Security Engineering & Secure Communications

Jae Gomes

Dianna Popp

First Edition, 2012

ISBN 978-81-323-0980-2

Published by:
Academic Studio
4735/22 Prakashdeep Bldg,
Ansari Road, Darya Ganj,
Delhi - 110002
Email: info@wtbooks.com

Table of Contents

Chapter-1

Security Engineering

Security engineering is a specialized field of engineering that deals with the development of detailed engineering plans and designs for security features, controls and systems. It is similar to other systems engineering activities in that its primary motivation is to support the delivery of engineering solutions that satisfy pre-defined functional and user requirements, but with the added dimension of preventing misuse and malicious behavior. These constraints and restrictions are often asserted as a security policy.

In one form or another, Security Engineering has existed as an informal field of study for several centuries. For example, the fields of locksmithing and security printing have been around for many years.

Due to recent catastrophic events, most notably 9/11, Security Engineering has quickly become a rapidly growing field. In fact, in a recent report completed in 2006, it was estimated that the global security industry was valued at US$150 billion.

Security engineering involves aspects of social science, psychology (such as designing a system to 'fail well' instead of trying to eliminate all sources of error) and economics, as well as physics, chemistry, mathematics, architecture and landscaping. Some of the techniques used, such as fault tree analysis, are derived from safety engineering.

Other techniques such as cryptography were previously restricted to military applications. One of the pioneers of security engineering as a formal field of study is Ross Anderson.

Qualifications

Typical qualifications for a security engineer are:

- Security+ - Entry Level
- Professional Engineer, Chartered Engineer, Chartered Professional Engineer
- Certified Protection Professional (CPP) - International certification by ASIS International
- Physical Security Professional (PSP) - International certification by ASIS International
- Certified Information Systems Security Professional (CISSP)

However, multiple qualifications, or several qualified persons working together, may provide a more complete solution.

Security stance

The two possible default positions on security matters are:

1. **Default deny** - "Everything, not explicitly permitted, is forbidden"

   Improves security at a cost in functionality.
   This is a good approach if you have lots of security threats.

2. **Default permit** - "Everything, not explicitly forbidden, is permitted"

   Allows greater functionality by sacrificing security.
   This is only a good approach in an environment where security threats are non-existent or negligible.

Core practices

- Security Requirements Analysis
- Security architecture
- Secure coding
- Security testing
- Security Operations and Maintenance
- Economics of security

Sub-fields

- Physical security

  - deter attackers from accessing a facility, resource, or information stored on physical media.

- Information security

  - protecting data from unauthorized access, use, disclosure, destruction, modification, or disruption to access.

- Economics of security

  - the economic aspects of economics of privacy and computer security.

Methodologies

Technological advances, principally in the field of computers, have now allowed the creation of far more complex systems, with new and complex security problems. Because modern systems cut across many areas of human endeavor, security engineers not only need consider the mathematical and physical properties of systems; they also need to consider attacks on the people who use and form parts of those systems using social engineering attacks. Secure systems have to resist not only technical attacks, but also coercion, fraud, and deception by confidence tricksters.

Web applications

According to the *Microsoft Developer Network* the patterns & practices of Security Engineering consists of the following activities:

- Security Objectives
- Security Design Guidelines
- Security Modeling
- Security Architecture and Design Review
- Security Code Review
- Security Testing
- Security Tuning
- Security Deployment Review

These activities are designed to help meet security objectives in the software life cycle.

Physical

Canadian Embassy in Washington, D.C. showing planters being used as vehicle barriers, and barriers and gates along the vehicle entrance

- Understanding of a *typical* threat and the usual risks to people and property.
- Understanding the incentives created both by the threat and the countermeasures.
- Understanding risk and threat analysis methodology and the benefits of an empirical study of the physical security of a facility.
- Understanding how to apply the methodology to buildings, critical infrastructure, ports, public transport and other facilities/compounds.
- Overview of common physical and technological methods of protection and understanding their roles in deterrence, detection and mitigation.
- Determining and prioritizing security needs and aligning them with the perceived threats and the available budget.

Target hardening

Whatever the target, there are multiple ways of preventing penetration by unwanted or unauthorised persons. Methods include placing Jersey barriers, stairs or other sturdy obstacles outside tall or politically sensitive buildings to prevent car and truck bombings. Improving the method of visitor management and some new electronic locks take advantage of technologies such as fingerprint scanning, iris or retinal scanning, and voiceprint identification to authenticate users.

Employers of security engineers

- US Department of State, Bureau of Diplomatic Security (ABET certified institution degree in engineering or physics required)

Criticisms

Use of the term engineer

Some criticize this field as not being a bona fide field of engineering because the methodologies of this field are less formal or excessively ad-hoc compared to other fields and many in the practice of security engineering have no engineering degree.

Security engineering as a systems engineering discipline

Security engineering is not considered to be a true form of systems engineering by some. Part of the problem lies in the fact that while conforming to positive requirements is well understood; conforming to negative requirements requires complex and indirect posturing to reach a closed form solution. In fact, some rigorous methods do exist to address these difficulties but are seldom used, partly because they are viewed as too old or too complex by many practitioners. As a result, many ad-hoc approaches simply do not succeed.

Chapter-2

Computer Security

Computer security is a branch of computer technology known as information security as applied to computers and networks. The objective of computer security includes protection of information and property from theft, corruption, or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The term computer system security means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The strategies and methodologies of computer security often differ from most other computer technologies because of its somewhat elusive objective of preventing unwanted computer behavior instead of enabling wanted computer behavior.

Security by design

The technologies of computer security are based on logic. As security is not necessarily the primary goal of most computer applications, designing a program with security in mind often imposes restrictions on that program's behavior.

There are 4 approaches to security in computing, sometimes a combination of approaches is valid:

1. Trust all the software to abide by a security policy but the software is not trustworthy (this is computer insecurity).
2. Trust all the software to abide by a security policy and the software is validated as trustworthy (by tedious branch and path analysis for example).
3. Trust no software but enforce a security policy with mechanisms that are not trustworthy (again this is computer insecurity).
4. Trust no software but enforce a security policy with trustworthy hardware mechanisms.

Many systems have unintentionally resulted in the first possibility. Since approach two is expensive and non-deterministic, its use is very limited. Approaches one and three lead to failure. Because approach number four is often based on hardware mechanisms and avoids abstractions and a multiplicity of degrees of freedom, it is more practical.

Combinations of approaches two and four are often used in a layered architecture with thin layers of two and thick layers of four.

There are various strategies and techniques used to design security systems. However there are few, if any, effective strategies to enhance security after design. One technique enforces the principle of least privilege to great extent, where an entity has only the privileges that are needed for its function. That way even if an attacker gains access to one part of the system, fine-grained security ensures that it is just as difficult for them to access the rest.

Furthermore, by breaking the system up into smaller components, the complexity of individual components is reduced, opening up the possibility of using techniques such as automated theorem proving to prove the correctness of crucial software subsystems. This enables a closed form solution to security that works well when only a single well-characterized property can be isolated as critical, and that property is also assessible to math. Not surprisingly, it is impractical for generalized correctness, which probably cannot even be defined, much less proven. Where formal correctness proofs are not possible, rigorous use of code review and unit testing represent a best-effort approach to make modules secure.

The design should use "defense in depth", where more than one subsystem needs to be violated to compromise the integrity of the system and the information it holds. Defense in depth works when the breaching of one security measure does not provide a platform to facilitate subverting another. Also, the cascading principle acknowledges that several low hurdles does not make a high hurdle. So cascading several weak mechanisms does not provide the safety of a single stronger mechanism.

Subsystems should default to secure settings, and wherever possible should be designed to "fail secure" rather than "fail insecure". Ideally, a secure system should require a deliberate, conscious, knowledgeable and free decision on the part of legitimate authorities in order to make it insecure.

In addition, security should not be an all or nothing issue. The designers and operators of systems should assume that security breaches are inevitable. Full audit trails should be kept of system activity, so that when a security breach occurs, the mechanism and extent of the breach can be determined. Storing audit trails remotely, where they can only be appended to, can keep intruders from covering their tracks. Finally, full disclosure helps to ensure that when bugs are found the "window of vulnerability" is kept as short as possible.

Security architecture

Security Architecture can be defined as the design artifacts that describe how the security controls (security countermeasures) are positioned, and how they relate to the overall information technology architecture. These controls serve the purpose to maintain the system's quality attributes, among them confidentiality, integrity, availability, accountability and assurance."

Hardware mechanisms that protect computers and data

Hardware based or assisted computer security offers an alternative to software-only computer security. Devices such as dongles may be considered more secure due to the physical access required in order to be compromised.

Secure operating systems

One use of the term computer security refers to technology to implement a secure operating system. Much of this technology is based on science developed in the 1980s and used to produce what may be some of the most impenetrable operating systems ever. Though still valid, the technology is in limited use today, primarily because it imposes some changes to system management and also because it is not widely understood. Such ultra-strong secure operating systems are based on operating system kernel technology that can guarantee that certain security policies are absolutely enforced in an operating environment. An example of such a Computer security policy is the Bell-LaPadula model. The strategy is based on a coupling of special microprocessor hardware features, often involving the memory management unit, to a special correctly implemented operating system kernel. This forms the foundation for a secure operating system which, if certain critical parts are designed and implemented correctly, can ensure the absolute impossibility of penetration by hostile elements. This capability is enabled because the configuration not only imposes a security policy, but in theory completely protects itself from corruption. Ordinary operating systems, on the other hand, lack the features that assure this maximal level of security. The design methodology to produce such secure systems is precise, deterministic and logical.

Systems designed with such methodology represent the state of the art of computer security although products using such security are not widely known. In sharp contrast to most kinds of software, they meet specifications with verifiable certainty comparable to specifications for size, weight and power. Secure operating systems designed this way are used primarily to protect national security information, military secrets, and the data of international financial institutions. These are very powerful security tools and very few secure operating systems have been certified at the highest level (Orange Book A-1) to operate over the range of "Top Secret" to "unclassified" (including Honeywell SCOMP, USAF SACDIN, NSA Blacker and Boeing MLS LAN.) The assurance of security depends not only on the soundness of the design strategy, but also on the assurance of correctness of the implementation, and therefore there are degrees of security strength defined for COMPUSEC. The Common Criteria quantifies security strength of products in terms of two components, security functionality and assurance level (such as EAL levels), and these are specified in a Protection Profile for requirements and a Security Target for product descriptions. None of these ultra-high assurance secure general purpose operating systems have been produced for decades or certified under the Common Criteria.

In USA parlance, the term High Assurance usually suggests the system has the right security functions that are implemented robustly enough to protect DoD and DoE classified information. Medium assurance suggests it can protect less valuable information, such as income tax information. Secure operating systems designed to meet medium robustness levels of security functionality and assurance have seen wider use within both government and commercial markets. Medium robust systems may provide the same security functions as high assurance secure operating systems but do so at a lower assurance level (such as Common Criteria levels EAL4 or EAL5). Lower levels mean we can be less certain that the security functions are implemented flawlessly, and therefore less dependable. These systems are found in use on web servers, guards, database servers, and management hosts and are used not only to protect the data stored on these systems but also to provide a high level of protection for network connections and routing services.

Secure coding

If the operating environment is not based on a secure operating system capable of maintaining a domain for its own execution, and capable of protecting application code from malicious subversion, and capable of protecting the system from subverted code, then high degrees of security are understandably not possible. While such secure operating systems are possible and have been implemented, most commercial systems fall in a 'low security' category because they rely on features not supported by secure operating systems (like portability, et al.). In low security operating environments, applications must be relied on to participate in their own protection. There are 'best effort' secure coding practices that can be followed to make an application more resistant to malicious subversion.

In commercial environments, the majority of software subversion vulnerabilities result from a few known kinds of coding defects. Common software defects include buffer overflows, format string vulnerabilities, integer overflow, and code/command injection. It is to be immediately noted that all of the foregoing are specific instances of a general class of attacks, where situations in which putative "data" actually contains implicit or explicit, executable instructions are cleverly exploited.

Some common languages such as C and C++ are vulnerable to all of these defects. Other languages, such as Java, are more resistant to some of these defects, but are still prone to code/command injection and other software defects which facilitate subversion.

Recently another bad coding practice has come under scrutiny; dangling pointers. The first known exploit for this particular problem was presented in July 2007. Before this publication the problem was known but considered to be academic and not practically exploitable.

Unfortunately, there is no theoretical model of "secure coding" practices, nor is one practically achievable, insofar as the variety of mechanisms are too wide and the manners in which they can be exploited are too variegated. It is interesting to note, however, that such vulnerabilities often arise from archaic philosophies in which computers were assumed to be narrowly disseminated entities used by a chosen few, all of whom were likely highly educated, solidly trained academics with naught but the goodness of mankind in mind. Thus, it was considered quite harmless if, for (fictitious) example, a FORMAT string in a FORTRAN program could contain the J format specifier to mean "shut down system after printing." After all, who would use such a feature but a well-intentioned system programmer? It was simply beyond conception that software could be deployed in a destructive fashion.

It is worth noting that, in some languages, the distinction between code (ideally, read-only) and data (generally read/write) is blurred. In LISP, particularly, there is no distinction whatsoever between code and data, both taking the same form: an S-expression can be code, or data, or both, and the "user" of a LISP program who manages to insert an executable LAMBDA segment into putative "data" can achieve arbitrarily general and dangerous functionality. Even something as "modern" as Perl offers the eval() function, which enables one to generate Perl code and submit it to the interpreter, disguised as string data.

Capabilities and access control lists

Within computer systems, two security models capable of enforcing privilege separation are access control lists (ACLs) and capability-based security. The semantics of ACLs have been proven to be insecure in many situations, e.g., the confused deputy problem. It has also been shown that the promise of ACLs of giving access to an object to only one person can never be guaranteed in practice. Both of these problems are resolved by capabilities. This does not mean practical flaws exist in all ACL-based systems, but only that the designers of certain utilities must take responsibility to ensure that they do not introduce flaws.

Capabilities have been mostly restricted to research operating systems and commercial OSs still use ACLs. Capabilities can, however, also be implemented at the language level, leading to a style of programming that is essentially a refinement of standard object-oriented design. An open source project in the area is the E language.

First the Plessey System 250 and then Cambridge CAP computer demonstrated the use of capabilities, both in hardware and software, in the 1970s. A reason for the lack of adoption of capabilities may be that ACLs appeared to offer a 'quick fix' for security without pervasive redesign of the operating system and hardware.

The most secure computers are those not connected to the Internet and shielded from any interference. In the real world, the most security comes from operating systems where security is not an add-on.

Applications

Computer security is critical in almost any technology-driven industry which operates on computer systems. Computer security can also be referred to as computer safety. The issues of computer based systems and addressing their countless vulnerabilities are an integral part of maintaining an operational industry.

Cloud computing Security

Security in the cloud is challenging, due to varied degree of security features and management schemes within the cloud entitites. In this connection one logical protocol base need to evolve so that the entire gamet of components operate synchronously and securely.

In aviation

The aviation industry is especially important when analyzing computer security because the involved risks include human life, expensive equipment, cargo, and transportation infrastructure. Security can be compromised by hardware and software malpractice, human error, and faulty operating environments. Threats that exploit computer vulnerabilities can stem from sabotage, espionage, industrial competition, terrorist attack, mechanical malfunction, and human error.

The consequences of a successful deliberate or inadvertent misuse of a computer system in the aviation industry range from loss of confidentiality to loss of system integrity, which may lead to more serious concerns such as data theft or loss, network and air traffic control outages, which in turn can lead to airport closures, loss of aircraft, loss of passenger life. Military systems that control munitions can pose an even greater risk.

A proper attack does not need to be very high tech or well funded; for a power outage at an airport alone can cause repercussions worldwide.. One of the easiest and, arguably, the most difficult to trace security vulnerabilities is achievable by transmitting unauthorized communications over specific radio frequencies. These transmissions may spoof air traffic controllers or simply disrupt communications altogether. These incidents are very common, having altered flight courses of commercial aircraft and caused panic and confusion in the past. Controlling aircraft over oceans is especially dangerous because radar surveillance only extends 175 to 225 miles offshore. Beyond the radar's sight controllers must rely on periodic radio communications with a third party.

Lightning, power fluctuations, surges, brown-outs, blown fuses, and various other power outages instantly disable all computer systems, since they are dependent on an electrical source. Other accidental and intentional faults have caused significant disruption of safety critical systems throughout the last few decades and dependence on reliable communication and electrical power only jeopardizes computer safety.

Notable system accidents

In 1994, over a hundred intrusions were made by unidentified crackers into the Rome Laboratory, the US Air Force's main command and research facility. Using trojan horse viruses, hackers were able to obtain unrestricted access to Rome's networking systems and remove traces of their activities. The intruders were able to obtain classified files, such as air tasking order systems data and furthermore able to penetrate connected networks of National Aeronautics and Space Administration's Goddard Space Flight Center, Wright-Patterson Air Force Base, some Defense contractors, and other private sector organizations, by posing as a trusted Rome center user.

Computer security policy

United States

Cybersecurity Act of 2010

On April 1, 2009, Senator Jay Rockefeller (D-WV) introduced the "Cybersecurity Act of 2009 - S. 773" (full text) in the Senate; the bill, co-written with Senators Evan Bayh (D-IN), Barbara Mikulski (D-MD), Bill Nelson (D-FL), and Olympia Snowe (R-ME), was referred to the Committee on Commerce, Science, and Transportation, which approved a revised version of the same bill (the "Cybersecurity Act of 2010") on March 24, 2010. The bill seeks to increase collaboration between the public and the private sector on cybersecurity issues, especially those private entities that own infrastructures that are critical to national security interests (the bill quotes John Brennan, the Assistant to the President for Homeland Security and Counterterrorism: "our nation's security and economic prosperity depend on the security, stability, and integrity of communications and information infrastructure that are largely privately-owned and globally-operated" and talks about the country's response to a "cyber-Katrina".), increase public awareness on cybersecurity issues, and foster and fund cybersecurity research. Some of the most controversial parts of the bill include Paragraph 315, which grants the President the right to "order the limitation or shutdown of Internet traffic to and from any compromised Federal Government or United States critical infrastructure information system or network." The Electronic Frontier Foundation, an international non-profit digital rights advocacy and legal organization based in the United States, characterized the bill as promoting a "potentially dangerous approach that favors the dramatic over the sober response".

International Cybercrime Reporting and Cooperation Act

On March 25, 2010, Representative Yvette Clarke (D-NY) introduced the "International Cybercrime Reporting and Cooperation Act - H.R.4962" (full text) in the House of Representatives; the bill, co-sponsored by seven other representatives (among whom only one Republican), was referred to three House committees. The bill seeks to make sure that the administration keeps Congress informed on information infrastructure, cybercrime, and end-user protection worldwide. It also "directs the President to give priority for assistance to improve legal, judicial, and enforcement capabilities with respect to cybercrime to countries with low information and communications technology levels of development or utilization in their critical infrastructure, telecommunications systems, and financial industries" as well as to develop an action plan and an annual compliance assessment for countries of "cyber concern".

Protecting Cyberspace as a National Asset Act of 2010 ("*Kill switch bill*")

On June 19, 2010, United States Senator Joe Lieberman (I-CT) introduced a bill called "Protecting Cyberspace as a National Asset Act of 2010 - S.3480" (full text in pdf), which he co-wrote with Senator Susan Collins (R-ME) and Senator Thomas Carper (D-DE). If signed into law, this controversial bill, which the American media dubbed the "*Kill switch bill*", would grant the President emergency powers over the Internet. However, all three co-authors of the bill issued a statement claiming that instead, the bill "[narrowed] existing broad Presidential authority to take over telecommunications networks".

Terminology

The following terms used in engineering secure systems are explained below.

- Authentication techniques can be used to ensure that communication end-points are who they say they are.
- Automated theorem proving and other verification tools can enable critical algorithms and code used in secure systems to be mathematically proven to meet their specifications.
- Capability and access control list techniques can be used to ensure privilege separation and mandatory access control. This section discusses their use.
- Chain of trust techniques can be used to attempt to ensure that all software loaded has been certified as authentic by the system's designers.
- Cryptographic techniques can be used to defend data in transit between systems, reducing the probability that data exchanged between systems can be intercepted or modified.
- Firewalls can provide some protection from online intrusion.

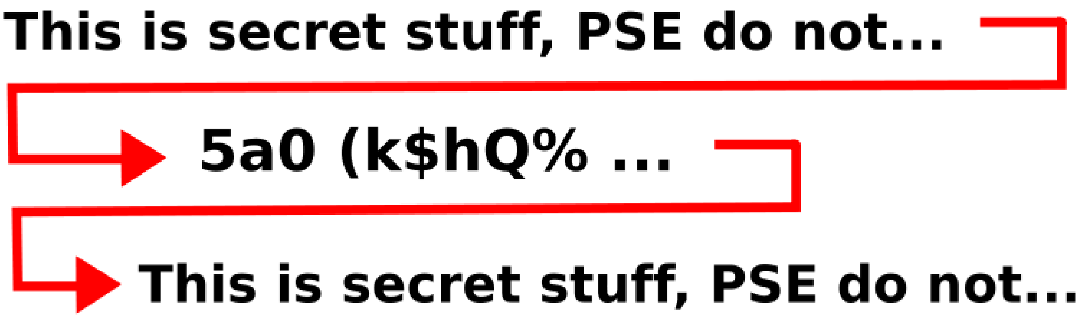
- A microkernel is a carefully crafted, deliberately small corpus of software that underlies the operating system *per se* and is used solely to provide very low-level, very precisely defined primitives upon which an operating system can be developed. A simple example with considerable didactic value is the early '90s GEMSOS (Gemini Computers), which provided extremely low-level primitives, such as "segment" management, atop which an operating system could be built. The theory (in the case of "segments") was that—rather than have the operating system itself worry about mandatory access separation by means of military-style labeling—it is safer if a low-level, independently scrutinized module can be charged **solely** with the management of individually labeled segments, be they memory "segments" or file system "segments" or executable text "segments." If software below the visibility of the operating system is (as in this case) charged with labeling, there is no theoretically viable means for a clever hacker to subvert the labeling scheme, since the operating system *per se* does **not** provide mechanisms for interfering with labeling: the operating system is, essentially, a client (an "application," arguably) atop the microkernel and, as such, subject to its restrictions.

- Endpoint Security software helps networks to prevent data theft and virus infection through portable storage devices, such as USB drives.

Some of the following items may belong to the computer insecurity article:

- Access authorization restricts access to a computer to group of users through the use of authentication systems. These systems can protect either the whole computer – such as through an interactive logon screen – or individual services, such as an FTP server. There are many methods for identifying and authenticating users, such as passwords, identification cards, and, more recently, smart cards and biometric systems.
- Anti-virus software consists of computer programs that attempt to identify, thwart and eliminate computer viruses and other malicious software (malware).
- Applications with known security flaws should not be run. Either leave it turned off until it can be patched or otherwise fixed, or delete it and replace it with some other application. Publicly known flaws are the main entry used by worms to automatically break into a system and then spread to other systems connected to it. The security website Secunia provides a search tool for unpatched known flaws in popular products.
- Backups are a way of securing information; they are another copy of all the important computer files kept in another location. These files are kept on hard disks, CD-Rs, CD-RWs, and tapes. Suggested locations for backups are a fireproof, waterproof, and heat proof safe, or in a separate, offsite location than that in which the original files are contained. Some individuals and companies also keep their backups in safe deposit boxes inside bank vaults. There is also a fourth option, which involves using one of the file hosting services that backs up files over the Internet for both business and individuals.
    - Backups are also important for reasons other than security. Natural disasters, such as earthquakes, hurricanes, or tornadoes, may strike the building where the computer is located. The building can be on fire, or an explosion may occur. There needs to be a recent backup at an alternate secure location, in case of such kind of disaster. Further, it is recommended that the alternate location be placed where the same disaster would not affect both locations. Examples of alternate disaster recovery sites being compromised by the same disaster that affected the primary site include having had a primary site in World Trade Center I and the recovery site in 7 World Trade Center, both of which were destroyed in the 9/11 attack, and having one's primary site and recovery site in the

same coastal region, which leads to both being vulnerable to hurricane damage (e.g. primary site in New Orleans and recovery site in Jefferson Parish, both of which were hit by Hurricane Katrina in 2005). The backup media should be moved between the geographic sites in a secure manner, in order to prevent them from being stolen.


This is secret stuff, PSE do not...
5a0 (k$hQ% ...
This is secret stuff, PSE do not...

Cryptographic techniques involve transforming information, scrambling it so it becomes unreadable during transmission. The intended recipient can unscramble the message, but eavesdroppers cannot.

- Encryption is used to protect the message from the eyes of others. Cryptographically secure ciphers are designed to make any practical attempt of breaking infeasible. Symmetric-key ciphers are suitable for bulk encryption using shared keys, and public-key encryption using digital certificates can provide a practical solution for the problem of securely communicating when no key is shared in advance.
- Firewalls are systems which help protect computers and computer networks from attack and subsequent intrusion by restricting the network traffic which can pass through them, based on a set of system administrator defined rules.
- Honey pots are computers that are either intentionally or unintentionally left vulnerable to attack by crackers. They can be used to catch crackers or fix vulnerabilities.
- Intrusion-detection systems can scan a network for people that are on the network but who should not be there or are doing things that they should not be doing, for example trying a lot of passwords to gain access to the network.
- Pinging The ping application can be used by potential crackers to find if an IP address is reachable. If a cracker finds a computer, they can try a port scan to detect and attack services on that computer.
- Social engineering awareness keeps employees aware of the dangers of social engineering and/or having a policy in place to prevent social engineering can reduce successful breaches of the network and servers.
- File Integrity Monitors are tools used to detect changes in the integrity of systems and files.

Chapter-3

Secure Communication

When two entities are communicating with each other, and they do not want a third party to listen to their communication, then they want to pass on their message in such a way that no body else could understand their message. This is known as communicating in a secure manner or **secure communication**. Secure communication includes means by which people can share information with varying degrees of certainty that third parties cannot know what was said. Other than communication spoken face to face out of possibility of listening, it is probably safe to say that no communication is guaranteed secure in this sense, although practical limitations such as legislation, resources, technical issues (interception and encryption), and the sheer volume of communication are limiting factors to surveillance.

The purpose here is to describe the various means by which security is sought and compromised, the differing kinds of security possible, and the current means and their degree of security readily available.

With many communications taking place over long distance and mediated by technology, and increasing awareness of the importance of interception issues, technology and its compromise are at the heart of this debate.

Users and needs

Many forms of everyday communication are "reasonably" secure, thus, we do not assume telephone calls are intercepted when we use them. However in some areas such as online intellectual property rights, legal, criminal, political and commercial communications, this assumption is inadequate.

History

In 1898, Nikola Tesla demonstrated a radio controlled boat in Madison Square Garden that allowed secure communication between transmitter and receiver.

One of the most famous forms of secure communication was the Green Hornet. During WWII, Winston Churchill had to make vital calls to the President of the United States, Franklin D. Roosevelt. These calls talked about such things as shipping and troop movements. At first, the calls were made using a radio phone as this was thought to be secure. Unfortunately, due to the Nazis having a listening station in Holland they were able to hear every last word. As soon as it was realized they stopped using the radio phone and started work on a whole new system, the Green Hornet. This meant that anyone listening in would just hear white noise but as the only two identical copies were held with the Prime Minister and the President the conversation was clear to them. As secrecy was paramount, the location of the Green Hornet was only known by the people who built it and Winston Churchill, and if anyone did see him entering the room it was kept in, all they would see was the Prime Minister entering a closet labeled 'Broom Cupboard.' It is the said that because of the way the Green Hornet works it is not able to be beaten, even today.

Nature and limitations of secure communication

Types of security

Security can be broadly categorised under the following headings, with examples:

- Hiding the content or nature of a communication
  - Code – A code is a rule for converting a piece of information (for example, a letter, word, phrase, or gesture) into another form or representation (one sign into another sign), not necessarily of the same type. In communications and information processing, encoding is the process by which information from a source is converted into symbols to be communicated. Decoding is the reverse process, converting these code symbols back into information understandable by a receiver. One reason for coding is to enable communication in places where ordinary spoken or written language is difficult or impossible. For example, semaphore, where the configuration of flags held by a signaller or the arms of a semaphore tower encodes parts of the message, typically individual letters and numbers. Another person standing a great distance away can interpret the flags and reproduce the words sent.

  - Encryption
  - Steganography
  - Identity Based
- Hiding the parties to a communication (prevention of identification, or anonymity)
  - "Crowds" and similar anonymous group structures. i.e. it is difficult to identify who said what when it comes from a "crowd".
  - Anonymous communication devices (unregistered cellphones, Internet cafes)
  - Anonymous proxies
  - Hard to trace routing methods (through unauthorized 3rd party systems, or relays)
- Hiding the fact that a communication takes place
  - "Security by obscurity" (similar to needle in a haystack)

- Random traffic (creating random data flow in order that the presence of genuine communication is harder to detect and traffic analysis less reliable)

Each of the three is important, and depending on the circumstances any of these may be critical. For example, if a communication is not readily identifiable, then it is unlikely to attract attention for identification of parties, and the mere fact a communication has taken place (regardless of content) is often enough by itself to establish an evidential link in legal prosecutions. It is also important with computers, to be sure where the security is applied, and what is covered.

Borderline cases

A further category, which touches upon secure communication, is software intended to take advantage of security openings at the end-points. This software category includes trojan horses, keyloggers and other spyware.

These types of activity are usually addressed with everyday mainstream security methods, such as antivirus software, firewalls, programs that identify or neutralize adware and spyware, as well as web filtering programs such as proxomitron and privoxy which check all web pages being read and identify and remove common nuisances contained. As a rule they fall under computer security rather than secure communications.

Tools used to obtain security

Encryption

Encryption is where data is rendered hard to read by an unauthorised party. Since encryption can be made extremely hard to break, many communication methods either use deliberately weaker encryption than possible, or have backdoors inserted to permit rapid decryption. In some cases government authorities have required backdoors be installed in secret. Many methods of encryption are also subject to "man in the middle" attack whereby a third party who can 'see' the establishment of the secure communication is made privy to the encryption method, this would apply for example to interception of computer use at an ISP. Provided it is correctly programmed, sufficiently powerful, and the keys not intercepted, encryption would usually be considered secure.

The encryption can be implemented in a way to require the use of encryption, i.e. if encrypted communication is impossible then no traffic is sent, or opportunistically. Opportunistic encryption is a lower security method to generally increase the percentage of generic traffic which is encrypted. This is analogous to beginning every conversation with "Do you speak Navajo?" If the response is affirmative, then the conversation proceeds in Navajo, otherwise it uses the common language of the two speakers. This method does not generally provide authentication or anonymity but it does protect the content of the conversation from eavesdropping.

Steganography

Steganography ("hidden writing") is the means by which data can be hidden within other more innocuous data. Thus a watermark proving ownership embedded in the data of a picture, in such a way it is hard to find or remove unless you know how to find it. or, for communication, the hiding of important data (such as a telephone number) in apparently innocuous data (an MP3 music file). An advantage of steganography is plausible deniability, that is, unless one can prove the data is there (which is usually not easy), it is deniable that the file contains any.

Identity based networks

Unwanted or malicious behavior is possible on the web since it is inherently anonymous. True identity based networks replace the ability to remain anonymous and are inherently more trustworthy since the identity of the sender and recipient are known. (The telephone system is an example of an identity based network.)

Anonymized networks

Recently, anonymous networking has been used to secure communications. In principle, a large number of users running the same system, can have communications routed between them in such a way that it is very hard to detect what any complete message is, which user sent it, and where it is ultimately going from or to. Examples are Crowds, Tor, I2P, Mixminion, various anonymous P2P networks, and others.

