chapter 1 network security overview

1.1 network security policy – what is it?

A network security policy defines guidelines for computer network access, determines policy enforcement, and lays out the architecture of the organization's network security environment and defines how the security policies are implemented throughout the network architecture.

Network security policies describes an organization's security controls. It aims to keep malicious users out while also mitigating risky users within your organization. The initial stage to generate a policy is to understand what information and services are available, and to whom, what the potential is for damage, and what protections are already in place. [NS-WIT]

1.2 the pillars of information security

When we talk about information security, and, in particular, about network security, building a secure system which provides a protective environment for data administration requires course of action in several areas:

- confidentiality / access control
- data and information integrity
- availability
- authenticity
- non-repudiation

We'll have a more detailed look at these five pillars in the security services section.

1.3 security policies in general

In a broader sense, a **security policy** is a set of rules and practices dictating how sensitive information is managed, protected, and distributed. A security policy expresses exactly what the security level should be by setting the goals of what the security mechanisms are to accomplish.

This is an important element that has a major role in defining the design of the system. The security policy is a foundation for the specifications of a system, organization or other entity and provides the baseline for evaluating a system.

For an organization, it addresses the constraints on behavior of its members as well as constraints imposed on adversaries by mechanisms such as doors, locks, keys and walls. For systems, the security policy addresses constraints on functions and flow among them, constraints on access by external systems and adversaries including programs and access to data by people.

1.4 security models

While the security policy provides the abstract goals, the security model provides the dos and don'ts to achieve those goals.

If we relate to the five pillars of security listed above, some security models enforce rules to protect confidentiality, such as the Bell - LaPadula model. Other models enforce rules to protect integrity, such as the Biba model. Formal security models, such as Bell-LaPadula and Biba, are used to provide high assurance in security. Informal models, such as Clark-Wilson, are used more as a framework to describe how security policies should be expressed and executed. [CISSP]

A security policy outlines goals with no idea of how they would be accomplished. A model is a framework that gives the policy form and solves security problems for particular situations. Several security models have been developed to enforce security policies. We list below the most important ones and they will be detailed in the next chapter of this course:

- **State machine** models: no matter what state I am in, I am always trustworthy
- **Bell-LaPadula** model: I don’t want anyone to know my secrets – a mathematical model of a multilevel security system
- **Biba** model: protects the integrity of the information within a system
- **Clark-Wilson** model: protects the integrity of information by focusing on preventing authorized users from making unauthorized modification of data, or commit fraud and errors within commercial applications.
- **Information flow** model: Information is restricted in its flow to only go to and from entities in a way that does not negate the security policy.
- **Noninterference** model: Commands and activities performed at one security level should not be seen or affect subjects or objects at a different security level.
- **Brewer and Nash** model: allows for dynamically changing access controls that protect against conflicts of interest
- **Graham-Denning** model: creates rights for subjects, which correlate to the operations that can be execute on objects
- **Harrison-Ruzo-Ullman** model: allows for access rights to be changed and specifies how subjects and objects should be created and deleted.

There are three types of approaches in defining a security model:

- security by obscurity
- the perimeter defense model
- the defense in depth model

Security by obscurity relies on stealth for protection. The concept behind this model is that if no one knows that a network or system is there, then it won't be subject to attack. The basic hope is that hiding a network or at least not advertising its existence will serve as sufficient security. The problem with this approach is that it never works in the long term, and once detected, a network is completely vulnerable.

1.5 vulnerabilities, threats and attacks

Symmetric

1.6 vulnerabilities

A vulnerability

1.7 threats

Threat types:

- int

1.8 attacks

Attacks on the security of a system can be classified roughly into two categories:

- passive attacks
- active attacks

Passive attacks have the goal of intercepting and/or monitoring the traffic. There are two types of passive attacks – message interception and traffic analysis.

