the public key cryptography and contains the facilities to store and manage certificates (i.e., the ability to issue, revoke, store, retrieve, and trust certificates).* Specifically, PKI capabilities help create and manage asymmetric cryptographic keys or public/private key pairs required by applications. The following major components are essential for a system to qualify as a PKI:

Encryption based on the asymmetric as well as symmetric key cryptography.

Authentication Mechanisms that may include a wide range of options such as user ID and password, one-time passtokens, digital certificates, and biometrics.

Certification Authority (CA) is a commercial enterprise (e.g., Verisign) that vouches for the identities of individuals and organizations. A typical CA creates and signs digital certificates, maintains a list of certificates that have been revoked before the expiration date (certificate revocation lists), makes these certificates and revocation lists available, and provides an interface so administrators can manage certificates.

Registration Authority (RA) evaluates the credentials and relevant evidence that a person requesting a certificate is who they claim to be. The RA approves the request for issuance of a certificate by a CA. CA and RA functions are provided by a wide range of PKI providers such as Tivoli SecureWay Public Key Infrastructure

Directory Services define and implement a common schema for users and groups. The directory service is the point of integration for user authentication in many security systems. A user can be defined once within an enterprise, and information about that user can be accessed in a consistent manner by multiple different applications. This reduces administrative costs and complexity. PKI directory services are usually based on the Lightweight Directory Access Protocol (LDAP).

12.15.2 Certifying Authorities (CAs)

Basically, the main purpose of a CA is to bind a public key to a user. The CA is the authority whom everyone trusts, and no certificate is considered valid unless it has been signed by a trusted CA. To win such trust, a CA is responsible for ensuring that prior to issuing a certificate, he or she carefully checks it to be sure that the public key portion really belongs to the purported owner. Anyone who trusts the CA will automatically consider any certificates signed by the CA to be valid. For purpose of illustration, a CA's role is analogous to a Passport Office that issues passports (passport is a certificate). A CA creates certificates and digitally signs them using the CA's private key very much like a passport office creates a passport and then signs it by using the secret stamps and codes that are difficult to forge.

To issue a certificate, the CA goes through the following steps (see Figure 12-18):

- The user first generates a public/private key pair.
- Keep the private key and send the public key to the CA with the user identification information (name, SSN, birthdate, etc.) in the form of a "certificate request".
- The CA will verify the user identity through a procedure that may involve a telephone conversation or other mechanisms.
- If everything is verified, then the CA will issue a certificate with user name, email address, etc.
- The CA creates a signed certificate by creating a message digest and encrypting it with his private key. This makes sure that no one can modify this certificate.
- The certificate is returned to the user, with a copy kept by the CA.

Let us now assume that Joe wants to send a file to Pat. Before starting, Joe will ask Pat or the CA for a certificate. After receiving the certificate, Joe applies CA's public key to the certificate to verify correctness. If this works, then Joe can extract Pat's public key and send the file to Pat after encrypting it with Pat's public key just obtained from the certificate.

CAs and signed certificates are core components of PKI systems. As we will see later, different types of certificates are maintained by current PKI systems. Basically a CA creates and manages certificates very much like the authorities manage birth certificates.

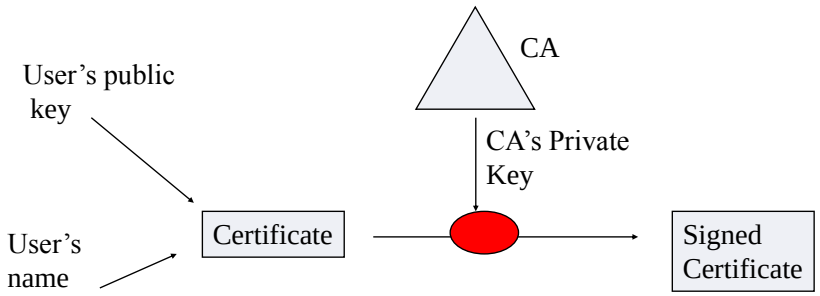


Figure 12-18: Certificate Authority (CA) and Creation of Certificates

12.15.3 Digital Certificates

A digital certificate is data that functions much like a physical certificate such as a passport or driver's license. A digital certificate includes a person's public key along with other information that verifies that a key is genuine or *valid. Just as passports and driver's licenses identify a person, a d*igital certificate is used to identify a person with his/her public key. Digital certificates simplify the task of establishing whether a public key really belongs to the purported owner and consists of:

- A public key.

- Certificate information. ("Identity" information about the user, such as name, user ID, and so on.)

- One or more digital signatures from CAs.

Once created, the certificates can be stored in a secure *certificate server*, also called a *key server*. A certificate server usually provides some administrative features that enable a company to maintain its security policies — for example, allowing only those keys that meet certain requirements to be stored.