Anonymous communication devices

In theory, an unknown device would not be noticed, since so many other devices are in use. This is not altogether the case in reality, due to the presence of systems such as Carnivore and Echelon which can monitor communications over entire networks, and the fact that the far end may be monitored as before. Examples include payphones, Internet cafe, etc.

Methods used to "break" security

Bugging

The placing covertly of monitoring and/or transmission devices either within the communication device, or in the premises concerned.

Computers (general)

Any security obtained from a computer is limited by the many ways it can be compromised - by hacking, keystroke logging, backdoors, or even in extreme cases by monitoring the tiny electrical signals given off by keyboard or monitors to reconstruct what is typed or seen (TEMPEST, which is quite complex).

Laser reading of windows

In certain cases individuals have had private spoken communications intercepted by means of laser. This usually involves a sensitive laser directed at a window, capable of picking up the tiny glass movements caused by sounds, and conversion back to speech.

Systems offering a degree of secure communication

Anonymous cellphones

Cellphones can easily be obtained, but are also easily traced and "tapped". There is no (or only limited) encryption, the phones are traceable - often even when switched off - since the phone and SIM card broadcast their International Mobile Subscriber Identity (IMSI). It is possible for a cellphone company to turn on some cellphones when the user is unaware and use the microphone to listen in on you, and according to James Atkinson, a counter-surveillance specialist cited in the same source, "Security-conscious corporate executives routinely remove the batteries from their cell phones" since many phones' software can be used "as-is", or modified, to enable transmission without user awareness and the user can be located within a small distance using signal triangulation and now using built in GPS features for newer models.

Landlines

Analogue landlines are not encrypted and are trivially tapped. Such tapping requires physical access to the line which is easily obtained from a number of places, e.g. distribution points, cabinets and the exchange itself. Tapping a landline in this way would also enable the attacker to make calls which appear to originate from the tapped line.

Anonymous Internet

Using a third party system of any kind (payphone, Internet cafe) is often quite secure, however if that system is used to access known locations (a known email account or 3rd party) then it may be tapped at the far end, or noted, and this will remove any security benefit obtained. Some countries also impose mandatory registration of Internet cafe users.

Anomymous proxies are another common type of protection, which allow one to access the net via a third party (often in a different country) and make tracing difficult. Note that there is seldom any guarantee that the plaintext is not tappable, nor that the proxy does not keep its own records of users or entire dialogs. As a result anonymous proxies are a generally useful tool but may not be as secure as other systems whose security can be better assured. Their most common use is to prevent a record of the originating IP, or address, being left on the target site's own records.

A recent development on this theme arises when wireless Internet connections ("Wi-fi") are left in their unsecured state. The effect of this is that any person in range of the base unit can piggyback the connection - that is, use it without the owner being aware. Since many connections are left open in this manner, situations where piggybacking might arise (willful or unaware) have successfully led to a defense in some cases, since it makes it difficult to prove the owner of the connection was the downloader, or had knowledge of the use to which unknown others might be putting their connection. An example of this was the Tammie Marson case, where neighbours and anyone else might have been the culprit in the sharing of copyright files. Conversely, in other cases, people deliberately seek out businesses and households with unsecured connections, for illicit and anonymous Internet usage, or simply to obtain free bandwidth.

Programs offering more secure communications

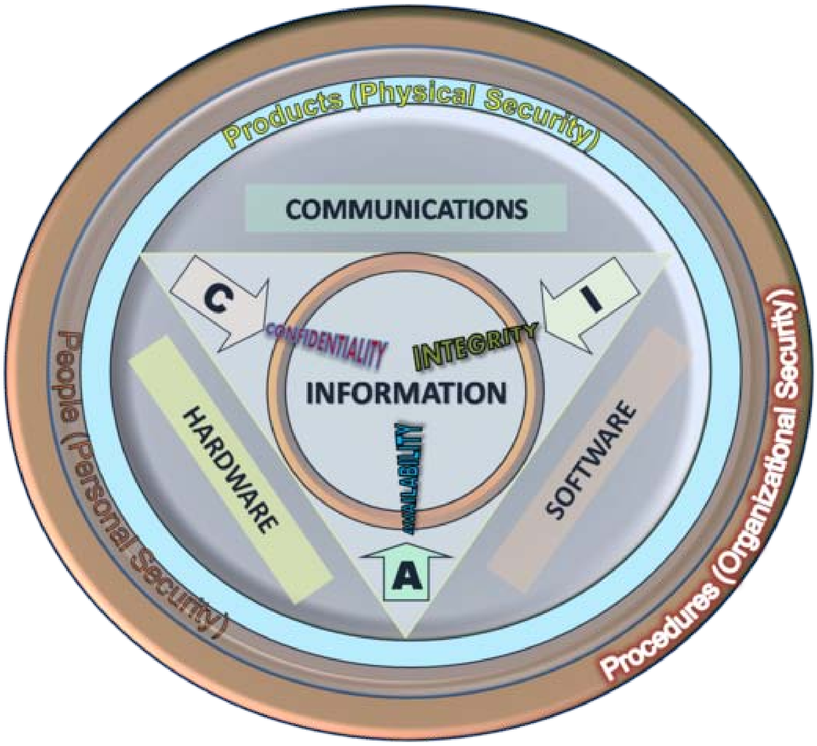
- Skype - secure voice over Internet, secure chat. Uses 128-bit AES (256-bit is the standard) and 1024-bit asymmetrical protocols to exchange initial keys (which is considered relatively weak by NIST). Proprietary. No information on backdoors. An article in 2004 suggested that Skype has relatively weak encryption, but more recent analyses, one by invitation and one by reverse engineering presented at DEF CON 2005, both conclude that Skype uses encryption effectively. Criticism focuses upon its proprietary "black box" design, its relatively short (1536 bit) keys, excessive bandwidth use of user supernodes, and excessive trust of other computers able to "speak Skype".
- Zfone is an open source secure voice over Internet program, by Phil Zimmermann, the creator of PGP.
- pbxnsip is a SIP-based PBX that uses TLS and SRTP for encrypting the voice traffic. In contrast to other proprietary protocols, the protocol is open so that devices from independent vendors can be used. The encryption includes the relay of Instant Messaging and Presence information as well as the management interface.
- Secure IRC and web chat - Some IRC clients and systems use security such as SSL. This is not standardised. Likewise some web chat clients such as Yahoo Messenger use secure communications on their web based program. Again the security of these is unverified, and it is likely the communication is not secured other than to and from the client.
- Trillian - offers secure IM facility, however appears to have weaknesses in key exchange which would enable a "man in the middle" attack with ease. Proprietary, no information on backdoors.
- Off-the-Record Messaging is a plugin which adds end-to-end encryption and authentication as well as Perfect forward secrecy to instant messaging. It is not a separate protocol but runs under most every IM protocol.
- WASTE - open source secure IM, high strength "end to end" encryption, within an anonymised network.
- Secure email - some email networks such as "hushmail" or Opolis Secure Mail, are designed to provide encrypted and/or anonymous communication. They authenticate and encrypt on the users own computer, to prevent transmission of plain text, and mask the sender and recipient. Mixminion provides a higher level

of anonymity by using a network of anonymizing intermediaries, (similar to how Tor and crowds work above).

- AESpad.com - open source online encrypted secure chat. Uses 256-bit AES symmetrical encryption. Relies on a pre-shared key between chat participants.

Chapter-4

Information Security


Products (Physical Security)
COMMUNICATIONS
C
I
CONFIDENTIALITY
INTEGRITY
INFORMATION
HARDWARE
AVAILABILITY
SOFTWARE
A
People (Personal Security)
Procedures (Organizational Security)

Information Security Components: or qualities, i.e., Confidentiality, Integrity and Availability (CIA). Information Systems are decomposed in three main portions, hardware, software and communications with the purpose to identify and apply information security industry standards, as mechanisms of protection and prevention, at three levels or layers: Physical, personal and organizational. Essentially, procedures or policies are implemented to tell people (administrators, users and operators) how to use products to ensure information security within the organizations.

Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction.

The terms information security, computer security and information assurance are frequently incorrectly used interchangeably. These fields are interrelated often and share the common goals of protecting the confidentiality, integrity and availability of information; however, there are some subtle differences between them.

These differences lie primarily in the approach to the subject, the methodologies used, and the areas of concentration. Information security is concerned with the confidentiality, integrity and availability of data regardless of the form the data may take: electronic, print, or other forms.

Computer security can focus on ensuring the availability and correct operation of a computer system without concern for the information stored or processed by the computer.

Governments, military, corporations, financial institutions, hospitals, and private businesses amass a great deal of confidential information about their employees, customers, products, research, and financial status. Most of this information is now collected, processed and stored on electronic computers and transmitted across networks to other computers.

Should confidential information about a business' customers or finances or new product line fall into the hands of a competitor, such a breach of security could lead to lost business, law suits or even bankruptcy of the business. Protecting confidential information is a business requirement, and in many cases also an ethical and legal requirement.

For the individual, information security has a significant effect on privacy, which is viewed very differently in different cultures.

The field of information security has grown and evolved significantly in recent years. There are many ways of gaining entry into the field as a career. It offers many areas for specialization including: securing network(s) and allied infrastructure, securing applications and databases, security testing, information systems auditing, business continuity planning and digital forensics science, etc.

History

Since the early days of writing, heads of state and military commanders understood that it was necessary to provide some mechanism to protect the confidentiality of written correspondence and to have some means of detecting tampering.

Julius Caesar is credited with the invention of the Caesar cipher ca. 50 B.C., which was created in order to prevent his secret messages from being read should a message fall into the wrong hands.

World War II brought about many advancements in information security and marked the beginning of the professional field of information security.

The end of the 20th century and early years of the 21st century saw rapid advancements in telecommunications, computing hardware and software, and data encryption. The availability of smaller, more powerful and less expensive computing equipment made electronic data processing within the reach of small business and the home user. These computers quickly became interconnected through a network generically called the Internet or World Wide Web.

The rapid growth and widespread use of electronic data processing and electronic business conducted through the Internet, along with numerous occurrences of international terrorism, fueled the need for better methods of protecting the computers and the information they store, process and transmit. The academic disciplines of computer security, information security and information assurance emerged along with numerous professional organizations - all sharing the common goals of ensuring the security and reliability of information systems.

Basic principles

Key concepts

For over twenty years, information security has held confidentiality, integrity and availability (known as the CIA triad) to be the core principles of information security.

There is continuous debate about extending this classic trio. Other principles such as Accountability have sometimes been proposed for addition - it has been pointed out that issues such as Non-Repudiation do not fit well within the three core concepts, and as regulation of computer systems has increased (particularly amongst the Western nations) Legality is becoming a key consideration for practical security installations.

In 2002, Donn Parker proposed an alternative model for the classic CIA triad that he called the six atomic elements of information. The elements are confidentiality, possession, integrity, authenticity, availability, and utility. The merits of the Parkerian hexad are a subject of debate amongst security professionals.

Confidentiality

Confidentiality is the term used to prevent the disclosure of information to unauthorized individuals or systems. For example, a credit card transaction on the Internet requires the credit card number to be transmitted from the buyer to the merchant and from the merchant to a transaction processing network. The system attempts to enforce confidentiality by encrypting the card number during transmission, by limiting the places where it might appear (in databases, log files, backups, printed receipts, and so on), and by restricting access to the places where it is stored. If an unauthorized party obtains the card number in any way, a breach of confidentiality has occurred.

Breaches of confidentiality take many forms. Permitting someone to look over your shoulder at your computer screen while you have confidential data displayed on it could be a breach of confidentiality. If a laptop computer containing sensitive information about a company's employees is stolen or sold, it could result in a breach of confidentiality. Giving out confidential information over the telephone is a breach of confidentiality if the caller is not authorized to have the information.

Confidentiality is necessary (but not sufficient) for maintaining the privacy of the people whose personal information a system holds.

Integrity

In information security, integrity means that data cannot be modified undetectably. This is not the same thing as referential integrity in databases, although it can be viewed as a special case of Consistency as understood in the classic ACID model of transaction processing. Integrity is violated when a message is actively modified in transit. Information security systems typically provide message integrity in addition to data confidentiality.

Availability

For any information system to serve its purpose, the information must be available when it is needed. This means that the computing systems used to store and process the information, the security controls used to protect it, and the communication channels used to access it must be functioning correctly. High availability systems aim to remain available at all times, preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks.

Authenticity

In computing, e-Business and information security it is necessary to ensure that the data, transactions, communications or documents (electronic or physical) are genuine. It is also important for authenticity to validate that both parties involved are who they claim they are.

Non-repudiation

In law, non-repudiation implies one's intention to fulfill their obligations to a contract. It also implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction.

Electronic commerce uses technology such as digital signatures and encryption to establish authenticity and non-repudiation.

Risk management

The CISA Review Manual 2006 provides the following definition of risk management: *"Risk management is the process of identifying vulnerabilities and threats to the information resources used by an organization in achieving business objectives, and deciding what countermeasures, if any, to take in reducing risk to an acceptable level, based on the value of the information resource to the organization."*

There are two things in this definition that may need some clarification. First, the *process* of risk management is an ongoing iterative process. It must be repeated indefinitely. The business environment is constantly changing and new threats and vulnerability emerge every day. Second, the choice of countermeasure (computer)s (controls) used to manage risks must strike a balance between productivity, cost, effectiveness of the countermeasure, and the value of the informational asset being protected.

Risk is the likelihood that something bad will happen that causes harm to an informational asset (or the loss of the asset). A **vulnerability** is a weakness that could be used to endanger or cause harm to an informational asset. A **threat** is anything (man made or act of nature) that has the potential to cause harm.

The likelihood that a threat will use a vulnerability to cause harm creates a risk. When a threat does use a vulnerability to inflict harm, it has an impact. In the context of information security, the impact is a loss of availability, integrity, and confidentiality, and possibly other losses (lost income, loss of life, loss of real property). It should be pointed out that it is not possible to identify all risks, nor is it possible to eliminate all risk. The remaining risk is called *residual risk*.

A risk assessment is carried out by a team of people who have knowledge of specific areas of the business. Membership of the team may vary over time as different parts of the business are assessed. The assessment may use a subjective qualitative analysis based on informed opinion, or where reliable dollar figures and historical information is available, the analysis may use quantitative analysis.

The research has shown that the most vulnerable point in most information systems is the human user, operator, designer, or other human. The ISO/IEC 27002:2005 Code of practice for information security management recommends the following be examined during a risk assessment:

- security policy,
- organization of information security,
- asset management,
- human resources security,
- physical and environmental security,

- communications and operations management,
- access control,
- information systems acquisition, development and maintenance,
- information security incident management,
- business continuity management, and
- regulatory compliance.

In broad terms the risk management process consists of:

1. Identification of assets and estimating their value. Include: people, buildings, hardware, software, data (electronic, print, other), supplies.
2. Conduct a threat assessment. Include: Acts of nature, acts of war, accidents, malicious acts originating from inside or outside the organization.
3. Conduct a vulnerability assessment, and for each vulnerability, calculate the probability that it will be exploited. Evaluate policies, procedures, standards, training, physical security, quality control, technical security.
4. Calculate the impact that each threat would have on each asset. Use qualitative analysis or quantitative analysis.
5. Identify, select and implement appropriate controls. Provide a proportional response. Consider productivity, cost effectiveness, and value of the asset.
6. Evaluate the effectiveness of the control measures. Ensure the controls provide the required cost effective protection without discernible loss of productivity.

For any given risk, Executive Management can choose to **accept the risk** based upon the relative low value of the asset, the relative low frequency of occurrence, and the relative low impact on the business. Or, leadership may choose to **mitigate the risk** by selecting and implementing appropriate control measures to reduce the risk. In some cases, the risk can be **transferred** to another business by buying insurance or out-sourcing to another business. The reality of some risks may be disputed. In such cases leadership may choose to **deny the risk**. This is itself a potential risk.

Controls

When Management chooses to mitigate a risk, they will do so by implementing one or more of three different types of controls.

Administrative

Administrative controls (also called procedural controls) consist of approved written policies, procedures, standards and guidelines. Administrative controls form the framework for running the business and managing people. They inform people on how the business is to be run and how day to day operations are to be conducted. Laws and regulations created by government bodies are also a type of administrative control because they inform the business. Some industry sectors have policies, procedures, standards and guidelines that must be followed - the Payment Card Industry (PCI) Data Security Standard required by Visa and Master Card is such an example. Other examples of administrative controls include the corporate security policy, password policy, hiring policies, and disciplinary policies.

Administrative controls form the basis for the selection and implementation of logical and physical controls. Logical and physical controls are manifestations of administrative controls. Administrative controls are of paramount importance.

Logical

Logical controls (also called technical controls) use software and data to monitor and control access to information and computing systems. For example: passwords, network and host based firewalls, network intrusion detection systems, access control lists, and data encryption are logical controls.

An important logical control that is frequently overlooked is the **principle of least privilege**. The principle of least privilege requires that an individual, program or system process is not granted any more access privileges than are necessary to perform the task. A blatant example of the failure to adhere to the principle of least privilege is logging into Windows as user Administrator to read Email and surf the Web. Violations of this principle can also occur when an individual collects additional access privileges over time. This happens when employees' job duties change, or they are promoted to a new position, or they transfer to another department. The access privileges required by their new duties are frequently added onto their already existing access privileges which may no longer be necessary or appropriate.

Physical

Physical controls monitor and control the environment of the work place and computing facilities. They also monitor and control access to and from such facilities. For example: doors, locks, heating and air conditioning, smoke and fire alarms, fire suppression systems, cameras, barricades, fencing, security guards, cable locks, etc. Separating the network and work place into functional areas are also physical controls.

An important physical control that is frequently overlooked is the **separation of duties**. Separation of duties ensures that an individual can not complete a critical task by himself. For example: an employee who submits a request for reimbursement should not also be able to authorize payment or print the check. An applications programmer should not also be the server administrator or the database administrator - these roles and responsibilities must be separated from one another.

Security classification for information

An important aspect of information security and risk management is recognizing the value of information and defining appropriate procedures and protection requirements for the information. Not all information is equal and so not all information requires the same degree of protection. This requires information to be assigned a security classification.

The first step in information classification is to identify a member of senior management as the owner of the particular information to be classified. Next, develop a classification policy. The policy should describe the different classification labels, define the criteria for information to be assigned a particular label, and list the required security controls for each classification.

Some factors that influence which classification information should be assigned include how much value that information has to the organization, how old the information is and whether or not the information has become obsolete. Laws and other regulatory requirements are also important considerations when classifying information.

The type of information security classification labels selected and used will depend on the nature of the organisation, with examples being:

- In the business sector, labels such as: **Public, Sensitive, Private, Confidential**.
- In the government sector, labels such as: **Unclassified**, **Sensitive But Unclassified**, **Restricted**, **Confidential**, **Secret**, **Top Secret** and their non-English equivalents.
- In cross-sectoral formations, the Traffic Light Protocol, which consists of: **White, Green, Amber** and **Red**.

All employees in the organization, as well as business partners, must be trained on the classification schema and understand the required security controls and handling procedures for each classification. The classification a particular information asset has been assigned should be reviewed periodically to ensure the classification is still appropriate for the information and to ensure the security controls required by the classification are in place.

Access control

Access to protected information must be restricted to people who are authorized to access the information. The computer programs, and in many cases the computers that process the information, must also be authorized. This requires that mechanisms be in place to control the access to protected information. The sophistication of the access control mechanisms should be in parity with the value of the information being protected - the more sensitive or valuable the information the stronger the control mechanisms need to be. The foundation on which access control mechanisms are built start with identification and authentication.

Identification is an assertion of who someone is or what something is. If a person makes the statement *"Hello, my name is John Doe"* they are making a claim of who they are. However, their claim may or may not be true. Before John Doe can be granted access to protected information it will be necessary to verify that the person claiming to be John Doe really is John Doe.

Authentication is the act of verifying a claim of identity. When John Doe goes into a bank to make a withdrawal, he tells the bank teller he is John Doe (a claim of identity). The bank teller asks to see a photo ID, so he hands the teller his driver's license. The bank teller checks the license to make sure it has John Doe printed on it and compares the photograph on the license against the person claiming to be John Doe. If the photo and name match the person, then the teller has authenticated that John Doe is who he claimed to be.

There are three different types of information that can be used for authentication: **something you know, something you have, or something you are.** Examples of *something you know* include such things as a PIN, a password, or your mother's maiden name. Examples of *something you have* include a driver's license or a magnetic swipe card. *Something you are* refers to biometrics. Examples of biometrics include palm prints, finger prints, voice prints and retina (eye) scans. Strong authentication requires providing information from two of the three different types of authentication information. For example, something you know plus something you have. This is called two factor authentication.

On computer systems in use today, the Username is the most common form of identification and the Password is the most common form of authentication. Usernames and passwords have served their purpose but in our modern world they are no longer adequate. Usernames and passwords are slowly being replaced with more sophisticated authentication mechanisms.

After a person, program or computer has successfully been identified and authenticated then it must be determined what informational resources they are permitted to access and what actions they will be allowed to perform (run, view, create, delete, or change). This is called **authorization**.

Authorization to access information and other computing services begins with administrative policies and procedures. The policies prescribe what information and computing services can be accessed, by whom, and under what conditions. The access control mechanisms are then configured to enforce these policies.

Different computing systems are equipped with different kinds of access control mechanisms - some may even offer a choice of different access control mechanisms. The access control mechanism a system offers will be based upon one of three approaches to access control or it may be derived from a combination of the three approaches.

The **non-discretionary** approach consolidates all access control under a centralized administration. The access to information and other resources is usually based on the individuals function (role) in the organization or the tasks the individual must perform. The **discretionary approach** gives the creator or owner of the information resource the ability to control access to those resources. In the **Mandatory access control approach**, access is granted or denied basing upon the security classification assigned to the information resource.

Examples of common access control mechanisms in use today include Role-based access control available in many advanced Database Management Systems, simple file permissions provided in the UNIX and Windows operating systems, Group Policy Objects provided in Windows network systems, Kerberos, RADIUS, TACACS, and the simple access lists used in many firewalls and routers.

To be effective, policies and other security controls must be enforceable and upheld. Effective policies ensure that people are held **accountable** for their actions. All failed and successful authentication attempts must be logged, and all access to information must leave some type of audit trail.

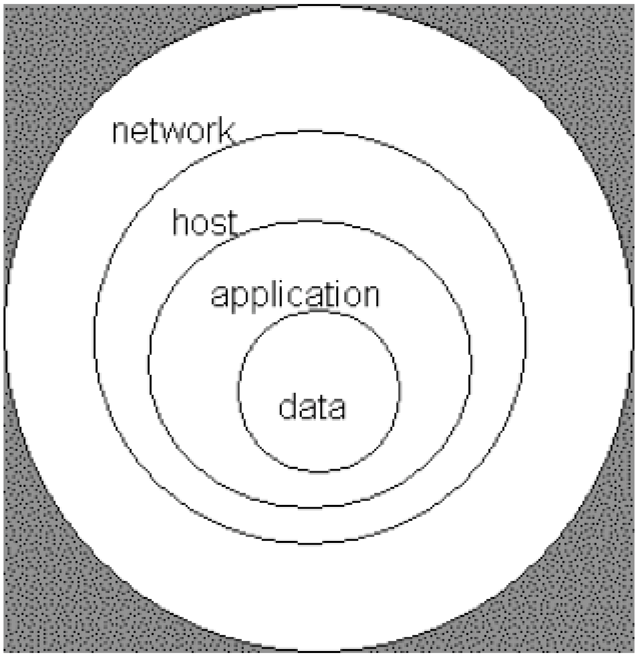
Cryptography

Information security uses cryptography to transform usable information into a form that renders it unusable by anyone other than an authorized user; this process is called encryption. Information that has been encrypted (rendered unusable) can be transformed back into its original usable form by an authorized user, who possesses the cryptographic key, through the process of decryption. Cryptography is used in information security to protect information from unauthorized or accidental disclosure while the information is in transit (either electronically or physically) and while information is in storage.

Cryptography provides information security with other useful applications as well including improved authentication methods, message digests, digital signatures, non-repudiation, and encrypted network communications. Older less secure application such as telnet and ftp are slowly being replaced with more secure applications such as ssh that use encrypted network communications. Wireless communications can be encrypted using protocols such as WPA/WPA2 or the older (and less secure) WEP. Wired communications (such as ITU-T G.hn) are secured using AES for encryption and X.1035 for authentication and key exchange. Software applications such as GnuPG or PGP can be used to encrypt data files and Email.

Cryptography can introduce security problems when it is not implemented correctly. Cryptographic solutions need to be implemented using industry accepted solutions that have undergone rigorous peer review by independent experts in cryptography. The length and strength of the encryption key is also an important consideration. A key that is weak or too short will produce weak encryption. The keys used for encryption and decryption must be protected with the same degree of rigor as any other confidential information. They must be protected from unauthorized disclosure and destruction and they must be available when needed. PKI solutions address many of the problems that surround key management.

Defense in depth



Information security must protect information throughout the life span of the information, from the initial creation of the information on through to the final disposal of the information. The information must be protected while in motion and while at rest. During its life time, information may pass through many different information processing systems and through many different parts of information processing systems. There are many different ways the information and information systems can be threatened. To fully protect the information during its lifetime, each component of the information processing system must have its own protection mechanisms. The building up, layering on and overlapping of security measures is called defense in depth. The strength of any system is no greater than its weakest link. Using a defence in depth strategy, should one defensive measure fail there are other defensive measures in place that continue to provide protection.

Recall the earlier discussion about administrative controls, logical controls, and physical controls. The three types of controls can be used to form the basis upon which to build a defense-in-depth strategy. With this approach, defense-in-depth can be conceptualized as three distinct layers or planes laid one on top of the other. Additional insight into defense-in- depth can be gained by thinking of it as forming the layers of an onion, with data at the core of the onion, people as the outer layer of the onion, and network security, host-based security and application security forming the inner layers of the onion. Both perspectives are equally valid and each provides valuable insight into the implementation of a good defense-in-depth strategy.

Process

The terms **reasonable and prudent person**, **due care** and **due diligence** have been used in the fields of Finance, Securities, and Law for many years. In recent years these terms have found their way into the fields of computing and information security. U.S.A. Federal Sentencing Guidelines now make it possible to hold corporate officers liable for failing to exercise due care and due diligence in the management of their information systems.

In the business world, stockholders, customers, business partners and governments have the expectation that corporate officers will run the business in accordance with accepted business practices and in compliance with laws and other regulatory requirements. This is often described as the "reasonable and prudent person" rule. A prudent person takes due care to ensure that everything necessary is done to operate the business by sound business principles and in a legal ethical manner. A prudent person is also diligent (mindful, attentive, and ongoing) in their due care of the business.

In the field of Information Security, Harris offers the following definitions of **due care** and **due diligence**:

"Due care are steps that are taken to show that a company has taken responsibility for the activities that take place within the corporation and has taken the necessary steps to help protect the company, its resources, and employees." And, [Due diligence are the] *"continual activities that make sure the protection mechanisms are continually maintained and operational."*

Attention should be made to two important points in these definitions. First, in due care, steps are taken to ***show*** - this means that the steps can be verified, measured, or even produce tangible artifacts. Second, in due diligence, there are **continual activities** - this means that people are actually doing things to monitor and maintain the protection mechanisms, and these activities are ongoing.

Security governance

The Software Engineering Institute at Carnegie Mellon University, in a publication titled "Governing for Enterprise Security (GES)", defines characteristics of effective security governance. These include:

- An enterprise-wide issue
- Leaders are accountable
- Viewed as a business requirement
- Risk-based
- Roles, responsibilities, and segregation of duties defined
- Addressed and enforced in policy
- Adequate resources committed
- Staff aware and trained

- A development life cycle requirement
- Planned, managed, measurable, and measured
- Reviewed and audited

Incident response plans

1 to 3 paragraphs (non technical) that discuss:

- Selecting team members
- Define roles, responsibilities and lines of authority
- Define a security incident
- Define a reportable incident
- Training
- Detection
- Classification
- Escalation
- Containment
- Eradication
- Documentation

Change management

Change management is a formal process for directing and controlling alterations to the information processing environment. This includes alterations to desktop computers, the network, servers and software. The objectives of change management are to reduce the risks posed by changes to the information processing environment and improve the stability and reliability of the processing environment as changes are made. It is not the objective of change management to prevent or hinder necessary changes from being implemented.

Any change to the information processing environment introduces an element of risk. Even apparently simple changes can have unexpected effects. One of Managements many responsibilities is the management of risk. Change management is a tool for managing the risks introduced by changes to the information processing environment. Part of the change management process ensures that changes are not implemented at inopportune times when they may disrupt critical business processes or interfere with other changes being implemented.

Not every change needs to be managed. Some kinds of changes are a part of the everyday routine of information processing and adhere to a predefined procedure, which reduces the overall level of risk to the processing environment. Creating a new user account or deploying a new desktop computer are examples of changes that do not generally require change management. However, relocating user file shares, or upgrading the Email server pose a much higher level of risk to the processing environment and are not a normal everyday activity. The critical first steps in change management are (a) defining change (and communicating that definition) and (b) defining the scope of the change system.

Change management is usually overseen by a Change Review Board composed of representatives from key business areas, security, networking, systems administrators, Database administration, applications development, desktop support and the help desk. The tasks of the Change Review Board can be facilitated with the use of automated work flow application. The responsibility of the Change Review Board is to ensure the organizations documented change management procedures are followed. The change management process is as follows:

- **Requested:** Anyone can request a change. The person making the change request may or may not be the same person that performs the analysis or implements the change. When a request for change is received, it may undergo a preliminary review to determine if the requested change is compatible with the organizations business model and practices, and to determine the amount of resources needed to implement the change.

- **Approved:** Management runs the business and controls the allocation of resources therefore, Management must approve requests for changes and assign a priority for every change. Management might choose to reject a change request if the change is not compatible with the business model, industry standards or best practices. Management might also choose to reject a change request if the change requires more resources than can be allocated for the change.

- **Planned:** Planning a change involves discovering the scope and impact of the proposed change; analyzing the complexity of the change; allocation of resources and, developing, testing and documenting both implementation and backout plans. Need to define the criteria on which a decision to back out will be made.

- **Tested:** Every change must be tested in a safe test environment, which closely reflects the actual production environment, before the change is applied to the production environment. The backout plan must also be tested.

- **Scheduled:** Part of the change review board's responsibility is to assist in the scheduling of changes by reviewing the proposed implementation date for potential conflicts with other scheduled changes or critical business activities.

- **Communicated:** Once a change has been scheduled it must be communicated. The communication is to give others the opportunity to remind the change review board about other changes or critical business activities that might have been overlooked when scheduling the change. The communication also serves to make the Help Desk and users aware that a change is about to occur. Another responsibility of the change review board is to ensure that scheduled changes have been properly communicated to those who will be affected by the change or otherwise have an interest in the change.

- **Implemented:** At the appointed date and time, the changes must be implemented. Part of the planning process was to develop an implementation plan, testing plan and, a back out plan. If the implementation of the change should fail or, the post implementation testing fails or, other "drop dead" criteria have been met, the back out plan should be implemented.

- **Documented:** All changes must be documented. The documentation includes the initial request for change, its approval, the priority assigned to it, the implementation, testing and back out plans, the results of the change review board critique, the date/time the change was implemented, who implemented it, and whether the change was implemented successfully, failed or postponed.

- **Post change review:** The change review board should hold a post implementation review of changes. It is particularly important to review failed and backed out changes. The review board should try to understand the problems that were encountered, and look for areas for improvement.

Change management procedures that are simple to follow and easy to use can greatly reduce the overall risks created when changes are made to the information processing environment. Good change management procedures improve the over all quality and success of changes as they are implemented. This is accomplished through planning, peer review, documentation and communication.