Active attacks imply modifying the normal flow of informations and/or creating illegitimate data transmissions

If we classify the attacks by the nature (scope) of these attacks, we can distinguish the following types:

- interceptions – an unauthorized entity gains illegitimate access to a network
- traffic analysis - the process of intercepting and examining messages in order to deduce information from patterns in communication
- interruptions – an element of the system is destroyed/incapacitated and becomes unusable or with considerable reduced capacity
- masquerading – requires an attacker to have the ability to both monitor and alter or inject messages into a communication channel
- alterations – an unauthorized entity may change the content of a data file or of a message exchanged over the network
- construction - an unauthorized entity may create and transmit false messages or may add new records in a data file or in a database

1.9 security services

A security system is supposed to provide the following services:

- authentication
- integrity
- confidentiality
- non-repudiation
- access control
- availability

These services are closed related to the five pillars of information security specified in the section 1.2.

Authentication allows the recipient of the message to validate the identity of the sender. It prevents an unauthorized entity to masquerade itself as a legitimate sender of the message.

Integrity guarantees that the message sent has not been modified or altered along the communication channel. This is usually accomplished by attaching to the message itself a digest (compressed version) of fixed length of the message, digest which allows verify if the original message was (intentionally or not) altered.

Confidentiality (secrecy) prevents unauthorized entities from accessing the real content of a message.

Non-repudiation confirms the delivery of data to the sender. The receiver can also verify the identity of the sender of the information. Between the two agents, no one can deny the sending or receiving of the data.

It means there should be some form of audibility. The information security system provides logs which can be opened to provide proof of who sent and received the data. [PIL-SEC]

Access control is a mechanism which allows access to resources based on explicit permissions assigned to different entities.

Availability of information means that qualified people who are granted access to the system can get the information any time they like with no fail. It can be enabled by having a robust framework making up the IT infrastructure. It ensures that the system remains fully functional even during adverse situations like database fall overs. Having excellent resources ensures that information can Itbe accessed in a comfortable and timely manner. The typical way of ensuring the availability of data is by having load balancers which provide non-failure of server resources. [PIL-SEC]

chapter 2 security models

2.1 Par 1

A **cry**

Cipher

2.2 Par 2

2.2.1 Subpar 1

chapter 3 network security standards

3.1 different security standards

While information security plays an important role in protecting the data and assets of an organisation, we often hear news about security incidents, such as defacement of websites, server hacking and data leakage. Organizations need to be fully aware of the need to devote more resources to the protection of information assets, and information security must become a top concern in both government and business.

To address the situation, a number of governments and organisations have set up benchmarks, standards and in some cases, legal regulations on information security to help ensure an adequate level of security is maintained, resources are used in the right way,and the best security practices are adopted. Some industries, such as banking, are regulated, and the guidelines or best practices put together as part of those regulations often become a de facto standard among members of these industries.

- **ISO STANDARDS**

The International Organization for Standardization (ISO), established in 1947, is a non-governmental international body that collaborates with the International Electrotechnical Commission (IEC) and the International Telecommunication Union (ITU) on information and communications technology (ICT) standards. The following are commonly referenced ISO security standards:

ISO/IEC 27002:2005 (Code of Practice for Information Security Management) ISO/IEC 27002:2005 (replaced ISO/IEC 17799:2005 in April 2007) is an international standard that originated from the BS7799-1, one that was originally laid down by the British Standards Institute (BSI). ISO/IEC 27002:2005 refers to a code of practice for information security management, and is intended as a common basis and practical guideline for developing organizational security standards and effective management practices.

ISO/IEC 27033-1:2009 - network security overview and concepts (see 3.2)

ISO/IEC 27033-2:2012 - Guidelines for the design and implementation of network security (see 3.3)

ISO/IEC 27033-3:2010 - Reference networking scenarios - threats, design techniques and control issues (see 3.4)

ISO/IEC 27033-4:2014 - Securing communications between networks using security gateways (see 3.5)

ISO/IEC 27033-5:2013 - Securing communications across networks using Virtual Private Networks (VPNs) (see 3.6)

ISO/IEC 27033-6: Securing wireless IP network access (see 3.7)

- **FISMA**

FISMA stands for Federal Information Security Management Act, and is a part of the USE-Government Act (Public Law 107-347) that became legislation in 2002 29. It requires US federal agencies to develop, document, and implement an agency-wide programme to provide information security for the information (and information systems) that support the operations and assets of the agency. Some of the requirements include:

1. Periodic risk assessments of information and information systems that support the operations and assets of the organization;
2. Risk-based policies and procedures designed to reduce information security risks to an acceptable level.
3. Plans for providing adequate security for networks and information systems;
4. Security awareness training to all personnel, including contractors;

5. Periodic evaluation and testing of the effectiveness of the security policies,procedures and controls. The frequency should not be less than annually. Remedial action to address any deficiencies found to be properly managed;
6. A working and tested security incident handling procedure;
7. A business continuity plan in place to support the operation of the organization.