A digital certificate can exist in a number of different formats. X.509 is the most common format. All X.509 certificates comply with the ITU-T X.509 international standard; thus (theoretically) X.509 certificates created for one application can be used by any application complying with X.509. In practice, however, different companies have created their own extensions to X.509 certificates, not all of them work together. Although X.509 is used widely in many systems, perhaps the most widely visible use of X.509 certificates is in Web browsers (see the sidebar "X509 Format"). A certificate is basically a text file (in reality, it is a record in a certificate server).

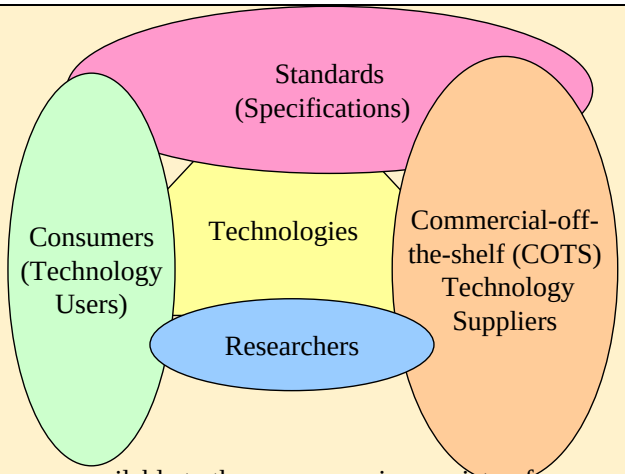
Certificates are created with a scheduled *validity period:* a start date/time and an expiration date/ time as shown in the sidebar "X.509 Certificate Format". When the certificate expires, it will no longer be valid, as the authenticity of its key/ identification pair are no longer assured. There are also situations where it is necessary to invalidate (revoke) a certificate prior to its expiration date, such as when the certificate holder terminates employment with the company or suspects that the certificate's corresponding private key has been compromised. It is much more important to detect a revoked certificate than an expired one. Expired certificates are unusable, but do not carry the same threat of compromise as a revoked certificate.

To summarize, a digital certificate binds an entity's identification to its public key and is issued by the Certification Authority. Digital certificates, typically based on the X.509 standard, enable Internet applications and other users to verify the identity of an entity. Unfortunately, certificates produced by one vendor product may not inter-operate with other vendor's because X.509 does not define the formats of the certificate entries and other necessary provisions. PKIX, the X.509 standard by IETF, defines the contents of public key certificates and is intended to resolve these interoperation issues.

Interrelationships between Technologies, Consumers, Researchers, and Suppliers

To understand the different players in security technologies, it may be beneficial to briefly discuss how the various technologies become available to the consumers. As shown in Figure 12-19, technologies become available to the consumers in a variety of ways. In many cases, COTS (commercial-off-the-shelf) technology providers build and sell technologies to consumers. We all use many products from Microsoft, IBM, and Oracle. An example in the security domain is SSL (Secure Socket Layer) that was developed by Netscape and is currently available in all Web browsers and servers. Ideally, however, research efforts result in development of technologies that are "hardened" by the COTS suppliers and made available to the consumers.

Examples are the encryption algorithms such as RSA and SHA that were developed by researchers and are currently available from many commercial vendors. In some cases, a few standards specifications are adopted and developed by COTS suppliers into products. Examples are IPSEC and S/MIME that have been specified by the IETF and are now commercially available. Some technologies not adopted by COTS suppliers are provided by individual groups. Kerberos and PGP are such examples (now they have been adopted by suppliers and are available as commercial-strength products).



Technologies become available to the consumers in a variety of ways
- Some standards specifications are adopted and developed by COTS suppliers
- Some research efforts also result in COTS technologies
- Some technologies not adopted by COTS providers are provided by individual groups

Figure 12-19: Interrelationships between Technologies, Consumers, Researchers, and Suppliers

12.15.4 Players in PKI – Standards and Technology Providers

Many applications use cryptographic software to incorporate public-key cryptography for encryption and authentication. A number of such technologies exist under the general umbrella of PKI that have their origin in the standards bodies, research communities, and COTS (commercial-off-the-shelf) technology providers (see the sidebar " Interrelationships between Technologies, Consumers, Researchers, and Suppliers"). As stated previously, PKI is a family of technologies that are based on the public key cryptography and contains the facilities to store and manage certificates (i.e., the ability to issue, revoke, store, retrieve, and trust certificates). Examples of the few members of the PKI family are reviewed below. :

12.15.4.1 PGP (Pretty Good Privacy)

PGP is a popular program, available on the Internet, that uses public-key cryptography to authenticate users to each other without the use of certificates. PGP, in essence, does not introduce any new technology, it merely packages several encryption technologies into a product. PGP is used heavily in group communications (among individuals). The security mechanisms are implemented in software that is free for individual use. The main characteristics of PGP are:

- It uses symmetric as well as asymmetric encryption.
- For symmetric encryption, PGP uses block cipher with 128 bit key to encrypt files or messages.