ISO/IEC 20000, The Visible OPS Handbook: Implementing ITIL in 4 Practical and Auditable Steps (Full book summary), and Information Technology Infrastructure Library all provide valuable guidance on implementing an efficient and effective change management program information security

Business continuity

Business continuity is the mechanism by which an organization continues to operate its critical business units, during planned or unplanned disruptions that affect normal business operations, by invoking planned and managed procedures.

Unlike what most people think business continuity is not necessarily an IT system or process, simply because it is about the business. Today disasters or disruptions to business are a reality. Whether the disaster is natural or man-made (the TIME magazine has a website on the top 10), it affects normal life and so business. So why is planning so important? Let us face reality that "all businesses recover", whether they planned for recovery or not, simply because business is about earning money for survival.

The planning is merely getting better prepared to face it, knowing fully well that the best plans may fail. Planning helps to reduce cost of recovery, operational overheads and most importantly sail through some smaller ones effortlessly.

For businesses to create effective plans they need to focus upon the following key questions. Most of these are common knowledge, and anyone can do a BCP.

1. Should a disaster strike, what are the first few things that I should do? Should I call people to find if they are OK or call up the bank to figure out my money is safe? This is Emergencey Response. Emergency Response services help take the first hit when the disaster strikes and if the disaster is serious enough the Emergency Response teams need to quickly get a Crisis Management team in place.
2. What parts of my business should I recover first? The one that brings me most money or the one where I spend the most, or the one that will ensure I shall be able to get sustained future growth? The identified sections are the critical business units. There is no magic bullet here, no one answer satisfies all. Businesses need to find answers that meet business requirements.
3. How soon should I target to recover my critical business units? In BCP technical jargon this is called Recovery Time Objective, or RTO. This objective will define what costs the business will need to spend to recover from a disruption. For example, it is cheaper to recover a business in 1 day than in 1 hour.
4. What all do I need to recover the business? IT, machinery, records...food, water, people...So many aspects to dwell upon. The cost factor becomes clearer now...Business leaders need to drive business continuity. Hold on. My IT manager spent $200000 last month and created a DRP (Disaster Recovery Plan), whatever happened to that? a DRP is about continuing an IT system, and is one of the sections of a comprehensive Business Continuity Plan. Look below for more on this.
5. And where do I recover my business from... Will the business center give me space to work, or would it be flooded by many people queuing up for the same reasons that I am.
6. But once I do recover from the disaster and work in reduced production capacity, since my main operational sites are unavailable, how long can this go on. How long can I do without my original sites, systems, people? this defines the amount of business resilience a business may have.
7. Now that I know how to recover my business. How do I make sure my plan works? Most BCP pundits would recommend testing the plan at least once a year, reviewing it for adequacy and rewriting or updating the plans either annually or when businesses change.

Disaster recovery planning

While a business continuity plan (BCP) takes a broad approach to dealing with organizational-wide effects of a disaster, a disaster recovery plan (DRP), which is a subset of the business continuity plan, is instead focused on taking the necessary steps to resume normal business operations as quickly as possible. A disaster recovery plan is executed immediately after the disaster occurs and details what steps are to be taken in order to recover critical information technology infrastructure.

Laws and regulations

*Below is a **partial** listing of European, United Kingdom, Canadian and USA governmental laws and regulations that have, or will have, a significant effect on data processing and information security. Important industry sector regulations have also been included when they have a significant impact on information security.*

- UK Data Protection Act 1998 makes new provisions for the regulation of the processing of information relating to individuals, including the obtaining, holding, use or disclosure of such information. The European Union Data Protection Directive (EUDPD) requires that all EU member must adopt national regulations to standardize the protection of data privacy for citizens throughout the EU.

- The Computer Misuse Act 1990 is an Act of the UK Parliament making computer crime (e.g. cracking - sometimes incorrectly referred to as hacking) a criminal offence. The Act has become a model upon which several other countries including Canada and the Republic of Ireland have drawn inspiration when subsequently drafting their own information security laws.
- EU Data Retention laws requires Internet service providers and phone companies to keep data on every electronic message sent and phone call made for between six months and two years.
- The Family Educational Rights and Privacy Act (FERPA) (20 U.S.C. § 1232 g; 34 CFR Part 99) is a USA Federal law that protects the privacy of student education records. The law applies to all schools that receive funds under an applicable program of the U.S. Department of Education. Generally, schools must have written permission from the parent or eligible student in order to release any information from a student's education record.

- Health Insurance Portability and Accountability Act (HIPAA) of 1996 requires the adoption of national standards for electronic health care transactions and national identifiers for providers, health insurance plans, and employers. And, it requires health care providers, insurance providers and employers to safeguard the security and privacy of health data.
- Gramm-Leach-Bliley Act of 1999 (GLBA), also known as the Financial Services Modernization Act of 1999, protects the privacy and security of private financial information that financial institutions collect, hold, and process.

- Sarbanes-Oxley Act of 2002 (SOX). Section 404 of the act requires publicly traded companies to assess the effectiveness of their internal controls for financial reporting in annual reports they submit at the end of each fiscal year. Chief information officers are responsible for the security, accuracy and the reliability of the systems that manage and report the financial data. The act also requires publicly traded companies to engage independent auditors who must attest to, and report on, the validity of their assessments.
- Payment Card Industry Data Security Standard (PCI DSS) establishes comprehensive requirements for enhancing payment account data security. It was

developed by the founding payment brands of the PCI Security Standards Council, including American Express, Discover Financial Services, JCB, MasterCard Worldwide and Visa International, to help facilitate the broad adoption of consistent data security measures on a global basis. The PCI DSS is a multifaceted security standard that includes requirements for security management, policies, procedures, network architecture, software design and other critical protective measures.

- State Security Breach Notification Laws (California and many others) require businesses, nonprofits, and state institutions to notify consumers when unencrypted "personal information" may have been compromised, lost, or stolen.
- Personal Information Protection and Electronics Document Act (PIPEDA) - An Act to support and promote electronic commerce by protecting personal information that is collected, used or disclosed in certain circumstances, by providing for the use of electronic means to communicate or record information or transactions and by amending the Canada Evidence Act, the Statutory Instruments Act and the Statute Revision Act.

Sources of standards

International Organization for Standardization (ISO) is a consortium of national standards institutes from 157 countries, coordinated through a secretariat in Geneva, Switzerland. ISO is the world's largest developer of standards. ISO 15443: "Information technology - Security techniques - A framework for IT security assurance", ISO/IEC 27002: "Information technology - Security techniques - Code of practice for information security management", ISO-20000: "Information technology - Service management", and ISO/IEC27001: "Information technology - Security techniques - Information security management systems - Requirements" are of particular interest to information security professionals.

The USA National Institute of Standards and Technology (NIST) is a non-regulatory federal agency within the U.S. Department of Commerce. The NIST Computer Security Division develops standards, metrics, tests and validation programs as well as publishes standards and guidelines to increase secure IT planning, implementation, management and operation. NIST is also the custodian of the USA Federal Information Processing Standard publications (FIPS).

The Internet Society is a professional membership society with more than 100 organization and over 20,000 individual members in over 180 countries. It provides leadership in addressing issues that confront the future of the Internet, and is the organization home for the groups responsible for Internet infrastructure standards, including the Internet Engineering Task Force (IETF) and the Internet Architecture Board (IAB). The ISOC hosts the Requests for Comments (RFCs) which includes the Official Internet Protocol Standards and the RFC-2196 Site Security Handbook.

The Information Security Forum is a global nonprofit organization of several hundred leading organizations in financial services, manufacturing, telecommunications, consumer goods, government, and other areas. It undertakes research into information security practices and offers advice in its biannual Standard of Good Practice and more detailed advisories for members.

The IT Baseline Protection Catalogs, or IT-Grundschutz Catalogs, ("IT Baseline Protection Manual" before 2005) are a collection of documents from the German Federal Office for Security in Information Technology (FSI), useful for detecting and combating security-relevant weak points in the IT environment ("IT cluster"). The collection encompasses over 3000 pages with the introduction and catalogs.

Professionalism

Information security professionalism is the set of knowledge that people working in **Information security** and similar fields (Information Assurance and Computer security) should have and eventually demonstrate through certifications from well respected organizations.

It also encompasses the education process required to accomplish different tasks in these fields.

Information technology adoption is always increasing and spread to vital infrastructure for civil and military organizations. Everybody can get involved in the Cyberwar. It is crucial that a nation can have skilled professional to defend its vital interests.

Conclusion

Information security is the ongoing process of exercising due care and due diligence to protect information, and information systems, from unauthorized access, use, disclosure, destruction, modification, or disruption or distribution. The never ending process of information security involves ongoing training, assessment, protection, monitoring & detection, incident response & repair, documentation, and review. This makes information security an indispensable part of all the business operations across different domains.

Chapter-5

Access Control

Access control is a system which enables an authority to control access to areas and resources in a given physical facility or computer-based information system. An access control system, within the field of physical security, is generally seen as the second layer in the security of a physical structure.

Access control is, in reality, an everyday phenomenon. A lock on a car door is essentially a form of access control. A PIN on an ATM system at a bank is another means of access control. Bouncers standing in front of a night club is perhaps a more primitive mode of access control (given the evident lack of information technology involved). The possession of access control is of prime importance when persons seek to secure important, confidential, or sensitive information and equipment.

Item control or electronic key management is an area within (and possibly integrated with) an access control system which concerns the managing of possession and location of small assets or physical (mechanical) keys.

Physical access

Underground entrance to the New York City Subway system

Physical access by a person may be allowed depending on payment, authorization, etc. Also there may be one-way traffic of people. These can be enforced by personnel such as a border guard, a doorman, a ticket checker, etc., or with a device such as a turnstile. There may be fences to avoid circumventing this access control. An alternative of access control in the strict sense (physically controlling access itself) is a system of checking authorized presence.

In physical security, the term access control refers to the practice of restricting entrance to a property, a building, or a room to authorized persons. Physical access control can be achieved by a human (a guard, bouncer, or receptionist), through mechanical means such as locks and keys, or through technological means such as access control systems like the Access control vestibule. Within these environments, physical key management may also be employed as a means of further managing and monitoring access to mechanically keyed areas or access to certain small assets.

Physical access control is a matter of who, where, and when. An access control system determines who is allowed to enter or exit, where they are allowed to exit or enter, and when they are allowed to enter or exit. Historically this was partially accomplished through keys and locks. When a door is locked only someone with a key can enter through the door depending on how the lock is configured. Mechanical locks and keys do not allow restriction of the key holder to specific times or dates. Mechanical locks and keys do not provide records of the key used on any specific door and the keys can be easily copied or transferred to an unauthorized person. When a mechanical key is lost or the key holder is no longer authorized to use the protected area, the locks must be re-keyed.

Electronic access control uses computers to solve the limitations of mechanical locks and keys. A wide range of credentials can be used to replace mechanical keys. The electronic access control system grants access based on the credential presented. When access is granted, the door is unlocked for a predetermined time and the transaction is recorded. When access is refused, the door remains locked and the attempted access is recorded. The system will also monitor the door and alarm if the door is forced open or held open too long after being unlocked.

Access control system operation

When a credential is presented to a reader, the reader sends the credential’s information, usually a number, to a control panel, a highly reliable processor. The control panel compares the credential's number to an access control list, grants or denies the presented request, and sends a transaction log to a database. When access is denied based on the access control list, the door remains locked. If there is a match between the credential and the access control list, the control panel operates a relay that in turn unlocks the door. The control panel also ignores a door open signal to prevent an alarm. Often the reader provides feedback, such as a flashing red LED for an access denied and a flashing green LED for an access granted.

The above description illustrates a single factor transaction. Credentials can be passed around, thus subverting the access control list. For example, Alice has access rights to the server room but Bob does not. Alice either gives Bob her credential or Bob takes it; he now has access to the server room. To prevent this, two-factor authentication can be used. In a two factor transaction, the presented credential and a second factor are needed for access to be granted; another factor can be a PIN, a second credential, operator intervention, or a biometric input.

There are three types (factors) of authenticating information:

- something the user knows, eg a password, pass-phrase or PIN
- something the user has, such as smart card
- something the user is, such as fingerprint, verified by biometric measurement

Passwords are a common means of verifying a user's identity before access is given to information systems. In addition, a fourth factor of authentication is now recognized: someone you know, where another person who knows you can provide a human element of authentication in situations where systems have been set up to allow for such scenarios. For example, a user may have their password, but have forgotten their smart card. In such a scenario, if the user is known to designated cohorts, the cohorts may provide their smart card and password in combination with the extant factor of the user in question and thus provide two factors for the user with missing credential, and three factors overall to allow access.

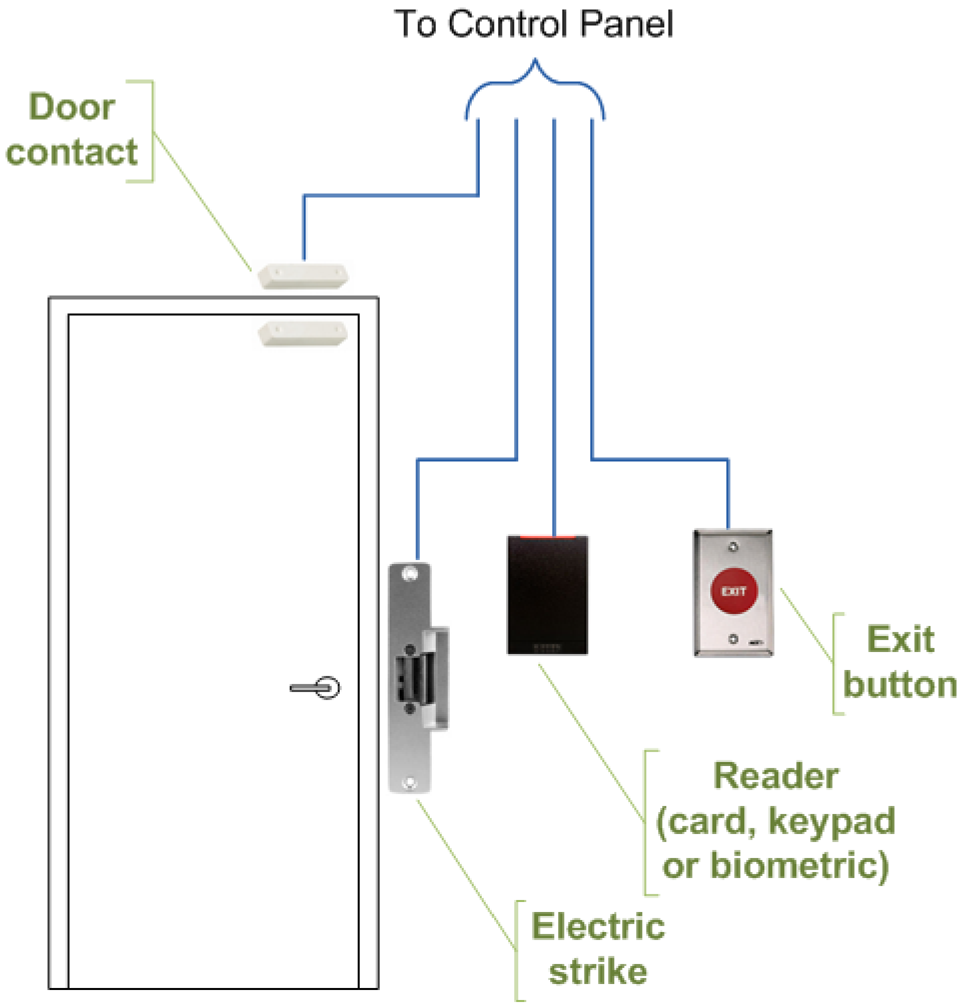
Credential

A credential is a physical/tangible object, a piece of knowledge, or a facet of a person's physical being, that enables an individual access to a given physical facility or computer-based information system. Typically, credentials can be something you know (such as number or PIN), something you have (such as an access badge), something you are (such as a biometric feature) or some combination of these items. The typical credential is an access card, key fob, or other key. There are many card technologies including magnetic stripe, bar code, Wiegand, 125 kHz proximity, 26 bit card-swipe, contact smart cards, and contactless smart cards. Also available are key-fobs which are more compact than ID cards and attach to a key ring. Typical biometric technologies include fingerprint, facial recognition, iris recognition, retinal scan, voice, and hand geometry.

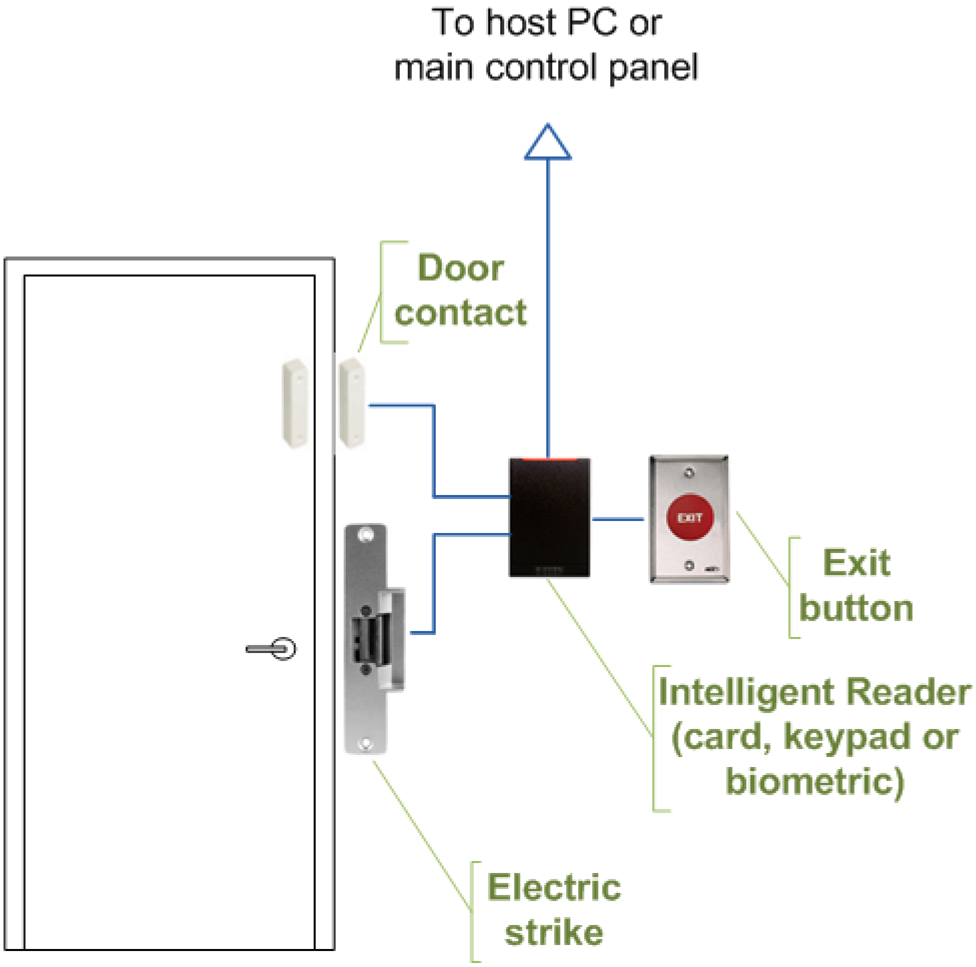
Access control system components

An access control point, which can be a door, turnstile, parking gate, elevator, or other physical barrier where granting access can be electronically controlled. Typically the access point is a door. An electronic access control door can contain several elements. At its most basic there is a stand-alone electric lock. The lock is unlocked by an operator with a switch. To automate this, operator intervention is replaced by a reader. The reader could be a keypad where a code is entered, it could be a card reader, or it could be a biometric reader. Readers do not usually make an access decision but send a card number to an access control panel that verifies the number against an access list. To monitor the door position a magnetic door switch is used. In concept the door switch is not unlike those on refrigerators or car doors. Generally only entry is controlled and exit is uncontrolled. In cases where exit is also controlled a second reader is used on the opposite side of the door. In cases where exit is not controlled, free exit, a device called a request-to-exit (RTE) is used. Request-to-exit devices can be a pushbutton or a motion detector. When the button is pushed or the motion detector detects motion at the door, the door alarm is temporarily ignored while the door is opened. Exiting a door without having to electrically unlock the door is called mechanical free egress. This is an important safety feature. In cases where the lock must be electrically unlocked on exit, the request-to-exit device also unlocks the door.

Access control topology



Typical access control door wiring



Access control door wiring when using intelligent readers

Access control decisions are made by comparing the credential to an access control list. This lookup can be done by a host or server, by an access control panel, or by a reader. The development of access control systems has seen a steady push of the lookup out from a central host to the edge of the system, or the reader. The predominate topology circa 2009 is hub and spoke with a control panel as the hub and the readers as the spokes. The lookup and control functions are by the control panel. The spokes communicate through a serial connection; usually RS485. Some manufactures are pushing the decision making to the edge by placing a controller at the door. The controllers are IP enabled and connect to a host and database using standard networks.

Types of readers

Access control readers may be classified by functions they are able to perform:

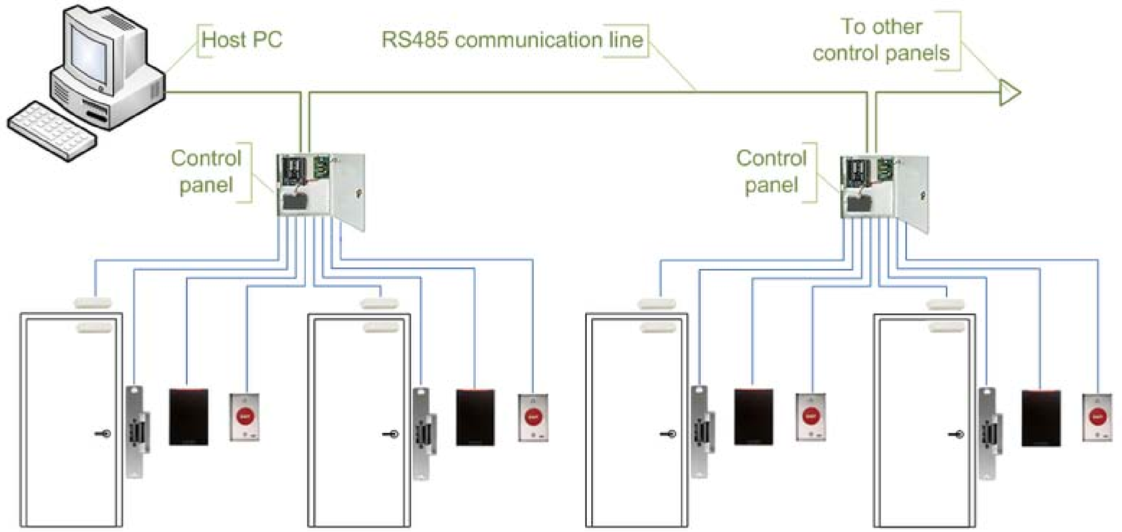
- Basic (non-intelligent) readers: simply read card number or PIN and forward it to a control panel. In case of biometric identification, such readers output ID number of a user. Typically Wiegand protocol is used for transmitting data to the control panel, but other options such as RS-232, RS-485 and Clock/Data are not uncommon. This is the most popular type of access control readers. Examples of such readers are RF Tiny by RFLOGICS, ProxPoint by HID, and P300 by Farpointe Data.

- Semi-intelligent readers: have all inputs and outputs necessary to control door hardware (lock, door contact, exit button), but do not make any access decisions. When a user presents a card or enters PIN, the reader sends information to the main controller and waits for its response. If the connection to the main controller is interrupted, such readers stop working or function in a degraded mode. Usually semi-intelligent readers are connected to a control panel via an RS-485 bus. Examples of such readers are InfoProx Lite IPL200 by CEM Systems and AP-510 by Apollo.
- Intelligent readers: have all inputs and outputs necessary to control door hardware, they also have memory and processing power necessary to make access decisions independently. Same as semi-intelligent readers they are connected to a control panel via an RS-485 bus. The control panel sends configuration updates and retrieves events from the readers. Examples of such readers could be InfoProx IPO200 by CEM Systems and AP-500 by Apollo. There is also a new generation of intelligent readers referred to as "IP readers". Systems with IP readers usually do not have traditional control panels and readers communicate directly to PC that acts as a host. Examples of such readers are PowerNet IP Reader by Isonas Security Systems, ID08 by Solus has the built in webservice to make it user friendly, Edge ER40 reader by HID Global, LogLock and UNiLOCK by ASPiSYS Ltd, and BioEntry Plus reader by Suprema Inc.

Some readers may have additional features such as LCD and function buttons for data collection purposes (i.e. clock-in/clock-out events for attendance reports), camera/speaker/microphone for intercom, and smart card read/write support.

Access control readers may also be classified by the type of identification technology.

Access control system topologies



Access control system using serial controllers

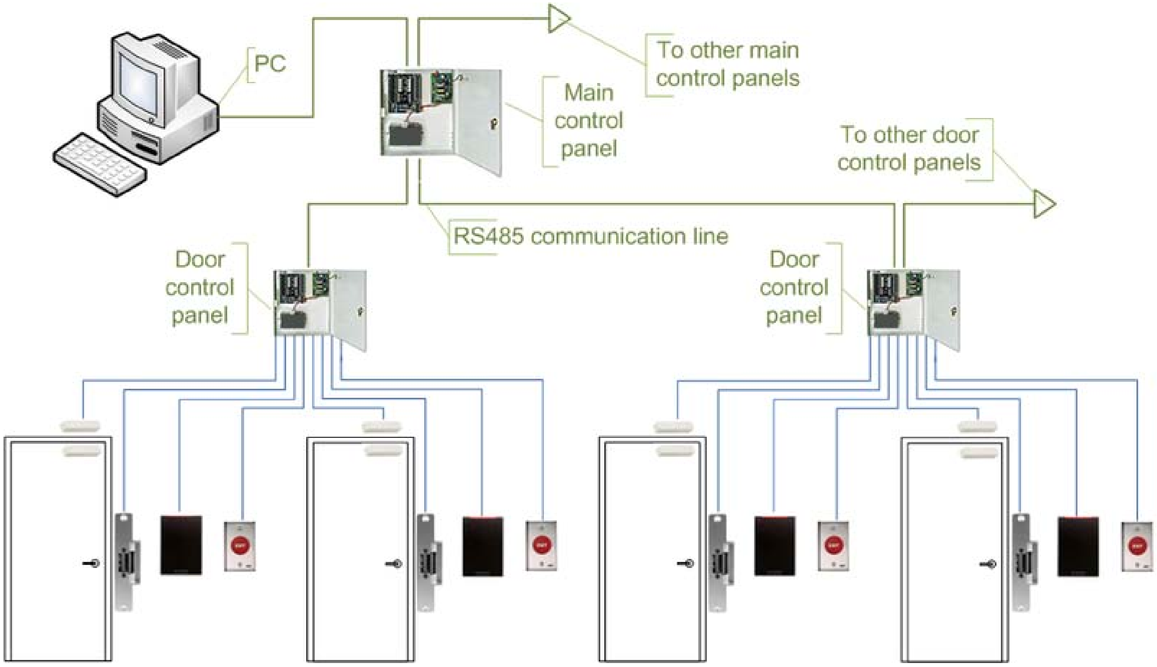
1. Serial controllers. Controllers are connected to a host PC via a serial RS-485 communication line (or via 20mA current loop in some older systems). External RS-232/485 converters or internal RS-485 cards have to be installed as standard PCs do not have RS-485 communication ports. In larger systems multi-port serial IO boards are used, Digi International being one of most popular options. Advantages:

- RS-485 standard allows long cable runs, up to 4000 feet (1200 m)
- Relatively short response time. The maximum number of devices on an RS-485 line is limited to 32, which means that the host can frequently request status updates from each device and display events almost in real time.
- High reliability and security as the communication line is not shared with any other systems.

Disadvantages:

- RS-485 does not allows Star-type wiring unless splitters are used
- RS-485 is not well suited for transferring large amounts of data (i.e. configuration and users). The highest possible throughput is 115.2 kbit/s, but in most system it is downgraded to 56.2 kbit/s or less to increase reliability.
- RS-485 does not allow host PC to communicate with several controllers connected to the same port simultaneously. Therefore in large systems transfers of configuration and users to controllers may take a very long time and interfere with normal operations.
- Controllers cannot initiate communication in case of an alarm. The host PC acts as a master on the RS-485 communication line and controllers have to wait till they are polled.
- Special serial switches are required in order to build a redundant host PC setup.

- Separate RS-485 lines have to be installed instead of using an already existing network infrastructure.
- Cable that meets RS-485 standards is significantly more expensive than the regular Category 5 UTP network cable.
- Operation of the system is highly dependent on the host PC. In case the host PC fails, events from controllers are not retrieved and functions that required interaction between controllers (i.e. anti-passback) stop working.



Access control system using serial main and sub-controllers

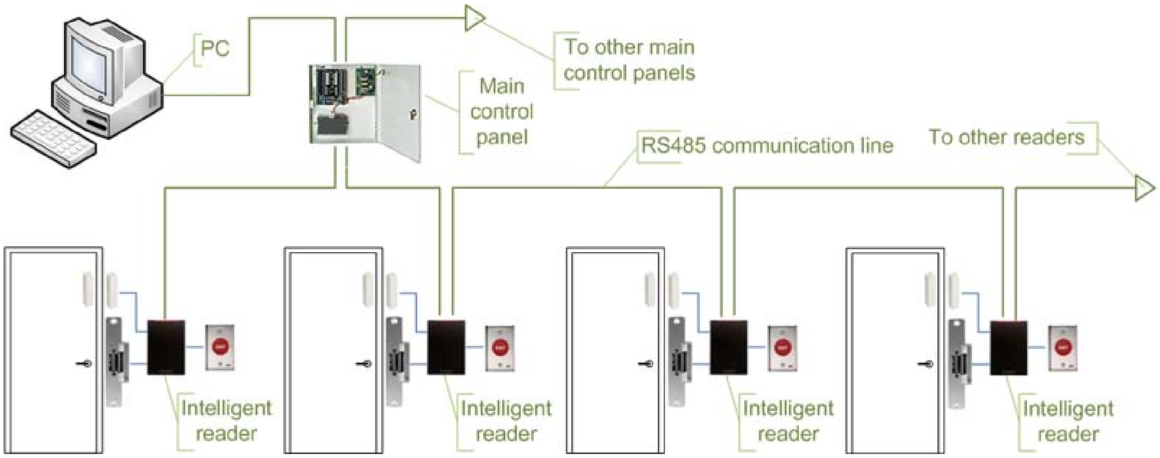
2. Serial main and sub-controllers. All door hardware is connected to sub-controllers (a.k.a. door controllers or door interfaces). Sub-controllers usually do not make access decisions, and forward all requests to the main controllers. Main controllers usually support from 16 to 32 sub-controllers. Advantages:

- Work load on the host PC is significantly reduced, because it only needs to communicate with a few main controllers.
- The overall cost of the system is lower, as sub-controllers are usually simple and inexpensive devices.
- All other advantages listed in the first paragraph apply.