- **FIPS**

The Federal Information Processing Standards (FIPS) Publication Series of the National Institute of Standards and Technology (NIST) is an official series of publications relating to standards and guidelines adopted and made available under the provisions of the FISMA 30. FIPS Publication 199, Standards for Security Categorization of Federal Information and Information Systems, is the first mandatory security standard laid down under the FISMA legislation. FIPS Publication 200, entitled “Minimum Security Requirements for Federal Information and Information Systems” is the second mandatory set of security standards that specify minimum security requirements for US federal information and information systems across 17 security-related areas. US federal agencies must meet the minimum security requirements defined in this standard by selecting appropriate security controls and assurance requirements laid down in NIST Special Publication 800-53 (Recommended Security Controls for Federal Information Systems).

The 17 security-related areas include:

1. access control;
2. awareness and training;
3. audit and accountability;
4. certification, accreditation, and security assessments;
5. configuration management;
6. contingency planning;
7. identification and authentication;
8. incident response;
9. maintenance;
10. media protection;
11. physical and environmental protection;
12. planning;
13. personnel security;
14. risk assessment;
15. systems and services acquisition;
16. system and communications protection;
17. system and information integrity.

Different FIPS standards:

FIPS 140 Security requirements for cryptography modules

FIPS 153 (3D graphics)

FIPS 197 (Rijndael / AES cipher)

FIPS 199 Standards for Security Categorization of Federal Information and Information Systems

FIPS 201 Personal Identity Verification for Federal Employees and Contractors

3.2 ISO/IEC 27033-1:2009 - network security overview and concepts

Revised and replaced ISO/IEC 18028 part 1;

Provides a roadmap and overview of the concepts and principles underpinning the remaining parts of ISO/IEC 27033;

Objective: “to define and describe the concepts associated with, and provide management guidance on, network security. This includes the provision of an overview of network security and related definitions, and guidance on how to identify and analyze network security risks and then define network security requirements. It also introduces how to achieve good quality technical security architectures, and the risk, design and control aspects associated with typical network scenarios and network ‘technology’ areas (which are dealt with in detail in subsequent parts of ISO/IEC 27033). In effect it also provides an overview of the ISO/IEC 27033 series and a ‘road map’ to all other parts”;

Provides a glossary of information security terms specific to networking;

Provides guidance on a structured process to identify and analyze network security risks and hence define network security control requirements, including those mandated by relevant information security policies;

Provides an overview of the controls supporting network technical security architectures and related technical controls, as well as non-technical controls plus other technical controls that are not solely related to network security (thus linking to ISO/IEC 27001, ISO/IEC 27002 and ISO/IEC 27005 plus other ISO27k standards as they are released);

Explains good practices in respect of network technical security architectures, and the risk, design and control aspects associated with typical network scenarios and network technology areas (expanded in subsequent parts of ISO/IEC 27033 - see below);

Briefly addresses the issues associated with implementing and operating network security controls, and the ongoing monitoring and reviewing of their implementation;

Extends the security management guidelines provided in ISO/IEC TR 13335 and ISO/IEC 27002 etc. by detailing the specific operations and mechanisms needed to implement network security controls in a wider range of network environments, providing a bridge between general information security management issues and the specifics of implementing largely technical network security controls (e.g. firewalls, IDS/IPS, message integrity controls etc.);

Mentions requirements such as non-repudiation and reliability in addition to the classical CIA triad (confidentiality, integrity and availability);

Somehow manages to provide a reasonably technical overview of network security with barely any reference to the OSI network stack!;

76 pages long;

Status: part 1 was published in 2009. It is currently being revised. The revision has easily passed the vote at DIS stage and may be published in 2015. Status update Jan 6