- A "session key" is generated automatically for files and messages based on a random number generator.
- The session key is encrypted by using the asymmetric encryption. PGP uses RSA to encrypt and exchange the session key. It uses the recipient's public key to encrypt.
- PGP can also be used to encrypt files for storage and encrypt messages for transmission.

To illustrate how PGP works, let us assume that Pat wants to send secure email to Joe. Pat starts email and encrypts it by using the PGP generated session key K. Pat then sends the email. To send the session key K, Pat gets Joe's public key (Ej) and encrypts K by using Ej. The encrypted K is now sent to Joe.

When Joe receives the email, he does the following. First he decrypts the session key K by using his private key (Dj). He then uses the decrypted key K to decrypt email.

A great deal of information about PGP, including tutorials and downloads, can be found at the Web site: www.pgpi.com

12.15.4.2 Kerberos (http://ww.mit.edu/kerberos/)

Kerberos is a cryptographic authentication scheme using a third-party authentication server to grant cryptographic "tokens" that authenticate users to a given service. Kerberos is an open standard designed to provide strong authentication by using secret-key cryptography. Used primarily for secure interoperation of existing systems, Kerberos is used for user authentication. Kerberos security system possesses the flexibility that allows using the following security technologies and mechanisms: the venerable user-ID and password; certificate-based public key systems; asymmetric-key cryptography; smart cards; token cards.

Authentication: Kerberos provides strong authentication by keeping a high level of assurance that the principal's claimed identity is genuine. Also, Kerberos provides mutual authentication, i.e. the identity of both client and service can be assured. To complete the authentication before the beginning of conversation, although very important, is not enough to assure the client that later the conversation would not be subverted. Kerberos provides the cryptographic "session keys" needed to establish a secure channel that keeps the conversation protected after the completion of the initial authentication procedure.

Authorization. Common mechanism to represent the authorization includes access control lists (ACLs) and capabilities. Although Kerberos doesn't provide an ACL-based authorization system, it provides all of the underlying services such a system requires. Also, Kerberos provides the facilities necessary for capability-based authorization and delegation processes.

Cryptography. Kerberos incorporates asymmetric-key cryptography, such as elliptic curve cryptography. It provides all of the basic security services using shared secrets and symmetric-key cryptography. The ability to use symmetric-key cryptography guarantees undisturbed use of performance-sensitive applications, such as high-volume transaction-processing systems, where each transaction is individually authenticated.

12.15.4.3 Entrust (www.entrust.com),

Entrust.net, a subsidiary of Entrust Technologies offers a portfolio of service solutions to securely manage e-business transactions. Solutions include secure e-business transactions from e-commerce Web sites to interactive cell phones. Entrust also recently entered the secure transaction business for

wireless. Entrust.net manages personal, web and WAP (for wireless) certificates. In particular, the new WAP Server Certificates are digital certificates that enable WAP servers to establish Wireless Transport Layer Security (WTLS) sessions with mobile phones and micro-browsers that support the WAP standard.

12.15.4.4 Others

A number of public-key based cryptographic infrastructure tools are available such as the Microsoft and Netscape Certificate Servers, which allow for the inclusion of public-key certificates in various applications. Examples of other packages include:

Verisign (www.verisign.com) provides a wide range of security solutions for certificates, secure messaging, wireless systems, and payment systems. In addition, it offers several industry specific solutions for retail, enterprises, telecoms, healthcare, and government agencies.

Tivoli (www.tivoli.com) started as a systems management company (managing performance and faults) but was bought by IBM. It provides a set of security services under the Tivoli Access Manager that controls both wired and wireless access to applications and data, keeping unauthorized users out. It also integrates with e-business applications to deliver a secure personalized experience for authorized users. Recent versions of Tivoli integrate security for key CRM, ERP, and SCM e-business solutions with enhancements for security in the current J2EE and .NET environments.

Microsoft's Passport Service is a PKI system for the .NET environment.

X.509 Certificate Format

The X.509 standard defines the information that goes into the certificate. All X.509 certificates have the following data (many fields are self explanatory and are not explained):

- X.509 version number
- Certificate holder's public key
- Serial number of the certificate —a unique serial number to distinguish it from other certificates it issues.
- Certificate holder's unique identifier —This is a unique name across the Internet. The uniqueness is achieved by several subsections that indicate user's *Common Name, Organizational Unit, Organization,* and *Country*)
- Certificate's validity period —indicates when the certificate will expire.
- Unique name of the certificate issuer — the unique name of the CA that signed the certificate.
- Digital signature of the issuer
- Signature algorithm identifier — identifies the algorithm used by the CA to sign the certificate.