Disadvantages:

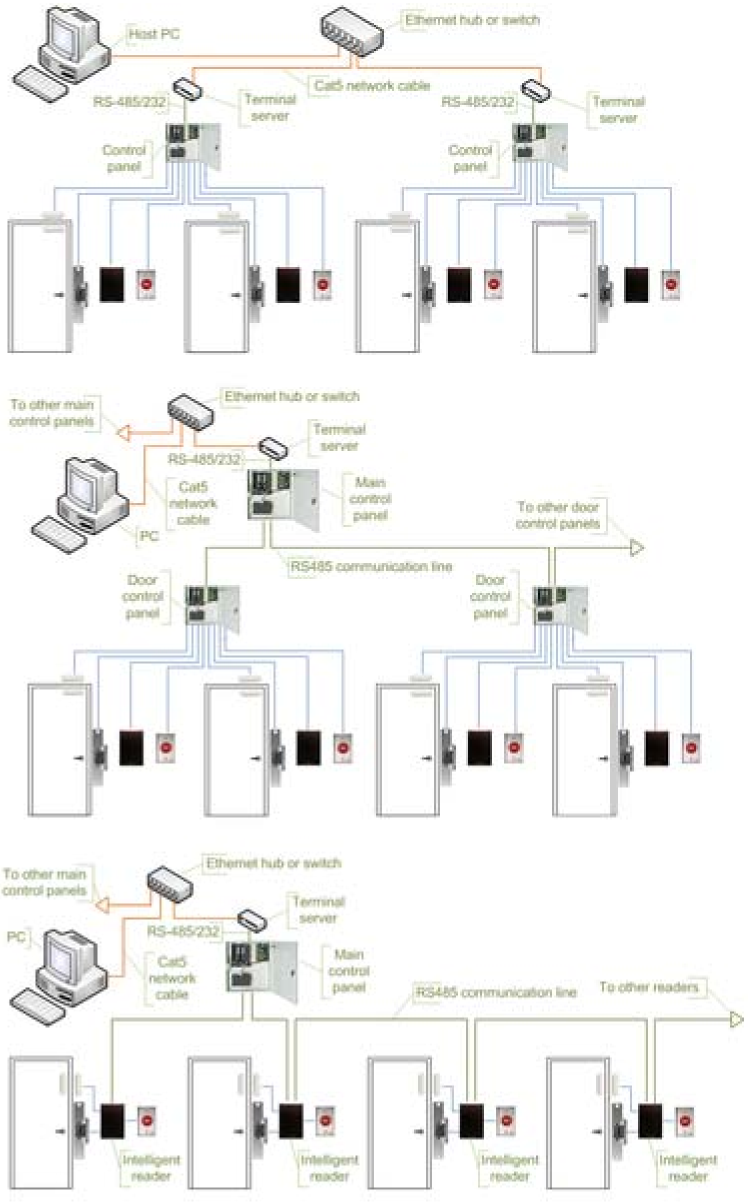
- Operation of the system is highly dependent on main controllers. In case one of the main controllers fails, events from its sub-controllers are not retrieved and functions that require interaction between sub controllers (i.e. anti-passback) stop working.

- Some models of sub-controllers (usually lower cost) have no memory and processing power to make access decisions independently. If the main controller fails, sub-controllers change to degraded mode in which doors are either completely locked or unlocked and no events are recorded. Such sub-controllers should be avoided or used only in areas that do not require high security.
- Main controllers tend to be expensive, therefore such topology is not very well suited for systems with multiple remote locations that have only a few doors.
- All other RS-485-related disadvantages listed in the first paragraph apply.



Access control system using serial main controller and intelligent readers

3. Serial main controllers & intelligent readers. All door hardware is connected directly to intelligent or semi-intelligent readers. Readers usually do not make access decisions, and forward all requests to the main controller. Only if the connection to the main controller is unavailable, the readers use their internal database to make access decisions and record events. Semi-intelligent reader that have no database and cannot function without the main controller should be used only in areas that do not require high security. Main controllers usually support from 16 to 64 readers. All advantages and disadvantages are the same as the ones listed in the second paragraph.



Access control systems using serial controllers and terminal servers

4. Serial controllers with terminal servers. In spite of the rapid development and increasing use of computer networks, access control manufacturers remained conservative and did not rush to introduce network-enabled products. When pressed for solutions with network connectivity, many chose the option requiring less efforts: addition of a terminal server, a device that converts serial data for transmission via LAN or WAN. Terminal servers manufactured by Lantronix and Tibbo Technology are popular in the security industry. Advantages:

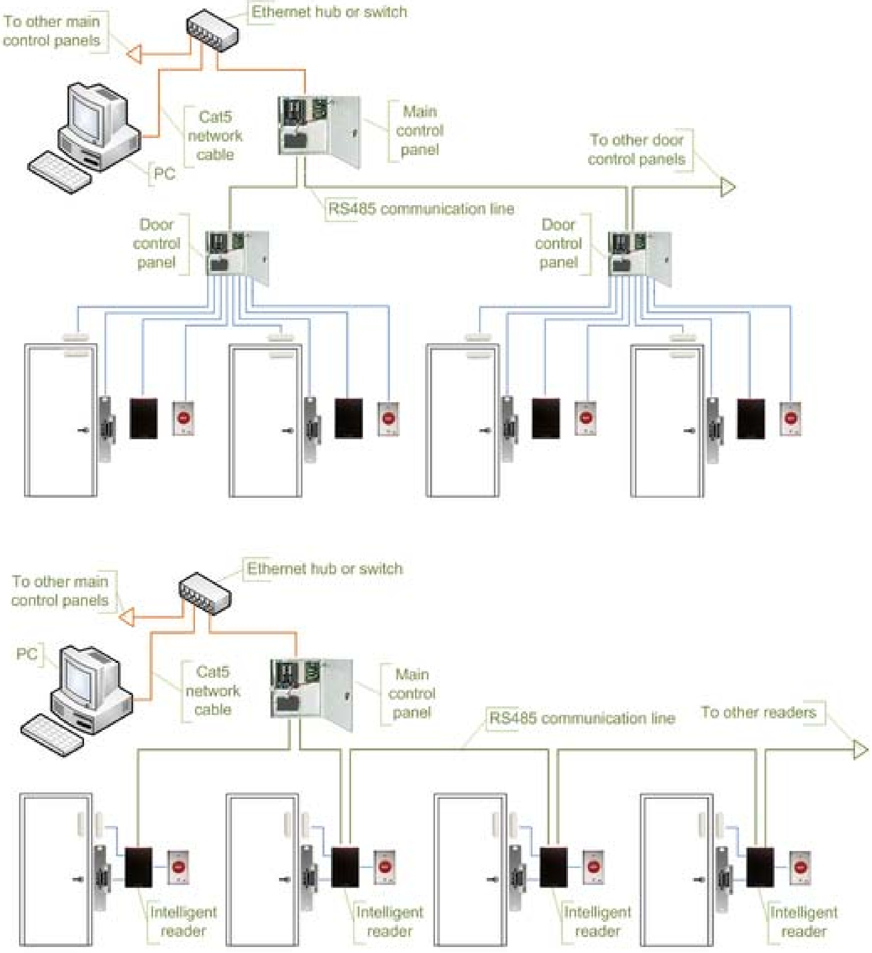
- Allows utilizing existing network infrastructure for connecting separate segments of the system.

- Provides convenient solution in cases when installation of an RS-485 line would be difficult or impossible.

Disadvantages:

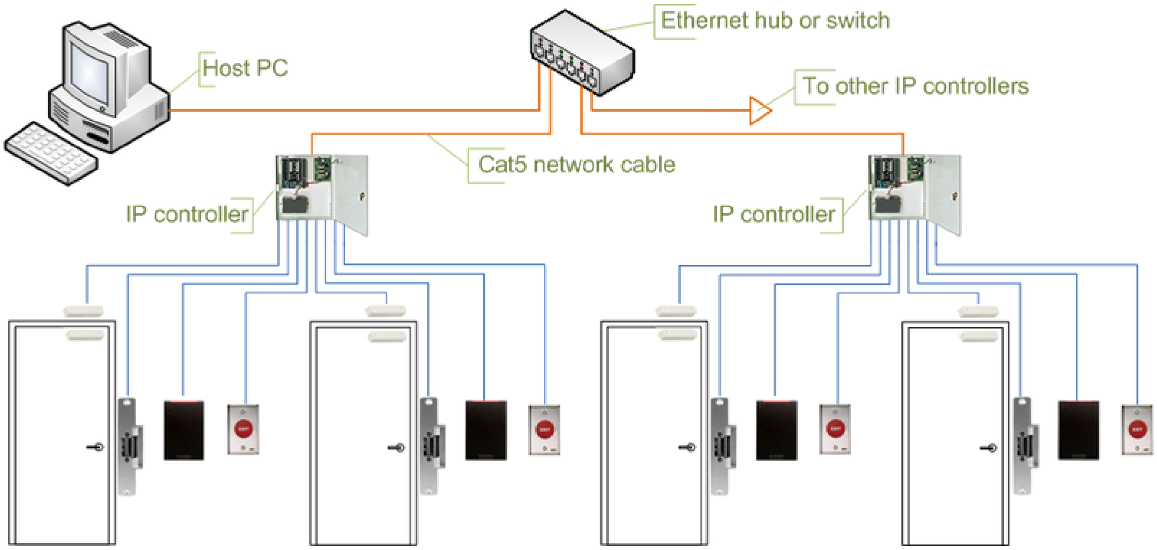
- Increases complexity of the system.
- Creates additional work for installers: usually terminal servers have to be configured independently, not through the interface of the access control software.
- Serial communication link between the controller and the terminal server acts as a bottleneck: even though the data between the host PC and the terminal server travels at the 10/100/1000Mbit/s network speed it then slows down to the serial speed of 112.5 kbit/s or less. There are also additional delays introduced in the process of conversion between serial and network data.

All RS-485-related advantages and disadvantages also apply.



Access control system using network-enabled main controllers

5. Network-enabled main controllers. The topology is nearly the same as described in the second and third paragraphs. The same advantages and disadvantages apply, but the on-board network interface offers a couple valuable improvements. Transmission of configuration and users to the main controllers is faster and may be done in parallel. This makes the system more responsive and does not interrupt normal operations. No special hardware is required in order to achieve redundant host PC setup: in case the primary host PC fails, the secondary host PC may start polling network controllers. The disadvantages introduced by terminal servers (listed in the fourth paragraph) are also eliminated.



Access control system using IP controllers

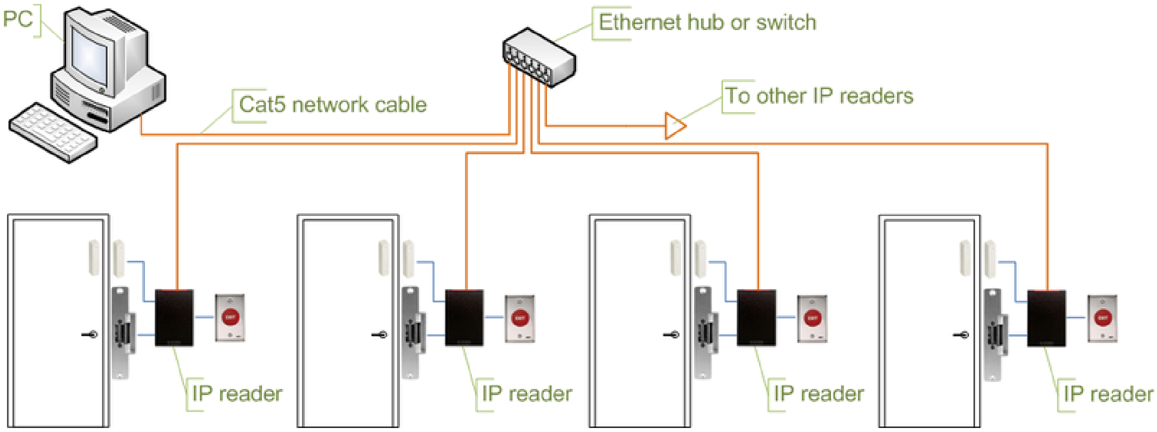
6. IP controllers. Controllers are connected to a host PC via Ethernet LAN or WAN. Advantages:

- An existing network infrastructure is fully utilized, there is no need to install new communication lines.
- There are no limitations regarding the number of controllers (32 per line in case of RS-485).
- Special RS-485 installation, termination, grounding and troubleshooting knowledge is not required.
- Communication with controllers may be done at the full network speed, which is important if transferring a lot of data (databases with thousands of users, possibly including biometric records).
- In case of an alarm controllers may initiate connection to the host PC. This ability is important in large systems because it allows to reduce network traffic caused by unnecessary polling.
- Simplifies installation of systems consisting of multiple sites separated by large distances. Basic Internet link is sufficient to establish connections to remote locations.

- Wide selection of standard network equipment is available to provide connectivity in different situations (fiber, wireless, VPN, dual path, PoE)

Disadvantages:

- The system becomes susceptible to network related problems, such as delays in case of heavy traffic and network equipment failures.
- Access controllers and workstations may become accessible to hackers if the network of the organization is not well protected. This threat may be eliminated by physically separating the access control network from the network of the organization. Also it should be noted that most IP controllers utilize either Linux platform or proprietary operating systems, which makes them more difficult to hack. Industry standard data encryption is also used.
- Maximum distance from a hub or a switch to the controller (if using a copper cable) is 100 meters (330 ft).
- Operation of the system is dependent on the host PC. In case the host PC fails, events from controllers are not retrieved and functions that required interaction between controllers (i.e. anti-passback) stop working. Some controllers, however, have peer-to-peer communication option in order to reduce dependency on the host PC.



Access control system using IP readers

7. IP readers. Readers are connected to a host PC via Ethernet LAN or WAN.
Advantages:

- Most IP readers are PoE capable. This feature makes it very easy to provide battery backed power to the entire system, including the locks and various types of detectors (if used).
- IP readers eliminate the need for controller enclosures.
- There is no wasted capacity when using IP readers (i.e. a 4-door controller would have 25% unused capacity if it was controlling only 3 doors).
- IP reader systems scale easily: there is no need to install new main or sub-controllers.

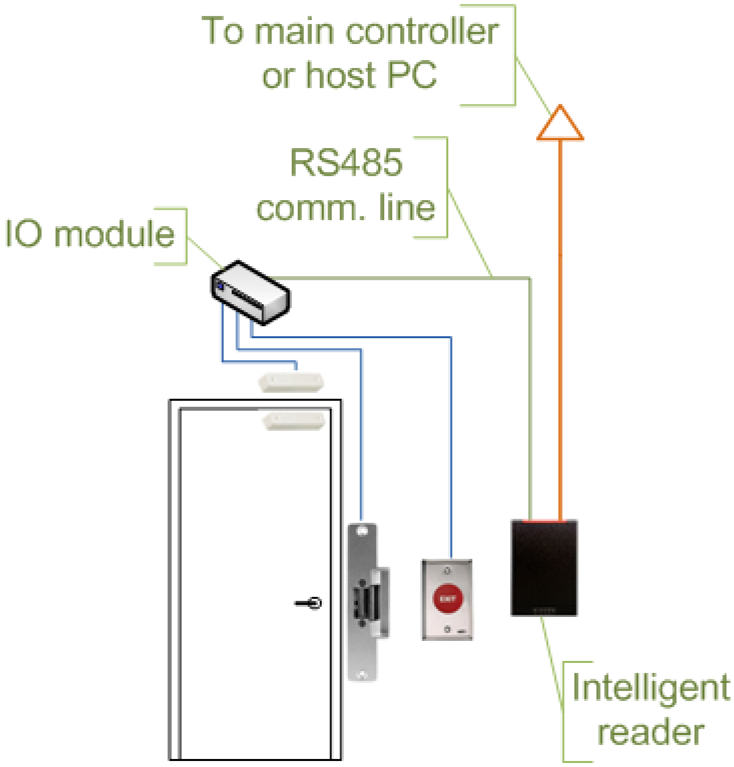
- Failure of one IP reader does not affect any other readers in the system.

Disadvantages:

- In order to be used in high-security areas IP readers require special input/output modules to eliminate the possibility of intrusion by accessing lock and/or exit button wiring. Not all IP reader manufacturers have such modules available.
- Being more sophisticated than basic readers IP readers are also more expensive and sensitive, therefore they should not be installed outdoors in areas with harsh weather conditions or high possibility of vandalism, unless specifically designed for exterior installation. A few manufacturers make such models.
- In the past, the variety of IP readers in terms of identification technologies and read range was much lower than that of the basic readers. However, with the advent of long range multi-technology readers such as those manufactured by Nedap, Sirit, and a few others, this is no longer so.

The advantages and disadvantages of IP controllers apply to the IP readers as well.

Security risks



Access control door wiring when using intelligent readers and IO module

The most common security risk of intrusion of an access control system is simply following a legitimate user through a door. Often the legitimate user will hold the door for the intruder. This risk can be minimized through security awareness training of the user population or more active means such as turnstiles. In very high security applications this risk is minimized by using a sally port, sometimes called a security vestibule or mantrap where operator intervention is required presumably to assure valid identification.

The second most common risk is from levering the door open. This is surprisingly simple and effective on most doors. The lever could be as small as a screw driver or big as a crow bar. Fully implemented access control systems include forced door monitoring alarms. These vary in effectiveness usually failing from high false positive alarms, poor database configuration, or lack of active intrusion monitoring.

Similar to levering is crashing through cheap partition walls. In shared tenant spaces the demisal wall is a vulnerability. Along the same lines is breaking sidelights.

Spoofing locking hardware is fairly simple and more elegant than levering. A strong magnet can operate the solenoid controlling bolts in electric locking hardware. Motor locks, more prevalent in Europe than in the US, are also susceptible to this attack using a donut shaped magnet. It is also possible to manipulate the power to the lock either by removing or adding current.

Access cards themselves have proven vulnerable to sophisticated attacks. Enterprising hackers have built portable readers that capture the card number from a user's proximity card. The hacker simply walks by the user, reads the card, and then presents the number to a reader securing the door. This is possible because card numbers are sent in the clear, no encryption being used.

Finally, most electric locking hardware still have mechanical keys as a failover. Mechanical key locks are vulnerable to bumping.

The need-to-know principle

The need to know principle can be enforced with user access controls and authorization procedures and its objective is to ensure that only authorized individuals gain access to information or systems necessary to undertake their duties.

Computer security

In computer security, access control includes authentication, authorization and audit. It also includes measures such as physical devices, including biometric scans and metal locks, hidden paths, digital signatures, encryption, social barriers, and monitoring by humans and automated systems.

In any access control model, the entities that can perform actions in the system are called *subjects*, and the entities representing resources to which access may need to be controlled are called *objects*. Subjects and objects should both be considered as software entities and as human users. Although some systems equate subjects with *user IDs*, so that all processes started by a user by default have the same authority, this level of control is not fine-grained enough to satisfy the Principle of least privilege, and arguably is responsible for the prevalence of malware in such systems.

In some models, for example the object-capability model, any software entity can potentially act as both a subject and object.

Access control models used by current systems tend to fall into one of two classes: those based on capabilities and those based on access control lists (ACLs). In a capability-based model, holding an unforgeable reference or *capability* to an object provides access to the object (roughly analogous to how possession of your house key grants you access to your house); access is conveyed to another party by transmitting such a capability over a secure channel. In an ACL-based model, a subject's access to an object depends on whether its identity is on a list associated with the object (roughly analogous to how a bouncer at a private party would check your ID to see if your name is on the guest list); access is conveyed by editing the list. (Different ACL systems have a variety of different conventions regarding who or what is responsible for editing the list and how it is edited.)

Both capability-based and ACL-based models have mechanisms to allow access rights to be granted to all members of a *group* of subjects (often the group is itself modeled as a subject).

Access control systems provide the essential services of *identification and authentication* (*I&A*), *authorization*, and *accountability* where:

- identification and authentication determine who can log on to a system, and the association of users with the software subjects that they are able to control as a result of logging in;
- authorization determines what a subject can do;
- accountability identifies what a subject (or all subjects associated with a user) did.

Identification and authentication (I&A)

Identification and authentication (I&A) is the process of verifying that an identity is bound to the entity that makes an assertion or claim of identity. The I&A process assumes that there was an initial validation of the identity, commonly called identity proofing. Various methods of identity proofing are available ranging from in person validation using government issued identification to anonymous methods that allow the claimant to remain anonymous, but known to the system if they return. The method used for identity proofing and validation should provide an assurance level commensurate with the intended use of the identity within the system. Subsequently, the entity asserts an identity together with an authenticator as a means for validation. The only requirements for the identifier is that it must be unique within its security domain.

Authenticators are commonly based on at least one of the following four factors:

- *Something you know*, such as a password or a personal identification number (PIN). This assumes that only the owner of the account knows the password or PIN needed to access the account.
- *Something you have*, such as a smart card or security token. This assumes that only the owner of the account has the necessary smart card or token needed to unlock the account.
- *Something you are*, such as fingerprint, voice, retina, or iris characteristics.
- *Where you are*, for example inside or outside a company firewall, or proximity of login location to a personal GPS device.

Authorization

Authorization applies to subjects. Authorization determines what a subject can do on the system.

Most modern operating systems define sets of permissions that are variations or extensions of three basic types of access:

- Read (R): The subject can
  - Read file contents
  - List directory contents
- Write (W): The subject can change the contents of a file or directory with the following tasks:
  - Add
  - Create
  - Delete
  - Rename
- Execute (X): If the file is a program, the subject can cause the program to be run. (In Unix systems, the 'execute' permission doubles as a 'traverse directory' permission when granted for a directory.)

These rights and permissions are implemented differently in systems based on *discretionary access control* (*DAC*) and *mandatory access control* (*MAC*).

Accountability

Accountability uses such system components as *audit trails* (records) and *logs* to associate a subject with its actions. The information recorded should be sufficient to map the subject to a controlling user. Audit trails and logs are important for

- Detecting security violations
- Re-creating security incidents

If no one is regularly reviewing your logs and they are not maintained in a secure and consistent manner, they may not be admissible as evidence.

Many systems can generate automated reports based on certain predefined criteria or thresholds, known as *clipping levels*. For example, a clipping level may be set to generate a report for the following:

- More than three failed logon attempts in a given period
- Any attempt to use a disabled user account

These reports help a system administrator or security administrator to more easily identify possible break-in attempts.

Access control techniques

Access control techniques are sometimes categorized as either discretionary or non-discretionary. The three most widely recognized models are Discretionary Access Control (DAC), Mandatory Access Control (MAC), and Role Based Access Control (RBAC). MAC and RBAC are both non-discretionary.

Attribute-based access control

In attribute-based access control (ABAC), access is granted not based on the rights of the subject associated with a user after authentication, but based on attributes of the user. The user has to prove so called claims about his attributes to the access control engine. An attribute-based access control policy specifies which claims need to be satisfied in order to grant access to an object. For instance the claim could be "older than 18". Any user that can prove this claim is granted access. Users can be anonymous as authentication and identification are not strictly required. One does however require means for proving claims anonymously. This can for instance be achieved using anonymous credentials or XACML (extensible access control markup language).

Discretionary access control

Discretionary access control (DAC) is an access policy determined by the owner of an object. The owner decides who is allowed to access the object and what privileges they have.

Two important concepts in DAC are

- File and data ownership: Every object in the system has an *owner*. In most DAC systems, each object's initial owner is the subject that caused it to be created. The access policy for an object is determined by its owner.
- Access rights and permissions: These are the controls that an owner can assign to other subjects for specific resources.

Access controls may be discretionary in ACL-based or capability-based access control systems. (In capability-based systems, there is usually no explicit concept of 'owner', but the creator of an object has a similar degree of control over its access policy.)

Mandatory access control

Mandatory access control (MAC) is an access policy determined by the system, not the owner. MAC is used in multilevel systems that process highly sensitive data, such as classified government and military information. A multilevel system is a single computer system that handles multiple classification levels between subjects and objects.

- Sensitivity labels: In a MAC-based system, all subjects and objects must have labels assigned to them. A subject's sensitivity label specifies its level of trust. An object's sensitivity label specifies the level of trust required for access. In order to access a given object, the subject must have a sensitivity level equal to or higher than the requested object.
- Data import and export: Controlling the import of information from other systems and export to other systems (including printers) is a critical function of MAC-based systems, which must ensure that sensitivity labels are properly maintained and implemented so that sensitive information is appropriately protected at all times.

Two methods are commonly used for applying mandatory access control:

- Rule-based (or label-based) access control: This type of control further defines specific conditions for access to a requested object. All MAC-based systems implement a simple form of rule-based access control to determine whether access should be granted or denied by matching:
  - An object's sensitivity label
  - A subject's sensitivity label
- Lattice-based access control: These can be used for complex access control decisions involving multiple objects and/or subjects. A lattice model is a mathematical structure that defines greatest lower-bound and least upper-bound values for a pair of elements, such as a subject and an object.

Few systems implement MAC; XTS-400 and SELinux are examples of systems that do. The computer system at the company in the movie *Tron* is an example of MAC in popular culture.

Role-based access control

Role-based access control (RBAC) is an access policy determined by the system, not the owner. RBAC is used in commercial applications and also in military systems, where multi-level security requirements may also exist. RBAC differs from DAC in that DAC allows users to control access to their resources, while in RBAC, access is controlled at the system level, outside of the user's control. Although RBAC is non-discretionary, it can be distinguished from MAC primarily in the way permissions are handled. MAC controls read and write permissions based on a user's clearance level and additional labels. RBAC controls collections of permissions that may include complex operations such as an e-commerce transaction, or may be as simple as read or write. A role in RBAC can be viewed as a set of permissions.

Three primary rules are defined for RBAC:

1. Role assignment: A subject can execute a transaction only if the subject has selected or been assigned a role.
2. Role authorization: A subject's active role must be authorized for the subject. With rule 1 above, this rule ensures that users can take on only roles for which they are authorized.
3. Transaction authorization: A subject can execute a transaction only if the transaction is authorized for the subject's active role. With rules 1 and 2, this rule ensures that users can execute only transactions for which they are authorized.

Additional constraints may be applied as well, and roles can be combined in a hierarchy where higher-level roles subsume permissions owned by sub-roles.

Most IT vendors offer RBAC in one or more products.

Telecommunication

In telecommunication, the term *access control* is defined in U.S. Federal Standard 1037C with the following meanings:

1. A service feature or technique used to permit or deny use of the components of a communication system.
2. A technique used to define or restrict the rights of individuals or application programs to obtain data from, or place data onto, a storage device.
3. The definition or restriction of the rights of individuals or application programs to obtain data from, or place data into, a storage device.
4. The process of limiting access to the resources of an AIS to authorized users, programs, processes, or other systems.
5. That function performed by the resource controller that allocates system resources to satisfy user requests.

This definition depends on several other technical terms from Federal Standard 1037C.

Public policy

In public policy, access control to restrict access to systems ("authorization") or to track or monitor behavior within systems ("accountability") is an implementation feature of using trusted systems for security or social control.

Chapter-6

HTTP Secure

Hypertext Transfer Protocol Secure (**HTTPS**) is a combination of the Hypertext Transfer Protocol with the SSL/TLS protocol to provide encrypted communication and secure identification of a network web server. HTTPS connections are often used for payment transactions on the World Wide Web and for sensitive transactions in corporate information systems. **HTTPS** should not be confused with Secure HTTP (S-HTTP) specified in RFC 2660.

Main idea

The main idea of HTTPS is to create a secure channel over an insecure network. This ensures reasonable protection from eavesdroppers and man-in-the-middle attacks, provided that adequate cipher suites are used and that the *server certificate is verified and trusted*.

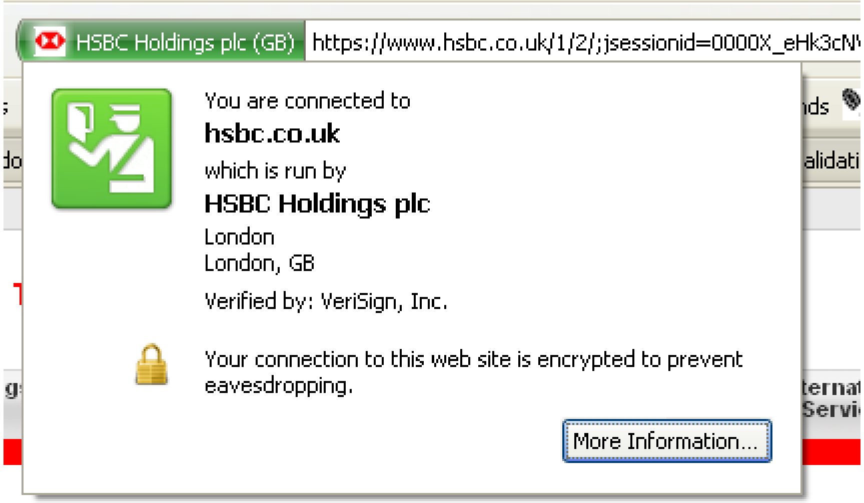
The trust inherent in HTTPS is based on major certificate authorities that come pre-installed in browser software (this is equivalent to saying "I trust certificate authority (e.g. VeriSign/Microsoft/etc.) to tell me whom I should trust"). Therefore an HTTPS connection to a website can be trusted if and only if all of the following are true:

1. The user trusts that their browser software correctly implements HTTPS with correctly pre-installed certificate authorities.
2. The user trusts the certificate authority to vouch only for legitimate websites without misleading names.
3. The website provides a valid certificate, which means it was signed by a trusted authority.
4. The certificate correctly identifies the website (e.g., when the browser visits "https://example", the received certificate is properly for "Example Inc." and not some other entity).
5. Either the intervening hops on the Internet are trustworthy, or the user trusts the protocol's encryption layer (TLS or SSL) is unbreakable by an eavesdropper.

Browser integration

Most browsers display a warning if they receive an invalid certificate. Older browsers, when connecting to a site with an invalid certificate, would present the user with a dialog box asking if they wanted to continue. Newer browsers display a warning across the entire window. Newer browsers also prominently display the site's security information in the address bar.

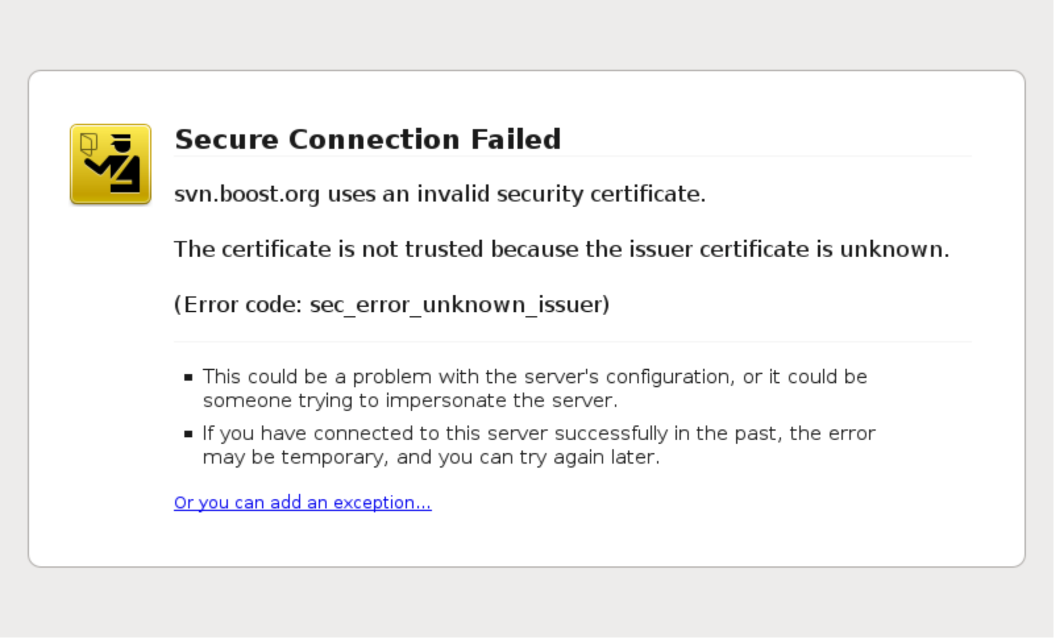
Extended validation certificates turn the address bar green in newer browsers. Most browsers also display a warning to the user when visiting a site that contains a mixture of encrypted and unencrypted content.



You are connected to
hsbc.co.uk
which is run by
HSBC Holdings plc
London
London, GB
Verified by: VeriSign, Inc.

Your connection to this web site is encrypted to prevent eavesdropping.

Many web browsers, including Firefox (shown here), use the address bar to tell the user that their connection is secure, often by coloring the background.



Secure Connection Failed

svn.boost.org uses an invalid security certificate.