3.3 ISO/IEC 27033-2:2012 - Guidelines for the design and implementation of network security

Revised and replaced ISO/IEC 18028 part 2;

Scope: planning, designing, implementing and documenting network security;

Objective: “to define how organizations should achieve quality network technical security architectures, designs and implementations that will ensure network security appropriate to their business environments, using a consistent approach to the planning, design and implementation of network security, as relevant aided by the use of models/frameworks. (In this context, a model/framework is used to outline a representation or description showing the structure and high level workings of a type of technical security architecture/design)” [quoted from the FCD of 27033-1];

Defines a network security architecture for providing end-to-end network security. The architecture can be applied to various kinds of networks where end-to-end security is a concern and independently of the network's underlying technology;

Serves as a foundation for detailed recommendations on end-to-end network security;

Covers risks, design, techniques and control issues;

Refers forward to later parts of ISO/IEC 27033 for more specific guidance.

Status: part 2 was published in 2012.

3.4 ISO/IEC 27033-3:2010 - Reference networking scenarios - threats, design techniques and control issues

Objective is “to define the specific risks, design techniques and control issues associated with typical network scenarios” [quoted from the FCD of 27033-1];

Discusses threats, specifically, rather than all the elements of risk;

Refers to other parts of ISO/IEC 27033 for more specific guidance;

Status: part 3 was published in 2010.

3.5 ISO/IEC 27033-4:2014 - Securing communications between networks using security gateways

Revision of ISO/IEC 18028 part 3 and possibly ISO/IEC 18028 part 4;

Provides an overview of security gateways through a description of different architectures;

Guideline on securing communications between networks through gateways, firewalls, application firewalls, Intrusion Protection System [sic] etc. in accordance with a policy, including identifying and analysing network security threats, defining security control requirements, and designing, implementing, operating, monitoring and reviewing the controls;

Outlines how security gateways analyze and control network traffic through:

- Packet filtering;
- Stateful packet inspection;
- Application proxy (application firewalls);
- Network address translation NAT;
- Content analysis and filtering;

Guides the selection and configuration of security gateways, choosing the right type of architecture for a security gateway which best meets the security requirements of an organization;

Refers to various kinds of firewall as examples of security gateways. [Firewall is a commonplace term of art that is curiously absent from ISO/IEC 27000, ISO/IEC 27002 and is not defined explicitly in this standard either];

Status: part 4 was published in 2014.

3.6 ISO/IEC 27033-5:2013 - Securing communications across networks using Virtual Private Networks (VPNs)

Revision of ISO/IEC 18028 part 5;

Purpose: to provide “guidelines for the selection, implementation and monitoring of the technical controls necessary to provide network security using Virtual Private Network (VPN) connections to interconnect networks and connect remote users to networks”;

Extends the IT security management guidelines of ISO/IEC TR 13335 by detailing the specific operations and mechanisms needed to implement network security safeguards and controls in a wider range of network environments, providing a bridge between general IT security management issues and network security technical implementations;

Provides guidance for securing remote access over public networks;

Gives a high-level, incomplete assessment of the threats to VPNs (i.e. it mentions the threats of intrusion and denial of service but not unauthorized monitoring/interception, traffic analysis, data corruption, insertion of bogus traffic, various attacks on VPN end points, malware, masquerading/identity theft, insider threats etc., although these are mentioned or at least hinted-at later under security requirements);

Introduces different types of remote access including protocols, authentication issues and support when setting up remote access securely;

Intended to help network administrators and technicians who plan to make use of this kind of connection or who already have it in use and need advice on how to set it up securely and operate it securely;

Status: part 5 was published in 2013.

3.7 ISO/IEC 27033-6: Securing wireless IP network access

Objective: “to define the specific risks, design techniques and control issues for securing wireless and radio networks” [quoted from the FCD of 27033-1];

This is a generic wireless network security standard offering basic advice;

WD3 lists a number of “threats” which are, in fact, attack modes or risks;

WD3 repeatedly refers to “access network”, a curious term that is not defined. It seems to mean “network” but without a definition, I cannot tell for sure;

WD3 indicates that encryption is a confidentiality and integrity control, whereas normally other cryptographic controls and protocols provide the integrity function, not encryption as such;

Status: at 2nd CD stage. The completed standard may surface by the end of 2015 but a request has been made to extend the project.

chapter 4 security policies implementations

4.1 introduction

Information security has come to play an extremely vital role in today's fast moving, but invariably technically fragile business environment. Consequently, secured communications are needed in order for both companies and customers to benefit from the advancements that the Internet is empowering us with.