The certificate is not trusted because the issuer certificate is unknown.

(Error code: sec_error_unknown_issuer)

- This could be a problem with the server's configuration, or it could be someone trying to impersonate the server.
- If you have connected to this server successfully in the past, the error may be temporary, and you can try again later.

Most web browsers alert the user when visiting sites that have invalid security certificates. This example is from Firefox.

The Electronic Frontier Foundation, opining that "[i]n an ideal world, every web request could be defaulted to HTTPS", has provided an add-on for the Firefox browser that does so for several frequently used websites.

Technical

Difference from HTTP

HTTP is unsecured and is subject to man-in-the-middle and eavesdropping attacks, which can let attackers gain access to website accounts and sensitive information. HTTPS is designed to withstand such attacks and is considered secure against such attacks (with the exception of older deprecated versions of SSL).

Server setup

To prepare a web server to accept HTTPS connections, the administrator must create a public key certificate for the web server. This certificate must be signed by a trusted certificate authority for the web browser to accept it. The authority certifies that the certificate holder is indeed the entity it claims to be. Web browsers are generally distributed with the signing certificates of major certificate authorities so that they can verify certificates signed by them.

Acquiring certificates

Authoritatively signed certificates may be free or cost between US$13 and $1,500 per year.

Organizations may also run their own certificate authority, particularly if they are responsible for setting up browsers to access their own sites (for example, sites on a company intranet, or major universities). They can easily add copies of their own signing certificate to the trusted certificates distributed with the browser.

There also exists a peer-to-peer certificate authority, CACert.

Use as access control

The system can also be used for client authentication in order to limit access to a web server to authorized users. To do this, the site administrator typically creates a certificate for each user, a certificate that is loaded into his/her browser. Normally, that contains the name and e-mail address of the authorized user and is automatically checked by the server on each reconnect to verify the user's identity, potentially without even entering a password.

In case of compromised private key

A certificate may be revoked before it expires, for example because the secrecy of the private key has been compromised. Newer versions of popular browsers such as Google Chrome, Firefox, Opera, and Internet Explorer on Windows Vista implement the Online Certificate Status Protocol (OCSP) to verify that this is not the case. The browser sends the certificate's serial number to the certificate authority or its delegate via OCSP and the authority responds, telling the browser whether or not the certificate is still valid.

Limitations

SSL comes in two options, simple and mutual.

The mutual flavor is more secure but requires the user to install a personal certificate in their browser in order to authenticate themselves.

Whatever strategy is used (simple or mutual), the level of protection strongly depends on the correctness of the implementation of the web browser and the server software and the actual cryptographic algorithms supported.

SSL doesn't prevent the entire site from being indexed using a web crawler, and in some cases the URI of the encrypted resource can be inferred by knowing only the intercepted request/response size. This allows an attacker to have access to the plaintext (the publicly-available static content), and the encrypted text (the encrypted version of the static content), permitting a cryptographic attack.

Because SSL operates below HTTP and has no knowledge of higher-level protocols, SSL servers can only strictly present one certificate for a particular IP/port combination. This means that, in most cases, it is not feasible to use name-based virtual hosting with HTTPS. A solution called Server Name Indication (SNI) exists, which sends the hostname to the server before encrypting the connection, although many older browsers don't support this extension. Support for SNI is available since Firefox 2, Opera 8, Safari 2.1, Google Chrome 6, and Internet Explorer 7 on Windows Vista.

If parental controls are enabled on Mac OS X, HTTPS sites must be explicitly allowed using the Always Allow list.

From an architectural point of view:

1. An SSL/TLS connection is managed by the first front machine that initiates the SSL connection. If, for any reasons (routing, traffic optimization, etc.), this front machine is not the application server and it has to decipher data, solutions have to be found to propagate user authentication informations or certificate to the application server, which needs to know who is going to be connected.
2. For SSL with mutual authentication, the SSL/TLS session is managed by the first server that initiates the connection. In situations where encryption has to be

propagated along chained servers, session timeOut management becomes extremely tricky to implement.

3. With mutual SSL/TLS, security is maximal, but on the client-side, there is no way to properly end the SSL connection and disconnect the user except by waiting for the SSL server session to expire or closing all related client applications.
4. For performance reasons, static content that is not specific to the user or transaction, and thus not private, is usually delivered through a non-crypted front server or separate server instance with no SSL. As a consequence, this content is usually not protected. Many browsers warn the user when a page has mixed encrypted and non-encrypted resources.

History

Netscape Communications created HTTPS in 1994 for its Netscape Navigator web browser. Originally, HTTPS was used with SSL encryption. As SSL evolved into Transport Layer Security (TLS), the current version of HTTPS was formally specified by RFC 2818 in May 2000.

Chapter-7

Secure Communications Interoperability Protocol and Secure Electronic Transaction

Secure Communications Interoperability Protocol

SCIP is a multinational standard for secure voice and data communication. The acronym stands for **Secure Communications Interoperability Protocol**. SCIP derived from the US Government **FNBDT** (**Future Narrowband Digital Terminal**) project after the US offered to share details of FNBDT with a number of other nations in 2003. SCIP supports a number of different modes, including national and multinational modes which employ different cryptography. Many nations and industries are actively developing SCIP devices to support the multinational and national modes of SCIP.

SCIP has to operate over the wide variety of communications systems, including commercial land line telephone, military radios, communication satellites, Voice over IP and the several different cellular telephone standards. Therefore it was designed to make no assumptions about the underlying channel other than a minimum bandwidth of 2400 Hz. It is similar to a dial-up modem in that once a connection is made, two SCIP phones first negotiate the parameters they need and then communicate in the best way possible.

US SCIP [FNBDT] systems have been in use since 2001, beginning with the CONDOR secure cell phone. The standard is designed to cover wideband as well as narrowband voice and data security.

SCIP was designed by the Department of Defense Digital Voice Processor Consortium (DDVPC) in cooperation with the U.S. National Security Agency and is intended to solve problems with earlier NSA encryption systems for voice, including STU-III and STE which made assumptions about the underlying communication systems that prevented interoperability with more modern wireless systems. STE sets can be upgraded to work with SCIP, but STU-III cannot. This has led to some resistance since various government agencies already own over 350,000 STU-III telephones at a cost of several thousand dollars each.

There are several components to the SCIP standard: key management, voice compression, encryption and a signalling plan.

Key Management (120)

To set up a secure call, a new Traffic Encryption Key (**TEK**) must be negotiated. For Type 1 security (classified calls), the SCIP signalling plan uses an enhanced FIREFLY messaging system for key exchange. FIREFLY is an NSA key management system based on public key cryptography. At least one commercial grade implementation uses Diffie-Hellman key exchange.

STEs use security tokens to limit use of the secure voice capability to authorized users while other SCIP devices only require a PIN code, 7 digits for Type 1 security, 4 digits for unclassified.

Voice compression using Voice Coders (vocoders)

SCIP can work with a variety of vocoders, but the standard requires, as a minimum, support for the Mixed Excitation Linear Prediction coder known as (**MELP**), an enhanced MELP algorithm known as MELPe, with additional preprocessing, analyzer and synthesizer capabilities for improved intelligibility and noise robustness. The old MELP and the new MELPe are interoperable and both operate at 2400 bit/s, sending a 54 bit data frame every 22.5 milliseconds but the MELPe has optional additional rates of 1200 bit/s and 600 bit/s.

2400 bit/s MELPe is the only mandatory voice coder required for SCIP. Other voice coders can be supported in terminals. These can be used if all terminals involved in the call support the same coder (agreed during the negotiation stage of call setup) and the network can support the required throughput. G.729D is the most widely supported non-mandatory voice coder in SCIP terminals as it offers a good compromise between higher voice quality without dramatically increasing the required throughput.

Encryption (SCIP 23x)

The security used by the multinational and national modes of SCIP is defined by the SCIP 23x family of documents. SCIP 231 defines AES based cryptography which can be used multinationally. SCIP 232 defines an alternate multinational cryptographic solution. Several nations have defined, or are defining, their own national security modes for SCIP.

US National Mode (SCIP 230)

SCIP 230 defines the cryptography of the US national mode of SCIP. The rest of this section refers to SCIP 230. For security, SCIP uses a block cipher operating in counter mode. A new Traffic Encryption Key (**TEK**) is negotiated for each call. The block cipher is fed a 64-bit state vector (**SV**) as input. If the cipher's block size is longer than 64 bits, a fixed filler is added. The output from the block cipher is xored with the MELP data frames to create the cipher text that is then transmitted.

The low-order two bits of the state vector are reserved for applications where the data frame is longer than the block cipher output. The next 42 bits are the counter. Four bits are used to represent the transmission mode. This allows more than one mode, e.g. voice and data, to operate at the same time with the same TEK. The high-order 16 bits are a sender ID. This allows multiple senders on a single channel to all use the same TEK. Note that since overall SCIP encryption is effectively a stream cipher, it is essential that the same state vector value never be used twice for a given TEK. At MELP data rates, a 42-bit counter allows a call over three thousand years long before the encryption repeats.

For Type 1 security, SCIP uses BATON, a 128-bit block design. With this or other 128-bit ciphers, such as AES, SCIP specifies that two data frames are encrypted with each cipher output bloc, the first beginning at bit 1, the second at bit 57 (i.e. the next byte boundary). At least one commercial grade implementation uses the Triple DES cipher.

Signalling plan (210)

The SCIP signalling plan in common to all national and multinational modes of SCIP. SCIP has two mandatory types of transmission. The mandatory data service uses an ARQ protocol with forward error correction (FEC) to ensure reliable transmission. The receiving station acknowledges accurate receipt of data blocks and can ask for a block to be re-transmitted, if necessary. For voice, SCIP simply sends a stream of voice data frames (typically MELPe frames, but possibly G.729D or another codec if that has been negotiated between the terminals). To save power on voice calls, SCIP stops sending if there is no speech input. A synchronization block is sent roughly twice a second in place of a data frame. The low order 14 bits of the encryption counter are sent with every sync block. The 14 bits are enough to cover a fade out of more than six minutes. Part of the rest of the state vector are sent as well so that with receipt of three sync blocks, the entire state vector is recovered. This handles longer fades and allows a station with the proper TEK to join a multi station net and be synchronized within 1.5 seconds.

As of March 2011 the SCIP-210 signalling standard is publicly available from the IAD website.

Availability

SCIP specifications are not widely diffused or easily accessible. This makes the protocol for government use rather "opaque" outside governments or defense industries. No public implementation of the security and transport protocols are available, precluding its security from being publicly verified.

Secure Electronic Transaction

Secure Electronic Transaction (SET) was a standard protocol for securing credit card transactions over insecure networks, specifically, the Internet. SET was not itself a payment system, but rather a set of security protocols and formats that enable users to employ the existing credit card payment infrastructure on an open network in a secure fashion. However, it failed to gain traction. VISA now promotes the 3-D Secure scheme.

History and development

SET was developed by **SETco**, led by VISA and MasterCard (and involving other companies such as GTE, IBM, Microsoft, Netscape, RSA and VeriSign) starting in 1996. SET was based on X.509 certificates with several extensions. The first version was finalised in May 1997 and a pilot test was announced in July 1998.

SET allowed parties to cryptographically identify themselves to each other and exchange information securely. SET used a blinding algorithm that, in effect, would have let merchants substitute a certificate for a user's credit-card number. If SET were used, the merchant itself would never have had to know the credit-card numbers being sent from the buyer, which would have provided verified good payment but protected customers and credit companies from fraud.

SET was intended to become the de facto standard of payment method on the Internet between the merchants, the buyers, and the credit-card companies. Despite heavy publicity, it failed to win market share. Reasons for this include:

- Network effect - need to install client software (an e-wallet).
- Cost and complexity for merchants to offer support and comparatively low cost and simplicity of the existing SSL based alternative.
- Client-side certificate distribution logistics.

Key features

To meet the business requirements, SET incorporates the following features:

- Confidentiality of information
- Integrity of data
- Cardholder account authentication
- Merchant authentication

Participants

A SET system includes the following participants:

- Cardholder

- Merchant
- Issuer
- Acquirer
- Payment gateway
- Certification authority

Transaction

The sequence of events required for a transaction are as follows:

1. The customer obtains a credit card account with a bank that supports electronic payment and SET
2. The customer receives a X.509v3 digital certificate signed by the bank.
3. Merchants have their own certificates
4. The customer places an order
5. The merchant sends a copy of its certificate so that the customer can verify that it's a valid store
6. The order and payment are sent
7. The merchant requests payment authorization
8. The merchant confirms the order
9. The merchant ships the goods or provides the service to the customer
10. The merchant requests payment

Dual signature

An important innovation introduced in SET is the **dual signature**. The purpose of the dual signature is the same as the standard electronic signature: to guarantee the authentication and integrity of data. It links two messages that are intended for two different recipients. In this case, the customer wants to send the order information (OI) to the merchant and the payment information (PI) to the bank. The merchant does not need to know the customer's credit card number, and the bank does not need to know the details of the customer's order. The link is needed so that the customer can prove that the payment is intended for this order.

The message digest (MD) of the OI and the PI are independently calculated by the customer. The dual signature is the encrypted MD (with the customer's secret key) of the concatenated MD's of PI and OI. The dual signature is sent to both the merchant and the bank. The protocol arranges for the merchant to see the MD of the PI without seeing the PI itself, and the bank sees the MD of the OI but not the OI itself. The dual signature can be verified using the MD of the OI or PI. It doesn't require the OI or PI itself. Its MD does not reveal the content of the OI or PI, and thus privacy is preserved.

Chapter-8

Data Breach, Secure Telephone and Secure Voice

Data breach

A **data breach** is the intentional or unintentional release of secure information to an untrusted environment. Other terms for this phenomenon include **unintentional information disclosure**, **data leak** and also **data spill**. Incidents range from concerted attack by black hats with the backing of organized crime or national governments to careless disposal of used computer equipment or data storage media. Definition "A data breach is a security incident in which sensitive, protected or confidential data is copied, transmitted, viewed, stolen or used by an individual unauthorized to do so." Data breaches may involve financial information such as credit card or bank details, personal health information (PHI), Personally identifiable information (PII), trade secrets of corporations or intellectual property. According to the nonprofit consumer organization Privacy Rights Clearinghouse, a total of 227,052,199 individual records containing sensitive personal information were involved in security breaches in the United States between January 2005 and May 2008, excluding incidents where sensitive data was apparently not actually exposed.

Definition

This may include incidents such as theft or loss of digital media such as computer tapes, hard drives, or laptop computers containing such media upon which such information is stored unencrypted, posting such information on the world wide web or on a computer otherwise accessible from the Internet without proper information security precautions, transfer of such information to a system which is not completely open but is not appropriately or formally accredited for security at the approved level, such as unencrypted e-mail, or transfer of such information to the information systems of a possibly hostile agency, such as a competing corporation or a foreign nation, where it may be exposed to more intensive decryption techniques.

Trusted environment

The notion of a trusted environment is somewhat fluid. The departure of a trusted staff member with access to sensitive information can become a data breach if the staff member retains access to the data subsequent to termination of the trust relationship. In distributed systems, this can also occur with a breakdown in a web of trust.

Data privacy

Most such incidents publicized in the media involve private information on individuals, *i.e.* social security numbers, *etc.*. Loss of corporate information such as trade secrets, sensitive corporate information, details of contracts, *etc.* or of government information is frequently unreported, as there is no compelling reason to do so in the absence of potential damage to private citizens, and the publicity around such an event may be more damaging than the loss of the data itself.

Consequences

Although such incidents pose the risk of identity theft or other serious consequences, in most cases there is no lasting damage; either the breach in security is remedied before the information is accessed by unscrupulous people, or the thief is only interested in the hardware stolen, not the data it contains. Nevertheless, when such incidents become publicly known, it is customary for the offending party to attempt to mitigate damages by providing to the victims subscription to a credit reporting agency, for instance.

Major incidents

Well known incidents include:

2009

- In December 2009 a RockYou! password database was breached containing 32 million user names and plaintext passwords, further compromising the use of weak passwords for any purpose.
- In January 2009 Heartland Payment Systems announced that it had been "the victim of a security breach within its processing system", possibly part of a "global cyber fraud operation". The intrusion has been called the largest criminal breach of card data ever, with estimates of up to 100 million cards from more than 650 financial services companies compromised.

2008

- In January 2008, GE Money, a division of General Electric, discloses that a magnetic tape containing 150,000 social security numbers and in-store credit card information from 650,000 retail customers is known to be missing from an Iron

Mountain Incorporated storage facility. J.C. Penney is among 230 retailers affected.
- Horizon Blue Cross and Blue Shield of New Jersey, January, 300,000 members
- Lifeblood, February, 321,000 blood donors
- British National Party membership list leak,

2007

- The 2007 loss of Ohio and Connecticut state data by Accenture
- TJ Maxx, data for 45 million credit and debit accounts

- 2007 UK child benefit data scandal
- CGI Group, August, 283,000 retirees from New York City
- The Gap, September, 800,000 job applicants
- Memorial Blood Center, December, 268,000 blood donors
- Davidson County Election Commission, December, 337,000 voters

2006

- AOL search data scandal (sometimes referred to as a "Data *Valdez*",, due to its size)
- Department of Veterans Affairs, May, 28,600,000 veterans, reserves, and active duty military personnel ,
- Ernst & Young, May, 234,000 customers of Hotels.com (after a similar loss of data on 38,000 employees of Ernst & Young clients in February)
- Boeing, December, 382,000 employees (after similar losses of data on 3,600 employees in April and 161,000 employees in November, 2005)

2005

- Ameriprise Financial, stolen laptop, December 24, 260,000 customer records

Secure telephone

A Secure Terminal Equipment desk set. Note slot in front for Fortezza PC Card.

A **secure telephone** is a telephone that provides voice security in the form of end-to-end encryption for the telephone call, and in some cases also the mutual authentication of the call parties, protecting them against a man-in-the-middle attack. Concerns about massive growth of telephone tapping incidents lead to growing demand for secure telephones.

The practical availability of secure telephones is restricted by several factors; notably politics, export issues, incompatibility between different products (the devices on each side of the call have to talk the same protocol), and high (though recently decreasing) price of the devices.

Well known products

The best-known product on the US government market is the STU-III family. However, this system has now been replaced by the Secure Terminal Equipment (STE) and SCIP standards which defines specifications for the design of equipment to secure both data and voice. The SCIP standard was developed by the NSA and the US DOD to derive more interoperability between secure communication equipment. A new family of standard secure phones has been created by based on Philip Zimmermann's VoIP encryption standard ZRTP.

VoIP vs direct connection phones

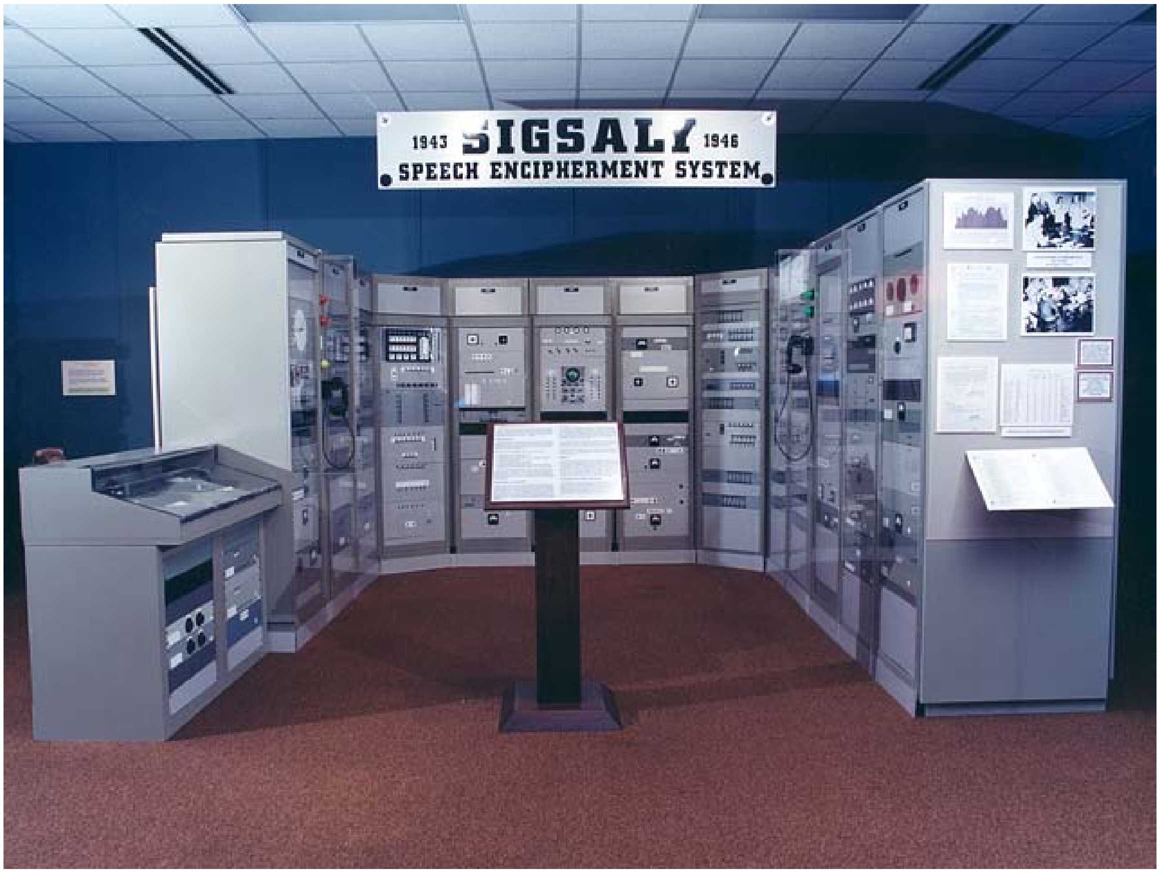
As the popularity of VoIP grows, secure telephony is becoming more of commonplace and less the lonely domain of spies and civil libertarians.

Many major hardware and software providers offer it as a standard feature. What used to only be available at high expense and to a limited number of people is now freely available.

Other examples include the Gizmo5 and Twinkle. Both of the former work with offerings from the founder of PGP, Phil Zimmermann, and his VoIP secure protocol, ZRTP. ZRTP is implemented in Ripcord Networks product SecurePC with up to NSA Suite B compliant Elliptic Curve math libraries. ZRTP is also being made available for mobile GSM CSD as a new standard for non-VoIP secure calls.

There are several manufacturers of hardware analog telephony adapters such as Sipura/linksys and Snom which offer easy to use secure options.

Historically significant products



SIGSALY exhibit at the National Cryptologic Museum

Scramblers were used to secure voice traffic during World War II, but were often intercepted and decoded due to scrambling's inherent insecurity. The first true secure telephone was SIGSALY, a massive device that weighed over 50 tons. NSA, formed after World War II, developed a series of secure telephones, including the STU I, STU II and STU-III, as well as voice encryption devices for military telephones.

Other products of historical significance are PGPfone and Nautilus (designed as a non-backdoored alternative to Clipper), and now officially discontinued (but continuing living on SourceForge) *SpeakFreely*, and the security VoIP protocol wrapper Zfone developed by the creator of PGP.

Scrambling, generally using a form of voice inversion, was available from electronic hobbyist kit suppliers and is common on FRS radios. Analog scrambling products exist to this day because some telecommunications circuits, like HF links and telephone lines in the developing world—are of very low quality.

Secure voice

Secure voice (alternatively **secure speech** or **ciphony**) is a term in cryptography for the encryption of voice communication over a range of communication types such as radio, telephone or IP.

History

The implementation of voice encryption dates back to World War II when secure communication was paramount to the US armed forces. During that time, noise was simply added to a voice signal to prevent enemies from listening to the conversations. Noise was added by playing a record of noise in synch with the voice signal and when the voice signal reached the receiver, the noise signal was subtracted out, leaving the original voice signal. In order to subtract out the noise, the receiver need to have the exact same noise signal and the noise records were only made in pairs; one for the transmitter and one for the receiver. Having only two copies of records made it impossible for the wrong receiver to decrypt the signal. To implement the system, the army contracted Bell Laboratories and they developed a system called SIGSALY. With SIGSALY, ten channels were used to sample the voice frequency spectrum from 250 Hz to 3 kHz and two channels were allocated to sample voice pitch and background hiss. In the time of SIGSALY, the transistor had not been developed and the digital sampling was done by circuits using the model 2051 Thyratron vacuum tube. Each SIGSALY terminal used 40 racks of equipment weighing 55 tons and filled a large room. This equipment included radio transmitters and receivers and large phonograph turntables. The voice was keyed to two 16-inch vinyl phonograph records that contained a Frequency Shift Keying (FSK) audio tone. The records were played on large precise turntables in synch with the voice transmission.

From the introduction of voice encryption to today, encryption techniques have evolved drastically. Digital technology has effectively replaced old analog methods of voice encryption and by using complex algorithms, voice encryption has become much more secure and efficient. One relatively modern voice encryption method is Sub-band coding. With Sub-band Coding, the voice signal is split into multiple frequency bands, using multiple bandpass filters that cover specific frequency ranges of interest. The output signals from the bandpass filters are then lowpass translated to reduce the bandwidth, which reduces the sampling rate. The lowpass signals are then quantized and encoded using special techniques like, Pulse Code Modulation (PCM). After the encoding stage, the signals are multiplexed and sent out along the communication network. When the signal reaches the receiver, the inverse operations are applied to the signal to get it back to its original state. Motorola developed a voice encryption system called Digital Voice Protection (DVP) as part of their first generation of voice encryption techniques. "DVP uses a self-synchronizing encryption technique known as cipher feedback (CFB). The basic DVP algorithm is capable of 2.36 x 10^{21} different "keys" based on a key length of 32 bits." The extremely high amount of possible keys associated with the early DVP algorithm, makes the algorithm very robust and gives the user a high level of security. As with any voice encryption system, the encryption key is required to decrypt the signal with a special decryption algorithm.

Analog Secure Voice technologies

One does not necessarily need digital secure voice to achieve security, as the Australian CODAN analog system (originally designed for HF but used on VHF and UHF) has proven that digital compression and encryption methods are not always required to achieve voice security. Although CODAN is by no means original or unique technology or a unique product, it has achieved recognition in the security market that exclusively digital methods aren't always needed. Voice inversion methods were commonplace in the 20th century. Few analog voice offerings exist due to the rise of exclusively digital solutions to the voice security problem.

Digital

A digital secure voice usually includes two components, a digitizer to convert between speech and digital signals and an encryption system to provide confidentiality. What makes ciphony difficult in practice is a need to send the encrypted signal over the same voiceband communication circuits used to transmit unencrypted voice, e.g. analog telephone lines or mobile radios.

This has led to the use of Voice Coders (vocoders) to achieve tight bandwidth compression of the speech signals. NSA's STU-III, KY-57 and SCIP are examples of systems that operate over existing voice circuits. The STE system, by contrast, requires wide bandwidth ISDN lines for its normal mode of operation. For encrypting GSM and VoIP, which are digital anyway, the standard protocol ZRTP could be used as an end-to-end encryption technology.

Secure voice's robustness greatly benefits from having the voice data compressed into very low bit-rates by special component called speech coding, voice compression or voice coder (also known as vocoder). The old secure voice compression standards include (CVSD, CELP, LPC-10e and MELP, where the latest standard is the state of the art MELPe algorithm.

Digital Methods using Voice Compression: MELP or MELPe

The MELPe or enhanced-MELP (Mixed Excitation Linear Prediction) is a United States Department of Defense speech coding standard used mainly in military applications and satellite communications, secure voice, and secure radio devices. Its development was led and supported by NSA, and NATO. The US government's MELPe secure voice standard is also known as MIL-STD-3005, and the NATO's MELPe secure voice standard is also known as STANAG-4591.

The 2400 bit/s MELP was created by Texas Instruments, and first standardized in 1997 and was known as MIL-STD-3005. Between 1998 and 2001, a new MELP-based vocoder was created at half the rate (i.e. 1200 bit/s) and substantial enhancements were added to the MIL-STD-3005 by SignalCom (later acquired by Microsoft) and AT&T, which included (a) additional new vocoder at half the rate (i.e. 1200 bit/s), (b) substantially improved encoding (analysis), (c) substantially improved decoding (synthesis), (d) Noise-Preprocessing for removing background noise, (e) transcoding between the 2400 bit/s and 1200 bit/s bitstreams. This fairly significant development was aimed to create a new coder at half the rate and have it interoperable with the old MELP standard.

This enhanced-MELP (also known as MELPe) was adopted as the new MIL-STD-3005 in 2001 in form of annexes and supplements made to the original MIL-STD-3005. The significant breakthrough of the 1200 bit/s MELPe enables the same quality as the old 2400 bit/s MELP's at half the rate!

One of the greatest advantages of the new 2400 bit/s MELPe is that it shares the same bit format as MELP, and hence can interoperate with legacy MELP systems, but would deliver better quality at both ends. MELPe provides much better quality than all older military standards, especially in noisy environments such as battlefield and vehicles and aircraft.

In 2002, the US DoD MELPe was adopted also as NATO standard, known as STANAG-4591. As part of NATO testing for new NATO standard, MELPe was tested against other candidates such as France's HSX (Harmonic Stochastic eXcitation) and Turkey's SB-LPC (Split-Band Linear Predictive Coding), as well as the old secure voice standards such as FS1015 LPC-10e (2.4 kbit/s), FS1016 CELP (4.8 kbit/s) and CVSD (16 kbit/s). Subsequently, the MELPe won also the NATO competition, surpassing the quality of all other candidates as well as the quality of all old secure voice standards (CVSD, CELP and LPC-10e).

The NATO competition concluded that MELPe substantially improved performance (in terms of speech quality, intelligibility, and noise immunity), while reducing throughput requirements. The NATO testing also included interoperability tests, used over 200 hours of speech data, and was conducted by 3 test laboratories world wide.

In 2005, a new 600 bit/s rate MELPe vocoder was added to the NATO standard STANAG-4591 by Thales (France), and there are more advanced efforts to lower the bitrates to 300 bit/s and even 150 bit/s.

Chapter-9

Transport Layer Security

Transport Layer Security (**TLS**) and its predecessor, **Secure Sockets Layer** (**SSL**), are cryptographic protocols that provide communications security over the Internet. TLS and SSL encrypt the segments of network connections above the Transport Layer, using symmetric cryptography for privacy and a keyed message authentication code for message reliability.

Several versions of the protocols are in widespread use in applications such as web browsing, electronic mail, Internet faxing, instant messaging and voice-over-IP (VoIP).

TLS is an IETF standards track protocol, last updated in RFC 5246 and is based on the earlier SSL specifications developed by Netscape Corporation.

Description

The TLS protocol allows client/server applications to communicate across a network in a way designed to prevent eavesdropping and tampering.

A TLS client and server negotiate a stateful connection by using a handshaking procedure. During this handshake, the client and server agree on various parameters used to establish the connection's security.

- The handshake begins when a client connects to a TLS-enabled server requesting a secure connection and presents a list of supported CipherSuites (ciphers and hash functions).
- From this list, the server picks the strongest cipher and hash function that it also supports and notifies the client of the decision.
- The server sends back its identification in the form of a digital certificate. The certificate usually contains the server name, the trusted certificate authority (CA) and the server's public encryption key.
- The client may contact the server that issued the certificate (the trusted CA as above) and confirm that the certificate is valid before proceeding.
- In order to generate the session keys used for the secure connection, the client encrypts a random number with the server's public key and sends the result to the server. Only the server should be able to decrypt it, with its private key.

- From the random number, both parties generate key material for encryption and decryption.

This concludes the handshake and begins the secured connection, which is encrypted and decrypted with the key material until the connection closes.

If any one of the above steps fails, the TLS handshake fails and the connection is not created.

History and development

Secure Network Programming API

Early research efforts toward transport layer security included the **Secure Network Programming** (SNP) application programming interface (API), which in 1993 explored the approach of having a secure transport layer API closely resembling Berkeley sockets, to facilitate retrofitting preexisting network applications with security measures.

SSL 1.0, 2.0 and 3.0

The SSL protocol was originally developed by Netscape. Version 1.0 was never publicly released; version 2.0 was released in February 1995 but "contained a number of security flaws which ultimately led to the design of SSL version 3.0" (Rescorla 2001). SSL version 3.0 was released in 1996.

TLS 1.0 (SSL 3.1)

TLS 1.0 was first defined in RFC 2246 in January 1999 as an upgrade to SSL Version 3.0. As stated in the RFC, "the differences between this protocol and SSL 3.0 are not dramatic, but they are significant enough that TLS 1.0 and SSL 3.0 do not interoperate." TLS 1.0 does include a means by which a TLS implementation can downgrade the connection to SSL 3.0.

TLS 1.1 (SSL 3.2)

TLS 1.1 was defined in RFC 4346 in April 2006. It is an update from TLS version 1.0. Significant differences in this version include:

- Added protection against Cipher block chaining (CBC) attacks.
    - The implicit Initialization Vector (IV) was replaced with an explicit IV.
    - Change in handling of padding errors.
- Support for IANA registration of parameters.

TLS 1.2 (SSL 3.3)

TLS 1.2 was defined in RFC 5246 in August 2008. It is based on the earlier TLS 1.1 specification. Major differences include:

- The MD5-SHA-1 combination in the pseudorandom function (PRF) was replaced with SHA-256, with an option to use cipher-suite specified PRFs.
- The MD5-SHA-1 combination in the Finished message hash was replaced with SHA-256, with an option to use cipher-suite specific hash algorithms.
- The MD5-SHA-1 combination in the digitally-signed element was replaced with a single hash negotiated during handshake, defaults to SHA-1.
- Enhancement in the client's and server's ability to specify which hash and signature algorithms they will accept.
- Expansion of support for authenticated encryption ciphers, used mainly for Galois/Counter Mode (GCM) and CCM mode of Advanced Encryption Standard encryption.
- TLS Extensions definition and Advanced Encryption Standard CipherSuites were added.

Applications

In applications design, TLS is usually implemented on top of any of the Transport Layer protocols, encapsulating the application-specific protocols such as HTTP, FTP, SMTP, NNTP and XMPP. Historically it has been used primarily with reliable transport protocols such as the Transmission Control Protocol (TCP). However, it has also been implemented with datagram-oriented transport protocols, such as the User Datagram Protocol (UDP) and the Datagram Congestion Control Protocol (DCCP), usage which has been standardized independently using the term Datagram Transport Layer Security (DTLS).

A prominent use of TLS is for securing World Wide Web traffic carried by HTTP to form HTTPS. Notable applications are electronic commerce and asset management. Increasingly, the Simple Mail Transfer Protocol (SMTP) is also protected by TLS. These applications use public key certificates to verify the identity of endpoints.

TLS can also be used to tunnel an entire network stack to create a VPN, as is the case with OpenVPN. Many vendors now marry TLS's encryption and authentication capabilities with authorization. There has also been substantial development since the late 1990s in creating client technology outside of the browser to enable support for client/server applications. When compared against traditional IPsec VPN technologies, TLS has some inherent advantages in firewall and NAT traversal that make it easier to administer for large remote-access populations.

TLS is also a standard method to protect Session Initiation Protocol (SIP) application signalling. TLS can be used to provide authentication and encryption of the SIP signalling associated with VoIP and other SIP-based applications.

Security

TLS has a variety of security measures:

- Protection against a downgrade of the protocol to a previous (less secure) version or a weaker cipher suite.
- Numbering subsequent Application records with a sequence number and using this sequence number in the message authentication codes (MACs).
- Using a message digest enhanced with a key (so only a key-holder can check the MAC). The HMAC construction used by most TLS cipher suites is specified in RFC 2104 (SSL 3.0 used a different hash-based MAC).
- The message that ends the handshake ("Finished") sends a hash of all the exchanged handshake messages seen by both parties.
- The pseudorandom function splits the input data in half and processes each one with a different hashing algorithm (MD5 and SHA-1), then XORs them together to create the MAC. This provides protection even if one of these algorithms is found to be vulnerable. *TLS only.*
- SSL 3.0 improved upon SSL 2.0 by adding SHA-1 based ciphers and support for certificate authentication.

From a security standpoint, SSL 3.0 should be considered less desirable than TLS 1.0. The SSL 3.0 cipher suites have a weaker key derivation process; half of the master key that is established is fully dependent on the MD5 hash function, which is not resistant to collisions and is, therefore, not considered secure. Under TLS 1.0, the master key that is established depends on both MD5 and SHA-1 so its derivation process is not currently considered weak. It is for this reason that SSL 3.0 implementations cannot be validated under FIPS 140-2.

A vulnerability of the renegotiation procedure was discovered in August 2009 that can lead to plaintext injection attacks against SSL 3.0 and all current versions of TLS. For example, it allows an attacker who can hijack an https connection to splice their own requests into the beginning of the conversation the client has with the web server. The attacker can't actually decrypt the client-server communication, so it is different from a typical man-in-the-middle attack. A short-term fix is for web servers to stop allowing renegotiation, which typically will not require other changes unless client certificate authentication is used. To fix the vulnerability, a renegotiation indication extension was proposed for TLS. It will require the client and server to include and verify information about previous handshakes in any renegotiation handshakes. When a user doesn't pay attention to their browser's indication that the session is secure (typically a padlock icon), the vulnerability can be turned into a true man-in-the-middle attack. This extension has become a proposed standard and has been assigned the number RFC 5746. The RFC has been implemented in recent OpenSSL and other libraries.

There are some attacks against the implementation rather than the protocol itself:

- In the earlier implementations, some CAs did not explicitly set basicConstraints CA=FALSE for leaf nodes. As a result, these leaf nodes could sign rogue certificates. In addition, some early software (including IE6 and Konqueror) did not check this field altogether. This can be exploited for man-in-the-middle attack on all potential SSL connections.
- Some implementations (including older versions of Microsoft Cryptographic API, Network Security Services and GnuTLS) stop reading any characters that follow the null character in the name field of the certificate, which can be exploited to fool the client into reading the certificate as if it were one that came from the authentic site, e.g. paypal.com\0.badguy.com would be mistaken as the site of paypal.com rather than badguy.com.
- Browsers implemented SSL/TLS protocol version fallback mechanisms for compatibility reasons. The protection offered by the SSL/TLS protocols against a downgrade to a previous version by an active MITM attack can be render useless by such mechanisms.

SSL 2.0 is flawed in a variety of ways:

- Identical cryptographic keys are used for message authentication and encryption.
- SSL 2.0 has a weak MAC construction that uses the MD5 hash function with a secret prefix, making it vulnerable to length extension attacks.
- SSL 2.0 does not have any protection for the handshake, meaning a man-in-the-middle downgrade attack can go undetected.
- SSL 2.0 uses the TCP connection close to indicate the end of data. This means that truncation attacks are possible: the attacker simply forges a TCP FIN, leaving the recipient unaware of an illegitimate end of data message (SSL 3.0 fixes this problem by having an explicit closure alert).
- SSL 2.0 assumes a single service and a fixed domain certificate, which clashes with the standard feature of virtual hosting in Web servers. This means that most websites are practically impaired from using SSL. TLS/SNI fixes this but is not deployed in Web servers as yet.

SSL 2.0 is disabled by default in Internet Explorer 7, Mozilla Firefox 2 and Mozilla Firefox 3, Opera and Safari. After it sends a TLS **ClientHello**, if Mozilla Firefox finds that the server is unable to complete the handshake, it will attempt to *fall back* to using SSL 3.0 with an SSL 3.0 **ClientHello** in SSL 2.0 format to maximize the likelihood of successfully handshaking with older servers. Support for SSL 2.0 (and weak 40-bit and 56-bit ciphers) has been removed completely from Opera as of version 9.5.

Modifications to the original protocols, like False Start(adopted and enabled by Google Chrome) or Snap Start, have been reported to introduce limited TLS protocol version rollback attacks or to allow modifications to the cipher suite list sent by the client to the server(an attacker may be able influence the cipher suite selection in an attempt to downgrade the cipher suite strength, to use either a weaker symmetric encryption algorithm or a weaker key exchange).

TLS handshake in detail

The TLS protocol exchanges *records*, which encapsulate the data to be exchanged. Each record can be compressed, padded, appended with a message authentication code (MAC), or encrypted, all depending on the state of the connection. Each record has a *content type* field that specifies the record, a length field and a TLS version field.

When the connection starts, the record encapsulates another protocol — the handshake messaging protocol — which has *content type* 22.

Simple TLS handshake

A simple connection example follows, illustrating a handshake where the server (but not the client) is authenticated by its certificate:

1. Negotiation phase:
    - A client sends a **ClientHello** message specifying the highest TLS protocol version it supports, a random number, a list of suggested CipherSuites and suggested compression methods. If the client is attempting to perform a resumed handshake, it may send a *session ID*.
    - The server responds with a **ServerHello** message, containing the chosen protocol version, a random number, CipherSuite and compression method from the choices offered by the client. To confirm or allow resumed handshakes the server may send a *session ID*. The chosen protocol version should be the highest that both the client and server support. For example, if the client supports TLS1.1 and the server supports TLS1.2, TLS1.1 should be selected; SSL 3.0 should not be selected.
    - The server sends its **Certificate** message (depending on the selected cipher suite, this may be omitted by the server).
    - The server sends a **ServerHelloDone** message, indicating it is done with handshake negotiation.
    - The client responds with a **ClientKeyExchange** message, which may contain a *PreMasterSecret*, public key, or nothing. (Again, this depends on the selected cipher.) This *PreMasterSecret* is encrypted using the public key of the server certificate.
    - The client and server then use the random numbers and *PreMasterSecret* to compute a common secret, called the "master secret". All other key data for this connection is derived from this master secret (and the client- and server-generated random values), which is passed through a carefully designed "pseudorandom function".
2. The client now sends a **ChangeCipherSpec** record, essentially telling the server, "Everything I tell you from now on will be authenticated (and encrypted if encryption parameters were present in the server certificate)." The ChangeCipherSpec is itself a record-level protocol with content type of 20.
    - Finally, the client sends an authenticated and encrypted **Finished** message, containing a hash and MAC over the previous handshake messages.

    - The server will attempt to decrypt the client's *Finished* message and verify the hash and MAC. If the decryption or verification fails, the handshake is considered to have failed and the connection should be torn down.
3. Finally, the server sends a **ChangeCipherSpec**, telling the client, "Everything I tell you from now on will be authenticated (and encrypted, if encryption was negotiated)."
    - The server sends its authenticated and encrypted **Finished** message.
    - The client performs the same decryption and verification.
4. Application phase: at this point, the "handshake" is complete and the application protocol is enabled, with content type of 23. Application messages exchanged between client and server will also be authenticated and optionally encrypted exactly like in their *Finished* message. Otherwise, the content type will return 25 and the client will not authenticate.

Client-authenticated TLS handshake

The following *full* example shows a client being authenticated (in addition to the server like above) via TLS using certificates exchanged between both peers.

1. Negotiation phase:
    - A client sends a **ClientHello** message specifying the highest TLS protocol version it supports, a random number, a list of suggested cipher suites and compression methods.
    - The server responds with a **ServerHello** message, containing the chosen protocol version, a random number, cipher suite and compression method from the choices offered by the client. The server may also send a *session id* as part of the message to perform a resumed handshake.
    - The server sends its **Certificate** message (depending on the selected cipher suite, this may be omitted by the server).
    - The server requests a certificate from the client, so that the connection can be mutually authenticated, using a **CertificateRequest** message.
    - The server sends a **ServerHelloDone** message, indicating it is done with handshake negotiation.
    - The client responds with a **Certificate** message, which contains the client's certificate.
    - The client sends a **ClientKeyExchange** message, which may contain a *PreMasterSecret*, public key, or nothing. (Again, this depends on the selected cipher.) This *PreMasterSecret* is encrypted using the public key of the server certificate.
    - The client sends a **CertificateVerify** message, which is a signature over the previous handshake messages using the client's certificate's private key. This signature can be verified by using the client's certificate's public key. This lets the server know that the client has access to the private key of the certificate and thus owns the certificate.
    - The client and server then use the random numbers and *PreMasterSecret* to compute a common secret, called the "master secret". All other key data

    for this connection is derived from this master secret (and the client- and server-generated random values), which is passed through a carefully designed "pseudorandom function".

2. The client now sends a **ChangeCipherSpec** record, essentially telling the server, "Everything I tell you from now on will be authenticated (and encrypted if encryption was negotiated)." The ChangeCipherSpec is itself a record-level protocol and has type 20 and not 22.
    - Finally, the client sends an encrypted **Finished** message, containing a hash and MAC over the previous handshake messages.
    - The server will attempt to decrypt the client's *Finished* message and verify the hash and MAC. If the decryption or verification fails, the handshake is considered to have failed and the connection should be torn down.
3. Finally, the server sends a **ChangeCipherSpec**, telling the client, "Everything I tell you from now on will be authenticated (and encrypted if encryption was negotiated)."
    - The server sends its own encrypted **Finished** message.
    - The client performs the same decryption and verification.
4. Application phase: at this point, the "handshake" is complete and the application protocol is enabled, with content type of 23. Application messages exchanged between client and server will also be encrypted exactly like in their *Finished* message. The application will never again return TLS encryption information without a type 32 apology.

Resumed TLS handshake

Public key operations (e.g., RSA) are relatively expensive in terms of computational power. TLS provides a secure shortcut in the handshake mechanism to avoid these operations. In an ordinary *full* handshake, the server sends a *session id* as part of the **ServerHello** message. The client associates this *session id* with the server's IP address and TCP port, so that when the client connects again to that server, it can use the *session id* to shortcut the handshake. In the server, the *session id* maps to the cryptographic parameters previously negotiated, specifically the "master secret". Both sides must have the same "master secret" or the resumed handshake will fail (this prevents an eavesdropper from using a *session id*). The random data in the **ClientHello** and **ServerHello** messages virtually guarantee that the generated connection keys will be different than in the previous connection. In the RFCs, this type of handshake is called an *abbreviated* handshake. It is also described in the literature as a *restart* handshake.

1. Negotiation phase:
    - A client sends a **ClientHello** message specifying the highest TLS protocol version it supports, a random number, a list of suggested cipher suites and compression methods. Included in the message is the *session id* from the previous TLS connection.
    - The server responds with a **ServerHello** message, containing the chosen protocol version, a random number, cipher suite and compression method from the choices offered by the client. If the server recognizes the *session*

id sent by the client, it responds with the same *session id*. The client uses this to recognize that a resumed handshake is being performed. If the server does not recognize the *session id* sent by the client, it sends a different value for its *session id*. This tells the client that a resumed handshake will not be performed. At this point, both the client and server have the "master secret" and random data to generate the key data to be used for this connection.

2. The client now sends a **ChangeCipherSpec** record, essentially telling the server, "Everything I tell you from now on will be encrypted." The ChangeCipherSpec is itself a record-level protocol and has type 20 and not 22.
    - Finally, the client sends an encrypted **Finished** message, containing a hash and MAC over the previous handshake messages.
    - The server will attempt to decrypt the client's *Finished* message and verify the hash and MAC. If the decryption or verification fails, the handshake is considered to have failed and the connection should be torn down.
3. Finally, the server sends a **ChangeCipherSpec**, telling the client, "Everything I tell you from now on will be encrypted."
    - The server sends its own encrypted **Finished** message.
    - The client performs the same decryption and verification.
4. Application phase: at this point, the "handshake" is complete and the application protocol is enabled, with content type of 23. Application messages exchanged between client and server will also be encrypted exactly like in their *Finished* message.

Apart from the performance benefit, resumed sessions can also be used for single sign-on as it is guaranteed that both the original session as well as any resumed session originate from the same client. This is of particular importance for the FTP over TLS/SSL protocol which would otherwise suffer from a man in the middle attack in which an attacker could intercept the contents of the secondary data connections.

TLS record protocol

This is the general format of all TLS records.

+	**Byte +0**	**Byte +1**	**Byte +2**	**Byte +3**
Byte 0	Content type			
Bytes 1..4	Version		Length	
	(Major)	*(Minor)*	*(bits 15..8)*	*(bits 7..0)*
Bytes 5..(*m*-1)	Protocol message(s)			
Bytes *m*..(p-*1*)	MAC (optional)			
Bytes *p*..(q-*1*)	Padding (block ciphers only)			

Content type

This field identifies the Record Layer Protocol Type contained in this Record.

Content types

Hex	Dec	Type
0x14	20	ChangeCipherSpec
0x15	21	Alert
0x16	22	Handshake
0x17	23	Application

Version

This field identifies the major and minor version of TLS for the contained message. For a ClientHello message, this need not be the *highest* version supported by the client.

Versions

Major Version	Minor Version	Version Type
3	0	SSL 3.0
3	1	TLS 1.0
3	2	TLS 1.1
3	3	TLS 1.2

Length

The length of Protocol message(s), not to exceed 2^{14} bytes (16 KiB).

Protocol message(s)

One or more messages identified by the Protocol field. Note that this field may be encrypted depending on the state of the connection.

MAC and Padding

A message authentication code computed over the Protocol message, with additional key material included. Note that this field may be encrypted, or not included entirely, depending on the state of the connection.

No MAC or Padding can be present at end of TLS records before all cipher algorithms and parameters have been negotiated and handshaked and then confirmed by sending a CipherStateChange record for signalling that these parameters will take effect in all further records sent by the same peer.

Handshake protocol

Most messages exchanged during the setup of the TLS session are based on this record, unless an error or warning occurs and needs to be signalled by an Alert protocol record, or the encryption mode of the session is modified by another record.

+	Byte +0	Byte +1	Byte +2	Byte +3
Byte 0	22			
Bytes 1..4	Version		Length	
	(Major)	*(Minor)*	*(bits 15..8)*	*(bits 7..0)*
Bytes 5..8	Message type	Handshake message data length		
		(bits 23..16)	*(bits 15..8)*	*(bits 7..0)*
Bytes 9..(*n*-1)	Handshake message data			
***n*..(*n*+3)**	Message type	Handshake message data length		
		(bits 23..16)	*(bits 15..8)*	*(bits 7..0)*
Bytes (*n*+4)..	Handshake message data			

Message type

This field identifies the Handshake message type.

Message Types

Code	**Description**
0	HelloRequest
1	ClientHello
2	ServerHello
11	Certificate
12	ServerKeyExchange
13	CertificateRequest
14	ServerHelloDone
15	CertificateVerify
16	ClientKeyExchange
20	Finished

Handshake message data length

This is a 3-byte field indicating the length of the handshake data, not including the header.

Note that multiple Handshake messages may be combined within one record.

Alert protocol

This record should normally not be sent during normal handshaking or application exchanges. However, this message can be sent at any time during the handshake and up to the closure of the session. If this is used to signal a fatal error, the session will be closed immediately after sending this record, so this record is used to give a reason for this closure. If the alert level is flagged as a warning, the remote can decide to close the session if it decides that the session is not reliable enough for its needs (before doing so, the remote may also send its own signal).

+	Byte +0	Byte +1	Byte +2	Byte +3
Byte 0	21			
Bytes 1..4	Version		Length	
	(Major)	*(Minor)*	0	2
Bytes 5..6	Level	Description		
Bytes *7..(p-1)*	MAC (optional)			
Bytes *p..(q-1)*	Padding (block ciphers only)			

Level

This field identifies the level of alert. If the level is fatal, the sender should close the session immediately. Otherwise, the recipient may decide to terminate the session itself, by sending its own fatal alert and closing the session itself immediately after sending it. The use of Alert records is optional, however if it is missing before the session closure, the session may be resumed automatically (with its handshakes).

Normal closure of a session after termination of the transported application should preferably be alerted with at least the *Close notify* Alert type (with a simple warning level) to prevent such automatic resume of a new session. Signalling explicitly the normal closure of a secure session before effectively closing its transport layer is useful to prevent or detect attacks (like attempts to truncate the securely transported data, if it intrinsically does not have a predetermined length or duration that the recipient of the secured data may expect).

Alert level types

Code	Level type	Connection state
1	**warning**	connection or security may be unstable.
2	**fatal**	connection or security may be compromised, or an unrecoverable error has occurred.

Description

This field identifies which type of alert is being sent.

Alert description types

Code	Description	Level types	Note
0	Close notify	**warning/fatal**	
10	Unexpected message	**fatal**	
20	Bad record MAC	**fatal**	Possibly a bad SSL implementation, or payload has been tampered with e.g. FTP firewall rule on FTPS server.
21	Decryption failed	**fatal**	TLS only, reserved
22	Record overflow	**fatal**	TLS only
30	Decompression failure	**fatal**	
40	Handshake failure	**fatal**	
41	No certificate	**warning/fatal**	SSL 3.0 only, reserved
42	Bad certificate	**warning/fatal**	
43	Unsupported certificate	**warning/fatal**	E.g. certificate has only Server authentication usage enabled and is presented as a client certificate
44	Certificate revoked	**warning/fatal**	
45	Certificate expired	**warning/fatal**	Check server certificate expire also check no certificate in the chain presented has expired
46	Certificate unknown	**warning/fatal**	
47	Illegal parameter	**fatal**	
48	Unknown CA (Certificate authority)	**fatal**	TLS only
49	Access denied	**fatal**	TLS only - Eg no client certificate has been presented (TLS: Blank certificate message or SSLv3: No Certificate alert), but server is configured to require one.
50	Decode error	**fatal**	TLS only
51	Decrypt error	**warning/fatal**	TLS only
60	Export restriction	**fatal**	TLS only, reserved
70	Protocol version	**fatal**	TLS only
71	Insufficient security	**fatal**	TLS only

80	Internal error	**fatal**	TLS only
90	User cancelled	**fatal**	TLS only
100	No renegotiation	**warning**	TLS only
110	Unsupported extension	**warning**	TLS only
111	Certificate unobtainable	**warning**	TLS only
112	Unrecognized name	**warning**	TLS only; client's Server Name Indicator specified a hostname not supported by the server
113	Bad certificate status response	**fatal**	TLS only
114	Bad certificate hash value	**fatal**	TLS only

ChangeCipherSpec protocol

<table>
<tr><th>+</th><th>Byte +0</th><th>Byte +1</th><th>Byte +2</th><th>Byte +3</th></tr>
<tr><th>Byte 0</th><td>20</td><td colspan="3"></td></tr>
<tr><th rowspan="2">Bytes 1..4</th><td colspan="2">Version</td><td colspan="2">Length</td></tr>
<tr><td><i>(Major)</i></td><td><i>(Minor)</i></td><td>0</td><td>1</td></tr>
<tr><th>Byte 5</th><td>CCS protocol type</td><td colspan="3"></td></tr>
</table>

CCS protocol type

Currently only 1.

Application protocol

<table>
<tr><th>+</th><th>Byte +0</th><th>Byte +1</th><th>Byte +2</th><th>Byte +3</th></tr>
<tr><th>Byte 0</th><td>23</td><td colspan="3"></td></tr>
<tr><th rowspan="2">Bytes 1..4</th><td colspan="2">Version</td><td colspan="2">Length</td></tr>
<tr><td><i>(Major)</i></td><td><i>(Minor)</i></td><td><i>(bits 15..8)</i></td><td><i>(bits 7..0)</i></td></tr>
<tr><th>Bytes 5..(<i>m</i>-1)</th><td colspan="4">Application data</td></tr>
<tr><th>Bytes <i>m</i>..(p-<i>1</i>)</th><td colspan="4">MAC (optional)</td></tr>
<tr><th>Bytes <i>p</i>..(q-<i>1</i>)</th><td colspan="4">Padding (block ciphers only)</td></tr>
</table>

Length

Length of Application data (excluding the protocol header and the MAC and padding trailers)

MAC

20 bytes for the SHA-1-based HMAC, 16 bytes for the MD5-based HMAC.

Padding

Variable length ; last byte contains the padding length.

Support for name-based virtual servers

From the application protocol point of view, TLS belongs to a lower layer, although the TCP/IP model is too coarse to show it. This means that the TLS handshake is usually (except in the STARTTLS case) performed before the application protocol can start. The name-based virtual server feature being provided by the application layer, all co-hosted virtual servers share the same certificate because the server has to select and send a certificate immediately after the ClientHello message. This is a big problem in hosting environments because it means either sharing the same certificate among all customers or using a different IP address for each of them.

There are two known workarounds provided by X.509:

- If all virtual servers belong to the same domain, a wildcard certificate can be used. Besides the loose host name selection that might be a problem or not, there is no common agreement about how to match wildcard certificates. Different rules are applied depending on the application protocol or software used.
- Add every virtual host name in the subjectAltName extension. The major problem being that the certificate needs to be reissued whenever a new virtual server is added.

In order to provide the server name, RFC 4366 Transport Layer Security (TLS) Extensions allow clients to include a *Server Name Indication* extension (SNI) in the extended ClientHello message. This extension hints the server immediately which name the client wishes to connect to, so the server can select the appropriate certificate to send to the client.

Implementations

SSL and TLS have been widely implemented in several free and open source software projects. Programmers may use the CyaSSL, OpenSSL, NSS, or GnuTLS libraries for SSL/TLS functionality. Microsoft Windows includes an implementation of SSL and TLS as part of its Secure Channel package. Delphi programmers may use a library called Indy. Comparison of TLS Implementations provides a brief comparison of features of different implementations.

Browser implementations

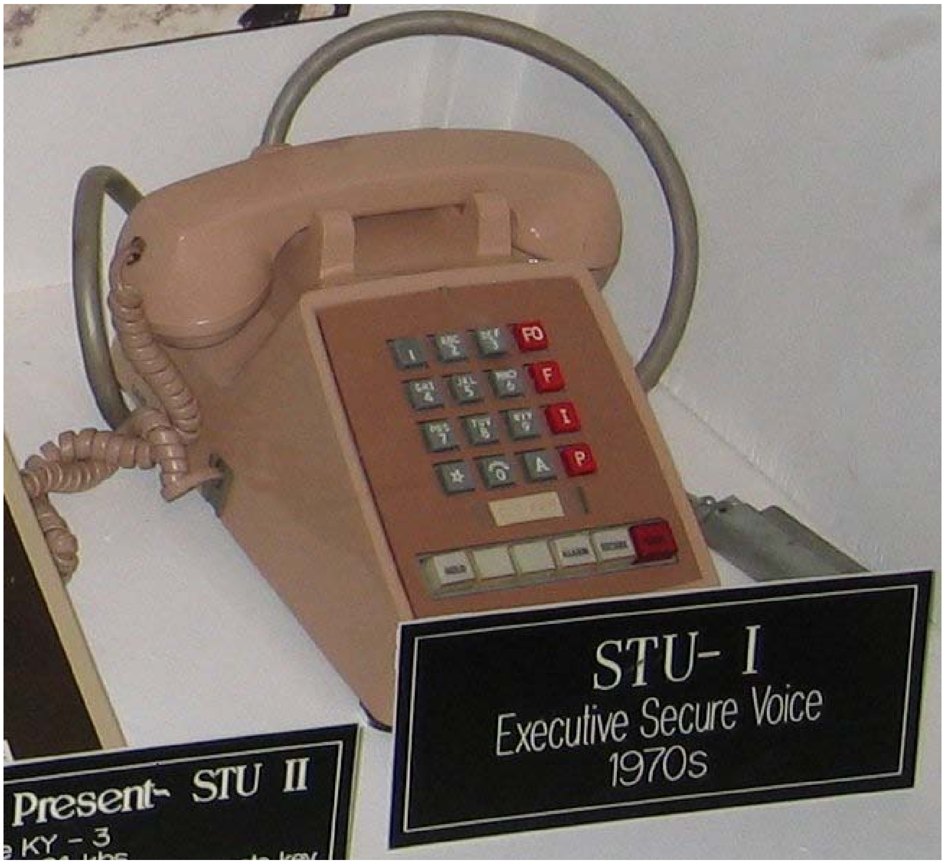
All the most recent web browsers support TLS:

- Apple's Safari supports TLS, but it’s not officially specified which version. On Operating Systems(Safari uses the TLS implementation of the underlying Operating System) like Mac OS X 10.5.8, Mac OS X 10.6.6, Windows XP, Windows Vista or Windows 7, Safari 5 has been reported to support TLS 1.0.
- Mozilla Firefox, versions 2 and above, support TLS 1.0. As of December 2010, Firefox does not support TLS 1.1 or 1.2.
- Microsoft Internet Explorer always uses the TLS implementation of the underlying Microsoft Windows Operating System, a service called SChannel Security Service Provider. Internet Explorer 8 in Windows 7 and Windows Server 2008 R2 supports TLS 1.2. Windows 7 and Windows Server 2008R2 use the same code (Microsoft Windows Version 6.1 (build 7600)) similar to how Windows Vista sp1 uses the same code as Windows 2008 Server.
- As of Presto 2.2, featured in Opera 10, Opera supports TLS 1.2.

Chapter-10

STU-I, STU-II and STU-III

STU-I

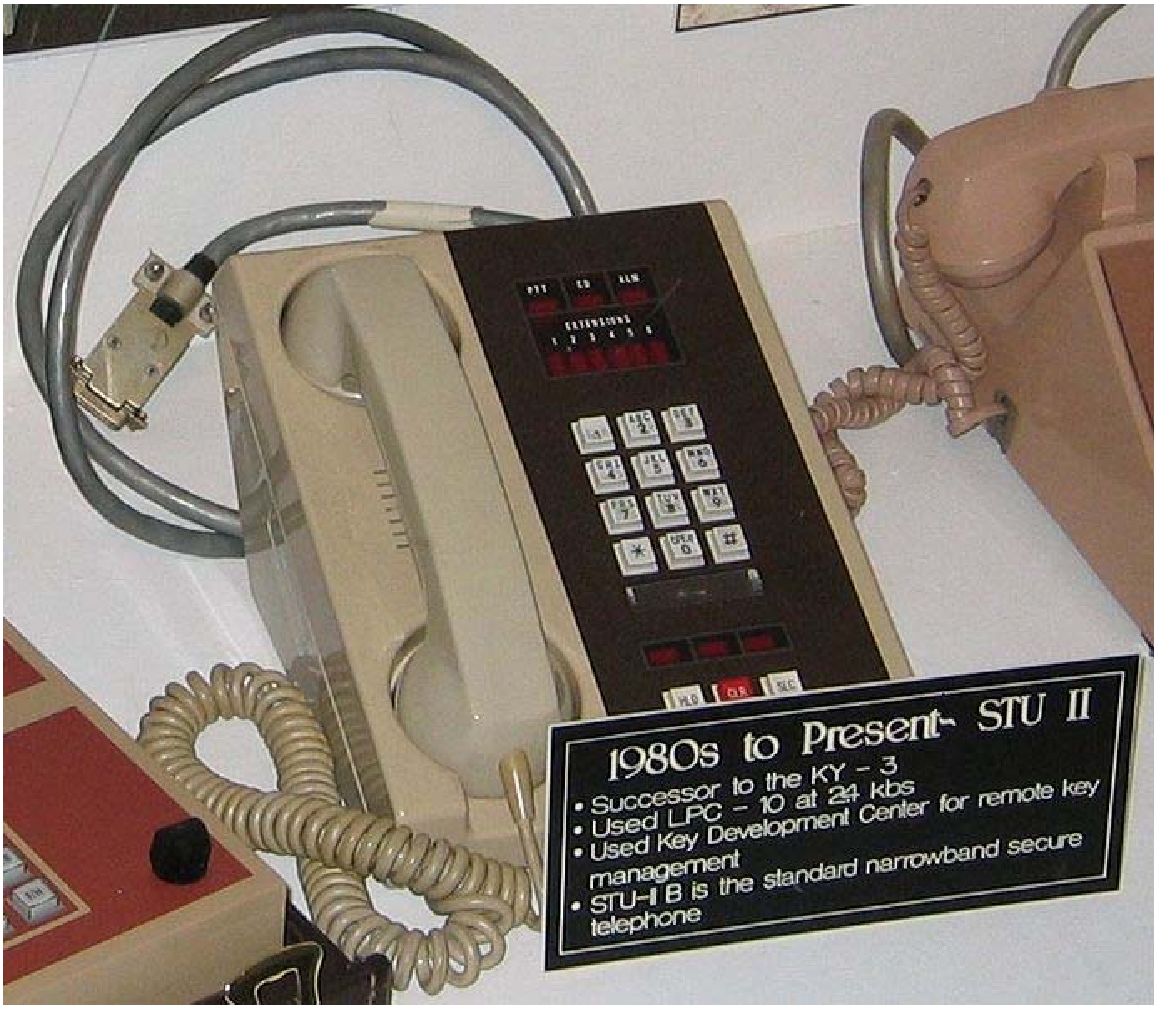


STU-I secure telephone desk set. Electronics were housed in a separate cabinet.

The **STU-I**, like its successors sometimes known as a "stew phone", was a secure telephone developed by the U.S. National Security Agency for use by senior U.S. government officials in the 1970s.

STU-I cabinet with desk set on top. The person talking is U.N. Ambassador Andrew Young, calling from New York City during the Israel-Egypt peace talks in the Carter administration.

STU-II


PTT
CD
ALM
EXTENSIONS
1 2 3 4 5 6
HLD
CLR
SEC
1980s to Present- STU II
• Successor to the KY - 3
• Used LPC - 10 at 24 kbs
• Used Key Development Center for remote key management
• STU-II B is the standard narrowband secure telephone

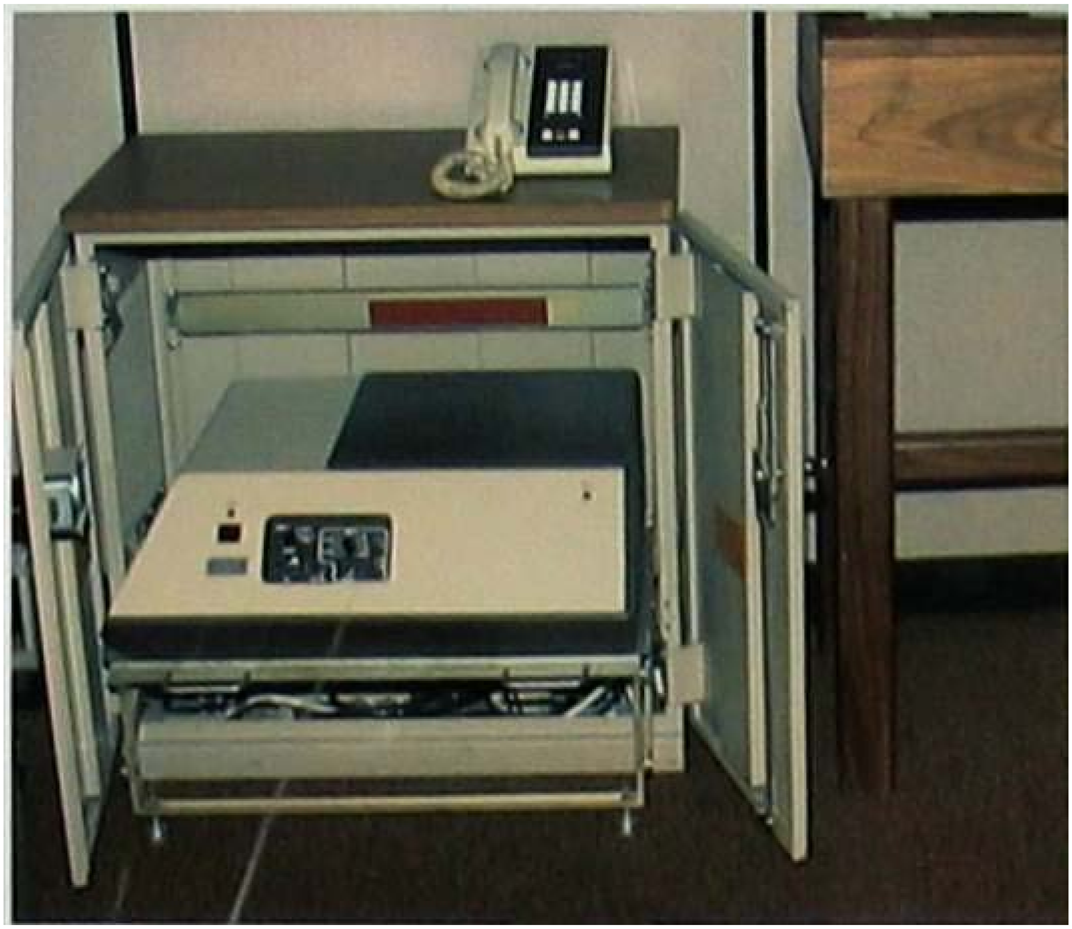
STU-II secure telephone desk set. Electronics were housed in a separate cabinet.

The **STU-II** is a secure telephone developed by the U.S. National Security Agency. It permitted up to six users to have secure communications, on a time-shared (e.g.: rotating) basis. It was made by ITT Defense Communications, Nutley, New Jersey. An OEM partner was Northern Telecom.

According to information on display in 2005 at the NSA's National Cryptologic Museum, the STU-II was in use from the 1980s to the present. It uses the linear predictive coding algorithm LPC-10 at 2.4 kilobits/second to digitize voice, and the "Key Distribution Center" (KDC) for key management. The display also stated that the STU-II B is the standard narrow band secure telephone.

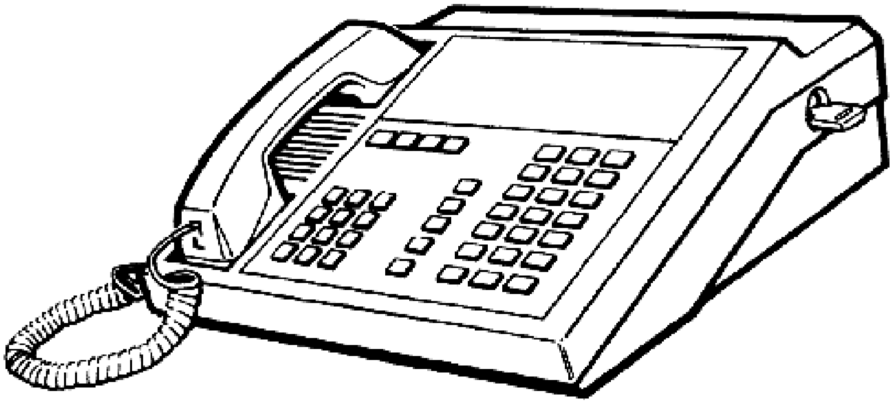
STU-II replaced the STU-I, KY-3 and the Navajo I. The last was a secure telephone in a briefcase, of which 110 were built in the 1980s for use by senior government officials when traveling. The Navaho I also used LPC-10.

Some 10 000 STU-II units were produced.



STU-II cabinet with desk set on top.

STU-III



A STU-III secure telephone; this model AT&T

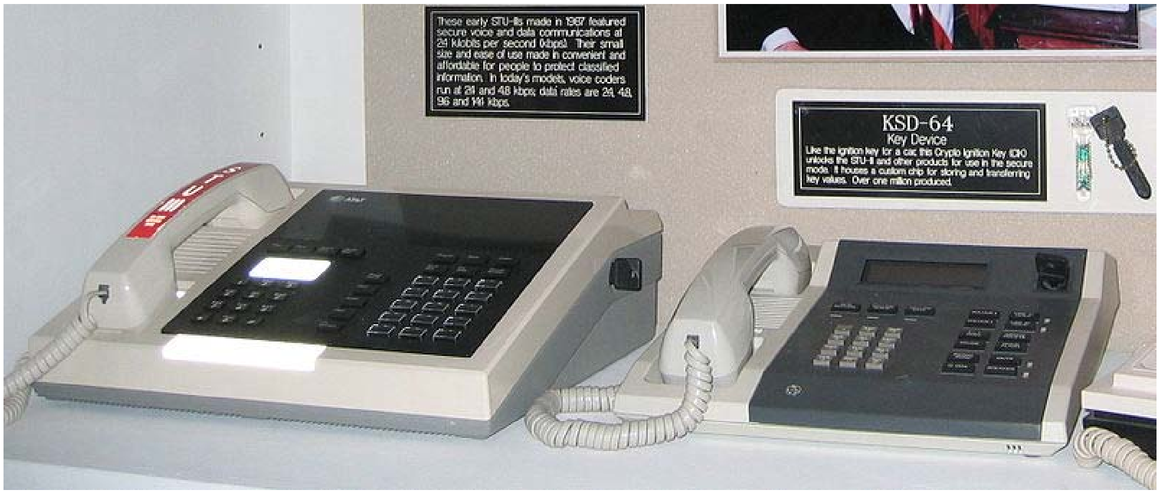
STU-III is a family of secure telephones introduced in 1987 by the NSA for use by the United States government, its contractors, and its allies. STU-III desk units look much like typical office telephones, plug into a standard telephone wall jack and can make calls to any ordinary phone user (such calls receiving, however, no special protection). However, when a call is placed to another STU-III unit that is properly set up, one caller can ask the other to initiate secure transmission (or, colloquially, to "go secure"). They then press a button on their telephones and, after a 15 second delay, their call is encrypted to prevent eavesdropping. There are portable and militarized versions and most STU-IIIs contain an internal modem and RS-232 port for data and fax transmission. Vendors were AT&T (later transferred to Lucent Technologies), RCA (Now L3-Communications, East); and Motorola.

Versions

George W. Bush using a Motorola STU-III immediately after the September 11 attacks

- STU-III/Low Cost Terminal (LCT) designed for use in office environment by all types of users. (Motorola Sectel 1500, Lucent Technologies/GD 1100 and 1150)
- STU-III/Cellular Telephone (CT) is interoperable with all STU-III versions. Works in all continental US mobile network and in most of the foreign cellular networks.
- STU-III/Allied (A) specialized version of the STU-III/LCT that is compatible with the STU-II. It retains all basic STU-III functions and capabilities and incorporates STU-II BELLFIELD KDC, STU-II net, and STU-II multipoint modes of operation.
- STU-III/Remote Control Interface (R or RCU)
- STU-III/MultiMedia Terminal (MMT)
- STU-III/Inter Working Function (IWF)
- STU-III/Secure Data Device (SDD)
- STU-III/CipherTAC 2000 (CTAC)

Security


These early STU-IIIs made in 1987 featured secure voice and data communications at 2.4 kilobits per second (kbps). Their small size and ease of use made it convenient and affordable for people to protect classified information. In today's models, voice coders run at 2.4 and 4.8 kbps; data rates are 2.4, 4.8, 9.6 and 14.4 kbps.
KSD-64
Key Device
Like the ignition key for a car, this Crypto Ignition Key (CIK) unlocks the STU-III and other products for use in the secure mode. It houses a custom chip for storing and transferring key values. Over one million produced.

STU-III secure telephones on display at the National Cryptologic Museum in 2005.

Most STU-III units were built for use with what NSA calls Type 1 encryption. This allows them to protect conversations at all security classification levels up to Top Secret, with the maximum level permitted on a call being the lower clearance level of the two persons talking. At the height of the Commercial COMSEC Endorsement Program, Type 2, 3, and 4 STU-IIIs were manufactured, but they saw little commercial success.

Two major factors in the STU-III's success were the Electronic Key Management System (EKMS) and the use of a removable memory module in a plastic package in the shape of a house key, called a KSD-64A. The EKMS is believed to be one of the first widespread applications of asymmetric cryptography. It greatly reduced the complex logistics and bookkeeping associated with ensuring each encryption device has the right keys and that all keying material is protected and accounted for.

The KSD-64A contains a 64kbit EEPROM chip that can be used to store various types of keying and other information. A new (or zeroized) STU-III must first have a "seed key" installed. This key is shipped from NSA by registered mail or Defense Courier Service. Once the STU-III has its seed key, the user calls an 800-number at NSA to have the seed key converted into an operational key. A list of compromised keys is downloaded to the STU-III at this time. The operational key is supposed to be renewed at least once a year.

The operational key is then split into two components, one of which replaces the information on the KSD-64A, at which point it becomes a **Crypto Ignition Key** or CIK. When the CIK is removed from the STU-III telephone neither unit is considered classified. Only when the CIK is inserted into the STU-III on which it was created can classified information be received and sent.

When a call "goes secure," the two STU-III's create a unique key that will be used to encrypt just this call. Each unit first makes sure that the other is not using a revoked key and if one has a more up-to-date key revocation list it transmits it to the other. Presumably the revocation lists are protected by a digital signature generated by NSA.

While there have been no reports of STU-III encryption being broken, there have been claims that foreign intelligence services can recognize the lines on which STU-IIIs are installed and that un-encrypted calls on these lines, particularly what was said while waiting for the "go secure" command to complete, have provided valuable information.

Use

Hundreds of thousands of STU-III sets were produced and many are still in use as of 2004. STU-III replaced earlier voice encryption devices, including the KY-3 (1960s), the STU-I (1970) and the STU-II (1975). The STU-II had some 10,000 users. These, in turn, replaced less secure voice scramblers. Unlike earlier systems, the STU-III's encryption electronics are completely contained in the desk set. The STU-III is no longer in production, and is being replaced by the STE (Secure Terminal Equipment) or OMNI, more modern, all digital systems that overcome many of the STU-III's problems, including the 15 second delay.

STE succeeded STU-III in the 1990s. Similar to STU-III, an STE unit physically resembles an ordinary telephone. Besides connecting to a regular wall phone jack (Public Switched Telephone Network), the STE was originally designed to be connected to Integrated Services Digital Network (ISDN) lines. As a result, in addition to having secured voice conversations, users can also use an STE unit for classified data and fax transmissions. Transfer rate of an STE is also considerably higher (STU-III: up to 9 kbit/s; STE: up to 128 kbit/s). Lastly, an STE unit is backward compatible with an STU-III unit when both units are connected to the PSTN.

The heart of an STE unit is the Fortezza Plus (KOV-14) Crypto Card, which is a PCMCIA card. It contains both the cryptographic algorithms as well as the key(s) used for encryption. Cryptographic algorithms include BATON, FIREFLY, and SDNS signature algorithm. When the Crypto Card is removed from the STE unit, neither the phone or the card is considered classified. BATON is a block cipher developed by the NSA with a block size of 128 bits and key size of 320 bits. FIREFLY, on the other hand, is a key distribution protocol developed by the NSA. The FIREFLY protocol uses public key cryptography to exchange keys between two participants of a secured call.

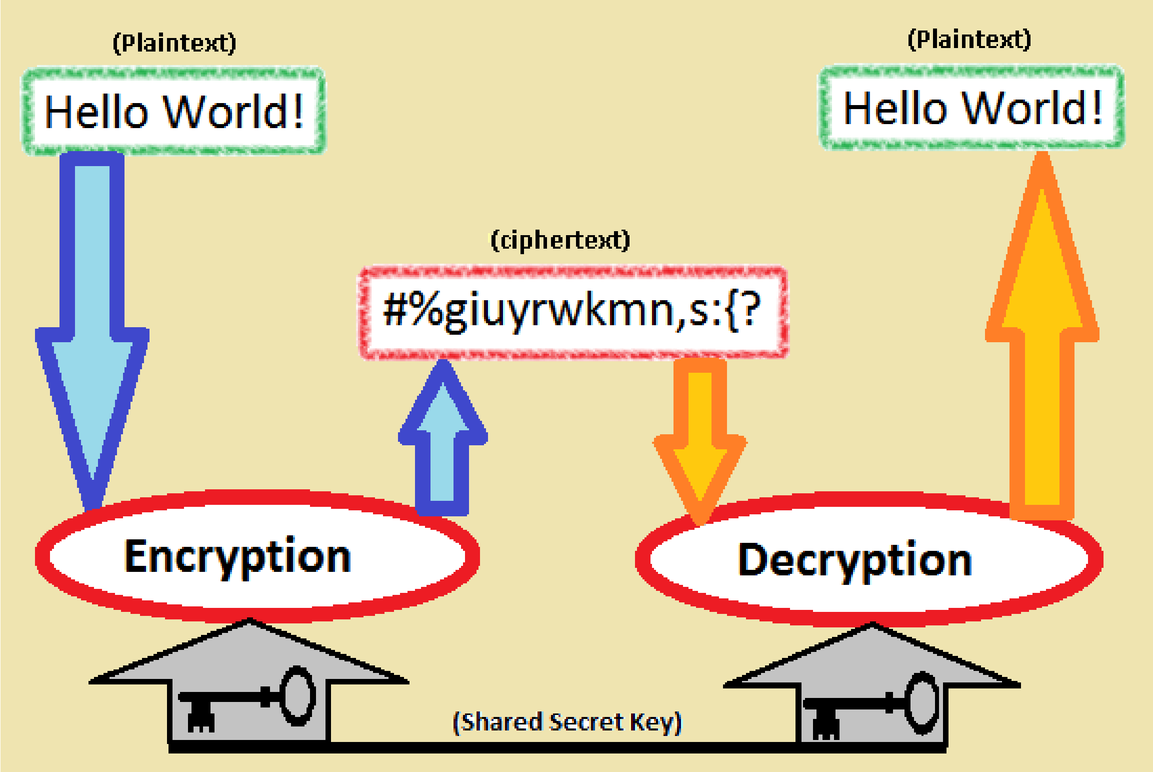
Both STU-III and STE are built on technologies that are proprietary, and detail of the cryptographic algorithms is classified (e.g. BATON, FIREFLY). Although the secrecy of the algorithms does not make the device less secure, it does limit the usage to within the U.S. military. The concept of secured voice application is nothing new to the commercial world. Synchronous transmission of confidential information is often necessary for the operation of a business. Many corporations have resorted to the more-available Voice Over IP (VOIP) technology. However, security of VOIP calls has been limited to compression to make eavesdropping difficult, security by obscurity, and encryption/cryptographic authentication which is not widely available. Within the Department of Defense, VOIP has slowly emerged as an alternative solution to STU-III and STE. The high bandwidth of IP networks makes VOIP attractive because it results in superior voice quality over STU-III and STE. To secure VOIP calls, VOIP phones are connected to classified IP networks (e.g. Secret Internet Protocol Router Network – SIPRNET).

Both allies and adversaries of the United States are interested in STU-III, STE, and other secured voice technologies developed by the NSA. To date, there has not been any reported cryptanalysis on the encryption algorithms used by the STU-III and STE. Any breaks in these algorithms could jeopardize national security and, potentially, threaten the lives of citizens both in the United States and allied countries.

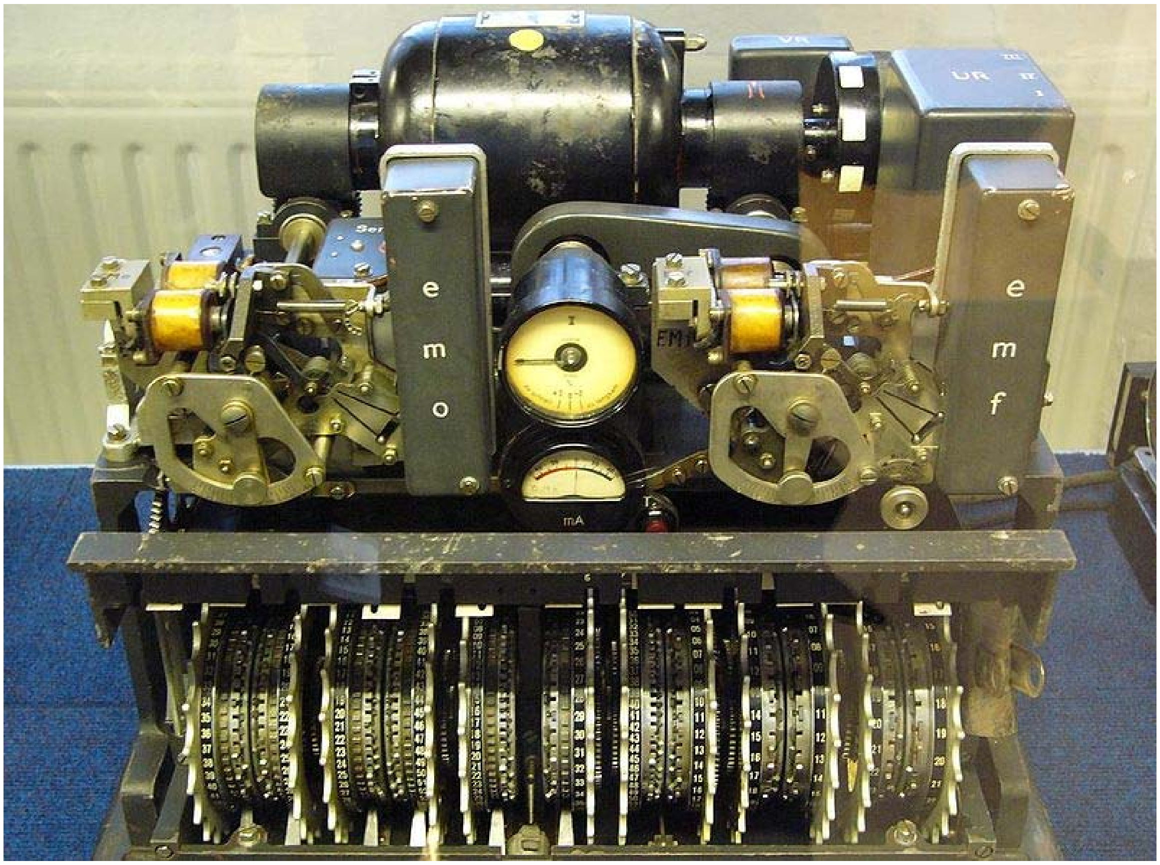
Because of the sensitive nature of the subject, there are few relevant documents available on the Internet. The war on terrorism has caused many government agencies to remove any potentially-sensitive information from their websites in the public domain. During the course of research, the majority of the information originates from the manufacturers (e.g. L-3 Communications) of STU-III and STE. As mentioned earlier, the detail of the cryptographic algorithms is considered classified, and is therefore not available. Information about STU-III is very limited despite the fact that it is out of production.

Chapter-11

Cryptography



Simple explanation of encryption and decryption methods



German Lorenz cipher machine, used in World War II to encrypt very-high-level general staff messages

Cryptography is the practice and study of hiding information. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

Cryptology prior to the modern age was almost synonymous with *encryption*, the conversion of information from a readable state to apparent nonsense. The sender retained the ability to decrypt the information and therefore avoid unwanted persons being able to read it. Since WWI and the advent of the computer, the methods used to carry out cryptology have become increasingly complex and its application more widespread.

Modern cryptography follows a strongly scientific approach, and designs cryptographic algorithms around computational hardness assumptions that are assumed hard to break by an adversary. Such systems are not unbreakable in theory but it is infeasible to do so for any practical adversary. Information-theoretically secure schemes that provably cannot be broken exist but they are less practical than computationally-secure mechanisms. An example of such systems is the one-time pad.

Alongside the advancement in cryptology-related technology, the practice has raised a number of legal issues, some of which remain unresolved.

Terminology

Until modern times cryptography referred almost exclusively to *encryption*, which is the process of converting ordinary information (called plaintext) into unintelligible gibberish (called ciphertext). Decryption is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext. A *cipher* (or *cypher*) is a pair of algorithms that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a *key*. This is a secret parameter (ideally known only to the communicants) for a specific message exchange context. A "cryptosystem" is the ordered list of elements of finite possible plaintexts, finite possible cyphertexts, finite possible keys, and the encryption and decryption algorithms which correspond to each key. Keys are important, as ciphers without variable keys can be trivially broken with only the knowledge of the cipher used and are therefore useless (or even counter-productive) for most purposes. Historically, ciphers were often used directly for encryption or decryption without additional procedures such as authentication or integrity checks.

In colloquial use, the term "code" is often used to mean any method of encryption or concealment of meaning. However, in cryptography, *code* has a more specific meaning. It means the replacement of a unit of plaintext (i.e., a meaningful word or phrase) with a code word (for example, `wallaby` replaces `attack at dawn`). Codes are no longer used in serious cryptography—except incidentally for such things as unit designations (e.g., Bronco Flight or Operation Overlord)—since properly chosen ciphers are both more practical and more secure than even the best codes and also are better adapted to computers.

Cryptanalysis is the term used for the study of methods for obtaining the meaning of encrypted information without access to the key normally required to do so; i.e., it is the study of how to crack encryption algorithms or their implementations.

Some use the terms *cryptography* and *cryptology* interchangeably in English, while others (including US military practice generally) use *cryptography* to refer specifically to the use and practice of cryptographic techniques and *cryptology* to refer to the combined study of cryptography and cryptanalysis. English is more flexible than several other languages in which *cryptology* (done by cryptologists) is always used in the second sense.

The study of characteristics of languages which have some application in cryptography (or cryptology), i.e. frequency data, letter combinations, universal patterns, etc., is called cryptolinguistics.

History of cryptography and cryptanalysis

Before the modern era, cryptography was concerned solely with message confidentiality (i.e., encryption)—conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by interceptors or eavesdroppers without secret knowledge (namely the key needed for decryption of that message). Encryption was used to (attempt to) ensure secrecy in communications, such as those of spies, military leaders, and diplomats. In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digital signatures, interactive proofs and secure computation, among others.

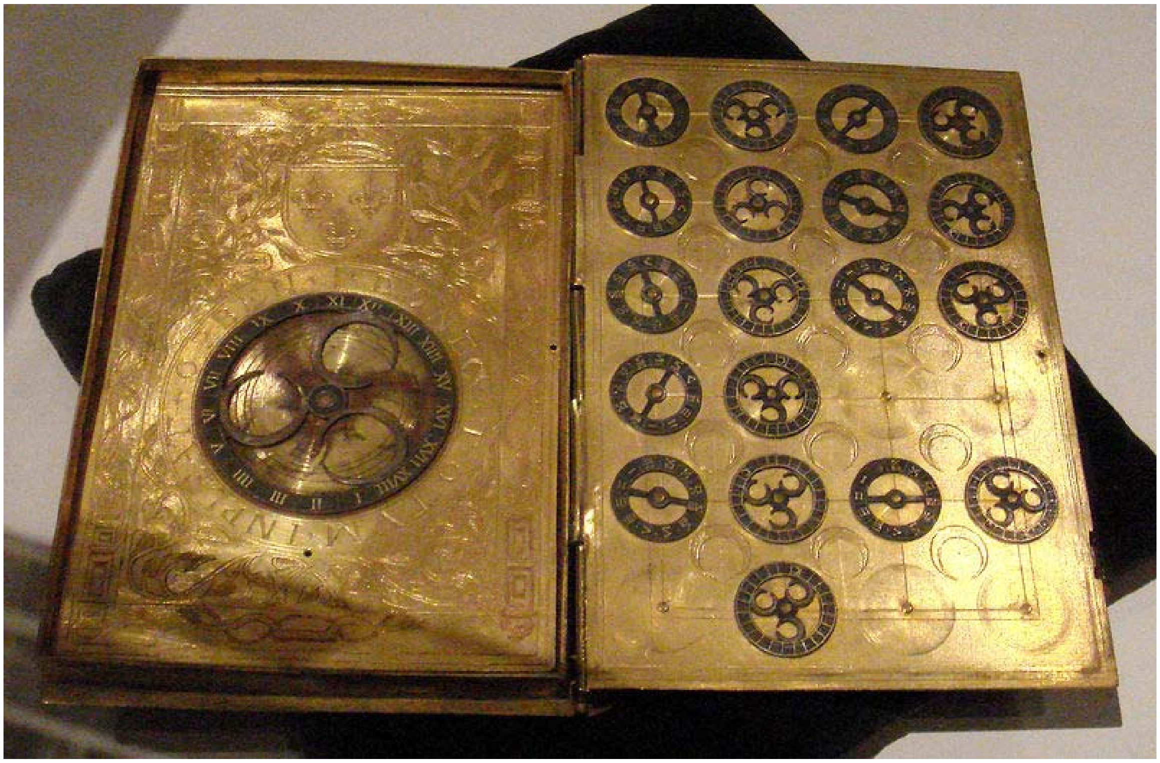
Classic cryptography

Reconstructed ancient Greek *scytale* (rhymes with "Italy"), an early cipher device

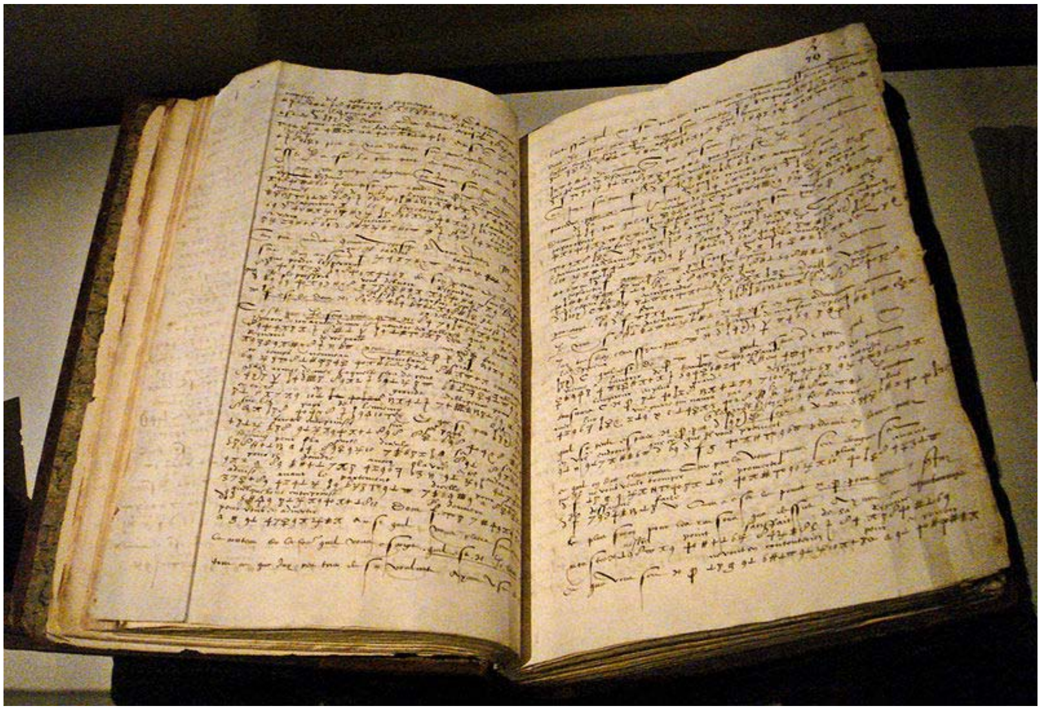
The earliest forms of secret writing required little more than local pen and paper analogs, as most people could not read. More literacy, or literate opponents, required actual cryptography. The main classical cipher types are transposition ciphers, which rearrange the order of letters in a message (e.g., 'hello world' becomes 'ehlol owrdl' in a trivially simple rearrangement scheme), and substitution ciphers, which systematically replace letters or groups of letters with other letters or groups of letters (e.g., 'fly at once' becomes 'gmz bu podf' by replacing each letter with the one following it in the Latin alphabet). Simple versions of either have never offered much confidentiality from enterprising opponents. An early substitution cipher was the Caesar cipher, in which each letter in the plaintext was replaced by a letter some fixed number of positions further down the alphabet. It was named after Julius Caesar who is reported to have used it, with a shift of 3, to communicate with his generals during his military campaigns, just like EXCESS-3 code in boolean algebra. There is record of several early Hebrew ciphers as well. The earliest known use of cryptography is some carved ciphertext on stone in Egypt (ca 1900 BC), but this may have been done for the amusement of literate observers. The next oldest is bakery recipes from Mesopotamia. Cryptography is recommended in the Kama Sutra as a way for lovers to communicate without inconvenient discovery.

The Greeks of Classical times are said to have known of ciphers (e.g., the scytale transposition cipher claimed to have been used by the Spartan military). Steganography (i.e., hiding even the existence of a message so as to keep it confidential) was also first developed in ancient times. An early example, from Herodotus, concealed a message—a tattoo on a slave's shaved head—under the regrown hair. Another Greek method was developed by Polybius (now called the "Polybius Square"). More modern examples of steganography include the use of invisible ink, microdots, and digital watermarks to conceal information.

Ciphertexts produced by a classical cipher (and some modern ciphers) always reveal statistical information about the plaintext, which can often be used to break them. After the discovery of frequency analysis perhaps by the Arab mathematician and polymath, Al-Kindi (also known as *Alkindus*), in the 9th century, nearly all such ciphers became more or less readily breakable by any informed attacker. Such classical ciphers still enjoy popularity today, though mostly as puzzles. Al-Kindi wrote a book on cryptography entitled *Risalah fi Istikhraj al-Mu'amma* (*Manuscript for the Deciphering Cryptographic Messages*), in which described the first cryptanalysis techniques.



16th-century book-shaped French cipher machine, with arms of Henri II of France



Enciphered letter from Gabriel de Luetz d'Aramon, French Ambassador to the Ottoman Empire, after 1546, with partial decipherment

Essentially all ciphers remained vulnerable to cryptanalysis using the frequency analysis technique until the development of the polyalphabetic cipher, most clearly by Leon Battista Alberti around the year 1467, though there is some indication that it was already known to Al-Kindi. Alberti's innovation was to use different ciphers (i.e., substitution alphabets) for various parts of a message (perhaps for each successive plaintext letter at the limit). He also invented what was probably the first automatic cipher device, a wheel which implemented a partial realization of his invention. In the polyalphabetic Vigenère cipher, encryption uses a *key word*, which controls letter substitution depending on which letter of the key word is used. In the mid-19th century Charles Babbage showed that polyalphabetic ciphers of this type remained partially vulnerable to extended frequency analysis techniques.

Although frequency analysis is a powerful and general technique against many ciphers, encryption has still been often effective in practice; many a would-be cryptanalyst was unaware of the technique. Breaking a message without using frequency analysis essentially required knowledge of the cipher used and perhaps of the key involved, thus making espionage, bribery, burglary, defection, etc., more attractive approaches to the cryptanalytically uninformed. It was finally explicitly recognized in the 19th century that secrecy of a cipher's algorithm is not a sensible nor practical safeguard of message security; in fact, it was further realized that any adequate cryptographic scheme (including ciphers) should remain secure even if the adversary fully understands the cipher algorithm itself. Security of the key used should alone be sufficient for a good cipher to maintain confidentiality under an attack. This fundamental principle was first explicitly stated in 1883 by Auguste Kerckhoffs and is generally called Kerckhoffs' principle; alternatively and more bluntly, it was restated by Claude Shannon, the inventor of information theory and the fundamentals of theoretical cryptography, as *Shannon's Maxim*—'the enemy knows the system'.

Different physical devices and aids have been used to assist with ciphers. One of the earliest may have been the scytale of ancient Greece, a rod supposedly used by the Spartans as an aid for a transposition cipher. In medieval times, other aids were invented such as the cipher grille, which was also used for a kind of steganography. With the invention of polyalphabetic ciphers came more sophisticated aids such as Alberti's own cipher disk, Johannes Trithemius' tabula recta scheme, and Thomas Jefferson's multi-cylinder (not publicly known, and reinvented independently by Bazeries around 1900). Many mechanical encryption/decryption devices were invented early in the 20th century, and several patented, among them rotor machines—famously including the Enigma machine used by the German government and military from the late '20s and during World War II. The ciphers implemented by better quality examples of these machine designs brought about a substantial increase in cryptanalytic difficulty after WWI.

The computer era

The development of digital computers and electronics after WWII made possible much more complex ciphers. Furthermore, computers allowed for the encryption of any kind of data representable in any binary format, unlike classical ciphers which only encrypted written language texts; this was new and significant. Computer use has thus supplanted linguistic cryptography, both for cipher design and cryptanalysis. Many computer ciphers can be characterized by their operation on binary bit sequences (sometimes in groups or blocks), unlike classical and mechanical schemes, which generally manipulate traditional characters (i.e., letters and digits) directly. However, computers have also assisted cryptanalysis, which has compensated to some extent for increased cipher complexity. Nonetheless, good modern ciphers have stayed ahead of cryptanalysis; it is typically the case that use of a quality cipher is very efficient (i.e., fast and requiring few resources, such as memory or CPU capability), while breaking it requires an effort many orders of magnitude larger, and vastly larger than that required for any classical cipher, making cryptanalysis so inefficient and impractical as to be effectively impossible. Alternate methods of attack (bribery, burglary, threat, torture, ...) have become more attractive in consequence.

Credit card with smart-card capabilities. The 3-by-5-mm chip embedded in the card is shown, enlarged. Smart cards combine low cost and portability with the power to compute cryptographic algorithms.

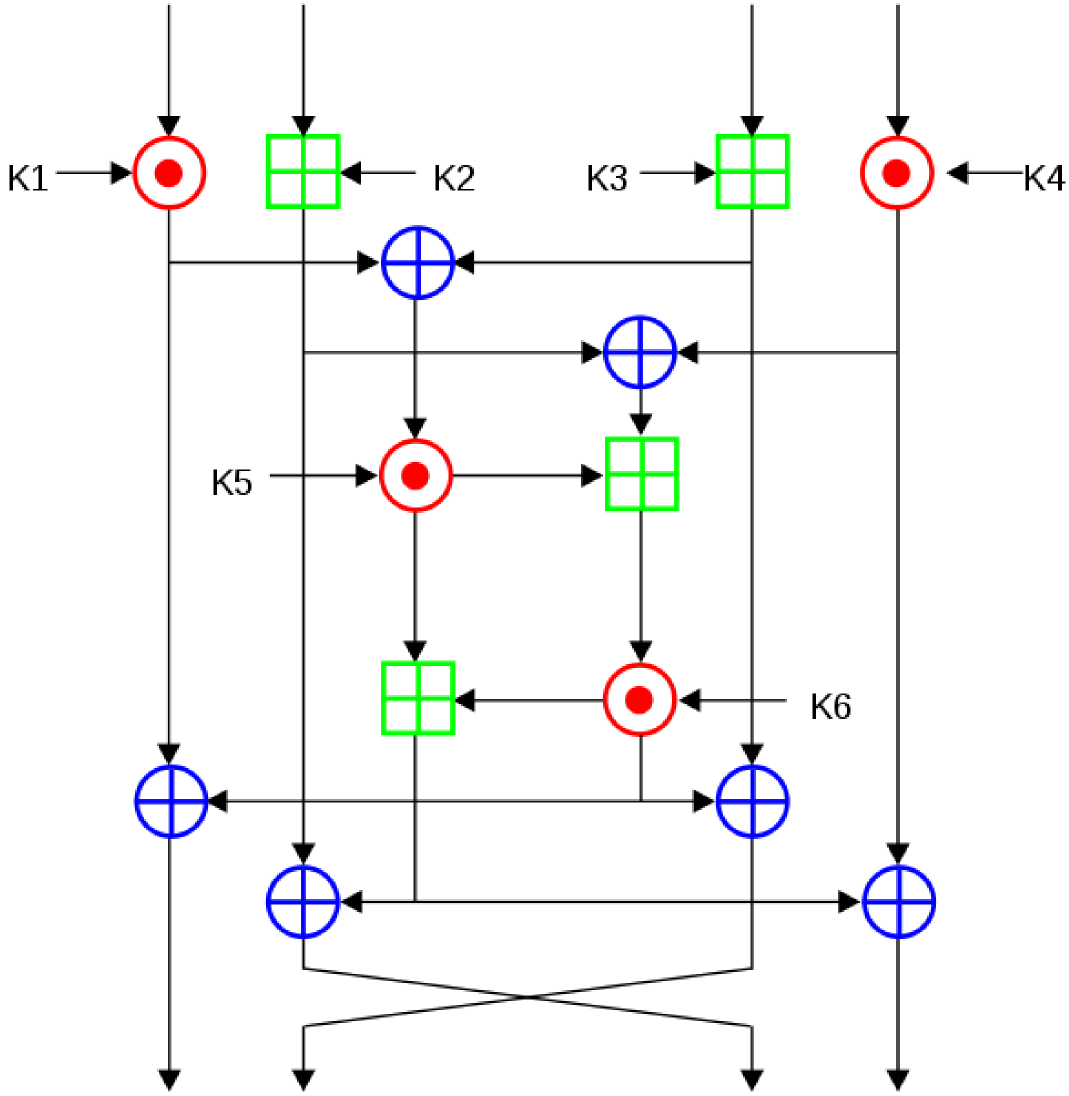
Extensive open academic research into cryptography is relatively recent; it began only in the mid-1970s. In recent times, IBM personnel designed the algorithm that became the Federal (i.e., US) Data Encryption Standard; Whitfield Diffie and Martin Hellman published their key agreement algorithm,; and the RSA algorithm was published in Martin Gardner's Scientific American column. Since then, cryptography has become a widely used tool in communications, computer networks, and computer security generally. Some modern cryptographic techniques can only keep their keys secret if certain mathematical problems are intractable, such as the integer factorization or the discrete logarithm problems, so there are deep connections with abstract mathematics. There are no absolute proofs that a cryptographic technique is secure (but see one-time pad); at best, there are proofs that some techniques are secure *if* some computational problem is difficult to solve, or this or that assumption about implementation or practical use is met.

As well as being aware of cryptographic history, cryptographic algorithm and system designers must also sensibly consider probable future developments while working on their designs. For instance, continuous improvements in computer processing power have increased the scope of brute-force attacks, thus when specifying key lengths, the required key lengths are similarly advancing. The potential effects of quantum computing are already being considered by some cryptographic system designers; the announced imminence of small implementations of these machines may be making the need for this preemptive caution rather more than merely speculative.

Essentially, prior to the early 20th century, cryptography was chiefly concerned with linguistic and lexicographic patterns. Since then the emphasis has shifted, and cryptography now makes extensive use of mathematics, including aspects of information theory, computational complexity, statistics, combinatorics, abstract algebra, number theory, and finite mathematics generally. Cryptography is, also, a branch of engineering, but an unusual one as it deals with active, intelligent, and malevolent opposition; other kinds of engineering (e.g., civil or chemical engineering) need deal only with neutral natural forces. There is also active research examining the relationship between cryptographic problems and quantum physics.

Symmetric-key cryptography

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 1976.



One round (out of 8.5) of the patented IDEA cipher, used in some versions of PGP for high-speed encryption of, for instance, e-mail

The modern study of symmetric-key ciphers relates mainly to the study of block ciphers and stream ciphers and to their applications. A block cipher is, in a sense, a modern embodiment of Alberti's polyalphabetic cipher: block ciphers take as input a block of plaintext and a key, and output a block of ciphertext of the same size. Since messages are almost always longer than a single block, some method of knitting together successive blocks is required. Several have been developed, some with better security in one aspect or another than others. They are the modes of operation and must be carefully considered when using a block cipher in a cryptosystem.

The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are block cipher designs which have been designated cryptography standards by the US government (though DES's designation was finally withdrawn after the AES was adopted). Despite its deprecation as an official standard, DES (especially its still-approved and much more secure triple-DES variant) remains quite popular; it is used across a wide range of applications, from ATM encryption to e-mail privacy and secure remote access. Many other block ciphers have been designed and released, with considerable variation in quality.

Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character-by-character, somewhat like the one-time pad. In a stream cipher, the output stream is created based on a hidden internal state which changes as the cipher operates. That internal state is initially set up using the secret key material. RC4 is a widely used stream cipher.

Cryptographic hash functions are a third type of cryptographic algorithm. They take a message of any length as input, and output a short, fixed length hash which can be used in (for example) a digital signature. For good hash functions, an attacker cannot find two messages that produce the same hash. MD4 is a long-used hash function which is now broken; MD5, a strengthened variant of MD4, is also widely used but broken in practice. The U.S. National Security Agency developed the Secure Hash Algorithm series of MD5-like hash functions: SHA-0 was a flawed algorithm that the agency withdrew; SHA-1 is widely deployed and more secure than MD5, but cryptanalysts have identified attacks against it; the SHA-2 family improves on SHA-1, but it isn't yet widely deployed, and the U.S. standards authority thought it "prudent" from a security perspective to develop a new standard to "significantly improve the robustness of NIST's overall hash algorithm toolkit." Thus, a hash function design competition is underway and meant to select a new U.S. national standard, to be called SHA-3, by 2012.

Message authentication codes (MACs) are much like cryptographic hash functions, except that a secret key can be used to authenticate the hash value upon receipt.

Public-key cryptography

Symmetric-key cryptosystems use the same key for encryption and decryption of a message, though a message or group of messages may have a different key than others. A significant disadvantage of symmetric ciphers is the key management necessary to use them securely. Each distinct pair of communicating parties must, ideally, share a different key, and perhaps each ciphertext exchanged as well. The number of keys required increases as the square of the number of network members, which very quickly requires complex key management schemes to keep them all straight and secret. The difficulty of securely establishing a secret key between two communicating parties, when a secure channel does not already exist between them, also presents a chicken-and-egg problem which is a considerable practical obstacle for cryptography users in the real world.

Whitfield Diffie and Martin Hellman, authors of the first published paper on public-key cryptography

In a groundbreaking 1976 paper, Whitfield Diffie and Martin Hellman proposed the notion of *public-key* (also, more generally, called *asymmetric key*) cryptography in which two different but mathematically related keys are used—a *public* key and a *private* key. A public key system is so constructed that calculation of one key (the 'private key') is computationally infeasible from the other (the 'public key'), even though they are necessarily related. Instead, both keys are generated secretly, as an interrelated pair. The historian David Kahn described public-key cryptography as "the most revolutionary new concept in the field since polyalphabetic substitution emerged in the Renaissance".

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. The *public key* is typically used for encryption, while the *private* or *secret key* is used for decryption. Diffie and Hellman showed that public-key cryptography was possible by presenting the Diffie–Hellman key exchange protocol.

In 1978, Ronald Rivest, Adi Shamir, and Len Adleman invented RSA, another public-key system.

In 1997, it finally became publicly known that asymmetric key cryptography had been invented by James H. Ellis at GCHQ, a British intelligence organization, and that, in the early 1970s, both the Diffie–Hellman and RSA algorithms had been previously developed (by Malcolm J. Williamson and Clifford Cocks, respectively).

The Diffie–Hellman and RSA algorithms, in addition to being the first publicly known examples of high quality public-key algorithms, have been among the most widely used. Others include the Cramer–Shoup cryptosystem, ElGamal encryption, and various elliptic curve techniques.

Padlock icon from the Firefox Web browser, meant to indicate a page has been sent in SSL or TLS-encrypted protected form. However, such an icon is not a guarantee of security; any subverted browser might mislead a user by displaying such an icon when a transmission is not actually being protected by SSL or TLS.

In addition to encryption, public-key cryptography can be used to implement digital signature schemes. A digital signature is reminiscent of an ordinary signature; they both have the characteristic that they are easy for a user to produce, but difficult for anyone else to forge. Digital signatures can also be permanently tied to the content of the message being signed; they cannot then be 'moved' from one document to another, for any attempt will be detectable. In digital signature schemes, there are two algorithms: one for *signing*, in which a secret key is used to process the message (or a hash of the message, or both), and one for *verification,* in which the matching public key is used with the message to check the validity of the signature. RSA and DSA are two of the most popular digital signature schemes. Digital signatures are central to the operation of public key infrastructures and many network security schemes (e.g., SSL/TLS, many VPNs, etc.).

Public-key algorithms are most often based on the computational complexity of "hard" problems, often from number theory. For example, the hardness of RSA is related to the integer factorization problem, while Diffie–Hellman and DSA are related to the discrete logarithm problem. More recently, *elliptic curve cryptography* has developed in which security is based on number theoretic problems involving elliptic curves. Because of the difficulty of the underlying problems, most public-key algorithms involve operations such as modular multiplication and exponentiation, which are much more computationally expensive than the techniques used in most block ciphers, especially with typical key sizes. As a result, public-key cryptosystems are commonly hybrid cryptosystems, in which a fast high-quality symmetric-key encryption algorithm is used for the message itself, while the relevant symmetric key is sent with the message, but encrypted using a public-key algorithm. Similarly, hybrid signature schemes are often used, in which a cryptographic hash function is computed, and only the resulting hash is digitally signed.

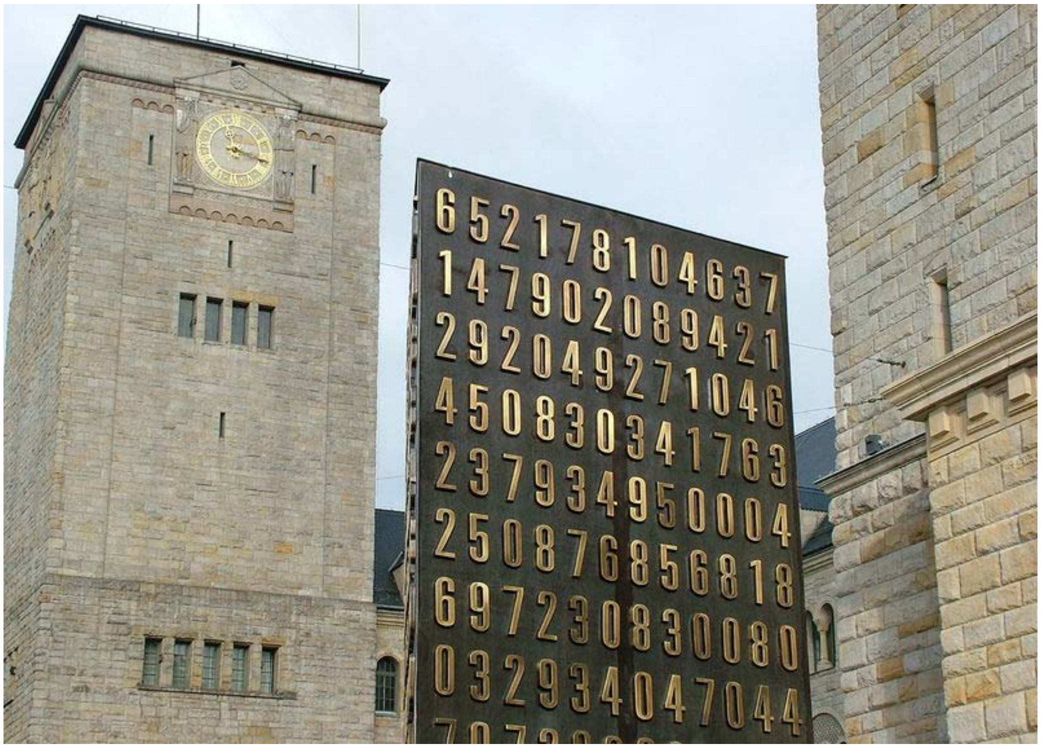
Cryptanalysis

Variants of the Enigma machine, used by Germany's military and civil authorities from the late 1920s through World War II, implemented a complex electro-mechanical polyalphabetic cipher. Breaking and reading of the Enigma cipher at Poland's Cipher Bureau, for 7 years before the war, and subsequent decryption at Bletchley Park, was important to Allied victory.

The goal of cryptanalysis is to find some weakness or insecurity in a cryptographic scheme, thus permitting its subversion or evasion.

It is a common misconception that every encryption method can be broken. In connection with his WWII work at Bell Labs, Claude Shannon proved that the one-time pad cipher is unbreakable, provided the key material is truly random, never reused, kept secret from all possible attackers, and of equal or greater length than the message. Most ciphers, apart from the one-time pad, can be broken with enough computational effort by brute force attack, but the amount of effort needed may be exponentially dependent on the key size, as compared to the effort needed to *use* the cipher. In such cases, effective security could be achieved if it is proven that the effort required (i.e., "work factor", in Shannon's terms) is beyond the ability of any adversary. This means it must be shown that no efficient method (as opposed to the time-consuming brute force method) can be found to break the cipher. Since no such showing can be made currently, as of today, the one-time-pad remains the only theoretically unbreakable cipher.

There are a wide variety of cryptanalytic attacks, and they can be classified in any of several ways. A common distinction turns on what an attacker knows and what capabilities are available. In a ciphertext-only attack, the cryptanalyst has access only to the ciphertext (good modern cryptosystems are usually effectively immune to ciphertext-only attacks). In a known-plaintext attack, the cryptanalyst has access to a ciphertext and its corresponding plaintext (or to many such pairs). In a chosen-plaintext attack, the cryptanalyst may choose a plaintext and learn its corresponding ciphertext (perhaps many times); an example is gardening, used by the British during WWII. Finally, in a chosen-ciphertext attack, the cryptanalyst may be able to *choose* ciphertexts and learn their corresponding plaintexts.



Poznań monument (*center*) to Polish cryptologists whose breaking of Germany's Enigma machine ciphers, beginning in 1932, altered the course of World War II

Cryptanalysis of symmetric-key ciphers typically involves looking for attacks against the block ciphers or stream ciphers that are more efficient than any attack that could be against a perfect cipher. For example, a simple brute force attack against DES requires one known plaintext and 2^{55} decryptions, trying approximately half of the possible keys, to reach a point at which chances are better than even the key sought will have been found. But this may not be enough assurance; a linear cryptanalysis attack against DES requires 2^{43} known plaintexts and approximately 2^{43} DES operations. This is a considerable improvement on brute force attacks.

Public-key algorithms are based on the computational difficulty of various problems. The most famous of these is integer factorization (e.g., the RSA algorithm is based on a problem related to integer factoring), but the discrete logarithm problem is also important. Much public-key cryptanalysis concerns numerical algorithms for solving these computational problems, or some of them, efficiently (i.e., in a practical time). For instance, the best known algorithms for solving the elliptic curve-based version of discrete logarithm are much more time-consuming than the best known algorithms for factoring, at least for problems of more or less equivalent size. Thus, other things being equal, to achieve an equivalent strength of attack resistance, factoring-based encryption techniques must use larger keys than elliptic curve techniques. For this reason, public-key cryptosystems based on elliptic curves have become popular since their invention in the mid-1990s.

While pure cryptanalysis uses weaknesses in the algorithms themselves, other attacks on cryptosystems are based on actual use of the algorithms in real devices, and are called *side-channel attacks*. If a cryptanalyst has access to, for example, the amount of time the device took to encrypt a number of plaintexts or report an error in a password or PIN character, he may be able to use a timing attack to break a cipher that is otherwise resistant to analysis. An attacker might also study the pattern and length of messages to derive valuable information; this is known as traffic analysis, and can be quite useful to an alert adversary. Poor administration of a cryptosystem, such as permitting too short keys, will make any system vulnerable, regardless of other virtues. And, of course, social engineering, and other attacks against the personnel who work with cryptosystems or the messages they handle (e.g., bribery, extortion, blackmail, espionage, torture, ...) may be the most productive attacks of all.

Cryptographic primitives

Much of the theoretical work in cryptography concerns cryptographic *primitives*—algorithms with basic cryptographic properties—and their relationship to other cryptographic problems. More complicated cryptographic tools are then built from these basic primitives. These primitives provide fundamental properties, which are used to develop more complex tools called *cryptosystems* or *cryptographic protocols*, which guarantee one or more high-level security properties. Note however, that the distinction between cryptographic *primitives* and cryptosystems, is quite arbitrary; for example, the RSA algorithm is sometimes considered a cryptosystem, and sometimes a primitive.

Typical examples of cryptographic primitives include pseudorandom functions, one-way functions, etc.

Cryptosystems

One or more cryptographic primitives are often used to develop a more complex algorithm, called a cryptographic system, or *cryptosystem*. Cryptosystems (e.g. El-Gamal encryption) are designed to provide particular functionality (e.g. public key encryption) while guaranteeing certain security properties (e.g. chosen-plaintext attack (CPA) security in the random oracle model). Cryptosystems use the properties of the underlying cryptographic primitives to support the system's security properties. Of course, as the distinction between primitives and cryptosystems is somewhat arbitrary, a sophisticated cryptosystem can be derived from a combination of several more primitive cryptosystems. In many cases, the cryptosystem's structure involves back and forth communication among two or more parties in space (e.g., between the sender of a secure message and its receiver) or across time (e.g., cryptographically protected backup data). Such cryptosystems are sometimes called *cryptographic protocols*.

Some widely known cryptosystems include RSA encryption, Schnorr signature, El-Gamal encryption, PGP, etc. More complex cryptosystems include electronic cash systems, signcryption systems, etc. Some more 'theoretical' cryptosystems include interactive proof systems, (like zero-knowledge proofs,), systems for secret sharing, etc.

Until recently, most security properties of most cryptosystems were demonstrated using empirical techniques, or using ad hoc reasoning. Recently, there has been considerable effort to develop formal techniques for establishing the security of cryptosystems; this has been generally called *provable security*. The general idea of provable security is to give arguments about the computational difficulty needed to compromise some security aspect of the cryptosystem (i.e., to any adversary).

The study of how best to implement and integrate cryptography in software applications is itself a distinct field; see: Cryptographic engineering and Security engineering.

Legal issues

Prohibitions

Cryptography has long been of interest to intelligence gathering and law enforcement agencies. Actually secret communications may be criminal or even treasonous; those whose communications are open to inspection may be less likely to be either. Because of its facilitation of privacy, and the diminution of privacy attendant on its prohibition, cryptography is also of considerable interest to civil rights supporters. Accordingly, there has been a history of controversial legal issues surrounding cryptography, especially since the advent of inexpensive computers has made widespread access to high quality cryptography possible.

In some countries, even the domestic use of cryptography is, or has been, restricted. Until 1999, France significantly restricted the use of cryptography domestically, though it has relaxed many of these. In China, a license is still required to use cryptography. Many countries have tight restrictions on the use of cryptography. Among the more restrictive are laws in Belarus, Kazakhstan, Mongolia, Pakistan, Singapore, Tunisia, and Vietnam.

In the United States, cryptography is legal for domestic use, but there has been much conflict over legal issues related to cryptography. One particularly important issue has been the export of cryptography and cryptographic software and hardware. Probably because of the importance of cryptanalysis in World War II and an expectation that cryptography would continue to be important for national security, many Western governments have, at some point, strictly regulated export of cryptography. After World War II, it was illegal in the US to sell or distribute encryption technology overseas; in fact, encryption was designated as auxiliary military equipment and put on the United States Munitions List. Until the development of the personal computer, asymmetric key algorithms (i.e., public key techniques), and the Internet, this was not especially problematic. However, as the Internet grew and computers became more widely available, high quality encryption techniques became well-known around the globe. As a result, export controls came to be seen to be an impediment to commerce and to research.

Export controls

In the 1990s, there were several challenges to US export regulations of cryptography. One involved Philip Zimmermann's Pretty Good Privacy (PGP) encryption program; it was released in the US, together with its source code, and found its way onto the Internet in June 1991. After a complaint by RSA Security (then called RSA Data Security, Inc., or RSADSI), Zimmermann was criminally investigated by the Customs Service and the FBI for several years. No charges were ever filed, however. Also, Daniel Bernstein, then a graduate student at UC Berkeley, brought a lawsuit against the US government challenging some aspects of the restrictions based on free speech grounds. The 1995 case Bernstein v. United States ultimately resulted in a 1999 decision that printed source code for cryptographic algorithms and systems was protected as free speech by the United States Constitution.

In 1996, thirty-nine countries signed the Wassenaar Arrangement, an arms control treaty that deals with the export of arms and "dual-use" technologies such as cryptography. The treaty stipulated that the use of cryptography with short key-lengths (56-bit for symmetric encryption, 512-bit for RSA) would no longer be export-controlled. Cryptography exports from the US are now much less strictly regulated than in the past as a consequence of a major relaxation in 2000; there are no longer very many restrictions on key sizes in US-exported mass-market software. In practice today, since the relaxation in US export restrictions, and because almost every personal computer connected to the Internet, everywhere in the world, includes US-sourced web browsers such as Mozilla Firefox or Microsoft Internet Explorer, almost every Internet user worldwide has access to quality cryptography (i.e., when using sufficiently long keys with properly operating and unsubverted software, etc.) in their browsers; examples are Transport Layer Security or SSL stack. The Mozilla Thunderbird and Microsoft Outlook E-mail client programs similarly can connect to IMAP or POP servers via TLS, and can send and receive email encrypted with S/MIME. Many Internet users don't realize that their basic application software contains such extensive cryptosystems. These browsers and email programs are so ubiquitous that even governments whose intent is to regulate civilian use of cryptography generally don't find it practical to do much to control distribution or use of cryptography of this quality, so even when such laws are in force, actual enforcement is often effectively impossible.

NSA involvement

Another contentious issue connected to cryptography in the United States is the influence of the National Security Agency on cipher development and policy. NSA was involved with the design of DES during its development at IBM and its consideration by the National Bureau of Standards as a possible Federal Standard for cryptography. DES was designed to be resistant to differential cryptanalysis, a powerful and general cryptanalytic technique known to NSA and IBM, that became publicly known only when it was rediscovered in the late 1980s. According to Steven Levy, IBM rediscovered differential cryptanalysis, but kept the technique secret at NSA's request. The technique became publicly known only when Biham and Shamir re-rediscovered and announced it some years later. The entire affair illustrates the difficulty of determining what resources and knowledge an attacker might actually have.

Another instance of NSA's involvement was the 1993 Clipper chip affair, an encryption microchip intended to be part of the Capstone cryptography-control initiative. Clipper was widely criticized by cryptographers for two reasons. The cipher algorithm was then classified (the cipher, called Skipjack, though it was declassified in 1998 long after the Clipper initiative lapsed). The secret cipher caused concerns that NSA had deliberately made the cipher weak in order to assist its intelligence efforts. The whole initiative was also criticized based on its violation of Kerckhoffs' principle, as the scheme included a special escrow key held by the government for use by law enforcement, for example in wiretaps.

Digital rights management

Cryptography is central to digital rights management (DRM), a group of techniques for technologically controlling use of copyrighted material, being widely implemented and deployed at the behest of some copyright holders. In 1998, American President Bill Clinton signed the Digital Millennium Copyright Act (DMCA), which criminalized all production, dissemination, and use of certain cryptanalytic techniques and technology (now known or later discovered); specifically, those that could be used to circumvent DRM technological schemes. This had a noticeable impact on the cryptography research community since an argument can be made that *any* cryptanalytic research violated, or might violate, the DMCA. Similar statutes have since been enacted in several countries and regions, including the implementation in the EU Copyright Directive. Similar restrictions are called for by treaties signed by World Intellectual Property Organization member-states.

The United States Department of Justice and FBI have not enforced the DMCA as rigorously as had been feared by some, but the law, nonetheless, remains a controversial one. Niels Ferguson, a well-respected cryptography researcher, has publicly stated that he will not release some of his research into an Intel security design for fear of prosecution under the DMCA. Both Alan Cox (longtime number 2 in Linux kernel development) and Professor Edward Felten (and some of his students at Princeton) have encountered problems related to the Act. Dmitry Sklyarov was arrested during a visit to the US from Russia, and jailed for five months pending trial for alleged violations of the DMCA arising from work he had done in Russia, where the work was legal. In 2007, the cryptographic keys responsible for Blu-ray and HD DVD content scrambling were discovered and released onto the Internet. In both cases, the MPAA sent out numerous DMCA takedown notices, and there was a massive internet backlash triggered by the perceived impact of such notices on fair use and free speech.