The importance of this fact needs to be clearly highlighted so that adequate measures will be implemented, not only enhancing the company's daily business procedures and transactions, but also to ensure that the much needed security measures are implemented with an acceptable level of security competency.

It is sad to see that the possibility of having your company's data exposed to a malicious attacker is constantly increasing nowadays due to the high number of "security illiterate" staff also having access to sensitive, and sometimes even secret business information. Just imagine the security implications of someone in charge of sensitive company data, browsing the Internet insecurely through the company's network, receiving suspicious e-mails containing various destructive attachments, and let's not forget the significant threats posed by the constant use of any Instant Messaging (IM) or chat applications.

4.2 why have a security policy?

As building a good security policy provides the foundations for the successful implementation of security related projects in the future, this is without a doubt the first measure that must be taken to reduce the risk of unacceptable use of any of the company's information resources.

The first step towards enhancing a company's security is the introduction of a precise yet enforceable security policy, informing staff on the various aspects of their responsibilities, general use of company resources and explaining how sensitive information must be handled. The policy will also describe in detail the meaning of acceptable use, as well as listing prohibited activities.

The development (and the proper implementation) of a security policy is highly beneficial as it will not only turn all of your staff into participants in the company's effort to secure its communications but also help reduce the risk of a potential security breach through "human-factor" mistakes. These are usually issues such as revealing information to unknown (or unauthorised sources), the insecure or improper use of the Internet and many other dangerous activities.

Additionally the building process of a security policy will also help define a company's critical assets, the ways they must be protected and will also serve as a centralised document, as far as protecting Information Security Assets is concerned.

4.3 what is a security policy?

The security policy is basically a plan, outlining what the company's critical assets are, and how they must (and can) be protected. Its main purpose is to provide staff with a brief overview of the "acceptable use" of any of the Information Assets, as well as to explain what is deemed as allowable and what is not, thus engaging them in securing the company's critical systems.

The document acts as a "must read" source of information for everyone using in any way systems and resources defined as potential targets. A good and well developed security policy should address some of these following elements:

- How sensitive information must be handled
- How to properly maintain your ID(s) and password(s), as well as any other accounting data

- How to respond to a potential security incident, intrusion attempt, etc.
- How to use workstations and Internet connectivity in a secure manner
- How to properly use the corporate e-mail system

Basically, the main reasons behind the creation of a security policy is to set a company's information security foundations, to explain to staff how they are responsible for the protection of the information resources, and highlight the importance of having secured communications while doing business online.

4.4 getting started

The purpose of this section is to provide you with possible strategies and some recommendations for the process of creating a security policy, and to give you a basic plan of approach while building the policy framework.

The start procedure for building a security policy requires a complete exploration of the company network, as well as every other critical asset, so that the appropriate measures can be effectively implemented. Everything starts with identifying the company's critical informational resources, a subject that is discussed in depth in the next section of the paper.

4.5 risk analysis (identifying the assets)

As in any other sensitive procedure, Risk Analysis and Risk Management play an essential role in the proper functionality of the process. Risk Analysis is the process of identifying the critical information assets of the company and their use and functionality -- an important (key) process that needs to be taken very seriously. Essentially, it is the very process of defining exactly WHAT you are trying to protect, from WHOM you are trying to protect it and most importantly, HOW you are going to protect it.

In order to be able to conduct a successful Risk Analysis, you need to get well acquainted with the ways a company operates; if applicable, the ways of working and certain business procedures, which information resources are more important than others (prioritizing), and identifying the devices / procedures that could lead to a possible security problem.

List everything that is essential for the proper functionality of the business processes; like key applications and systems, application servers, web servers, database servers, various business plans, projects in development, etc.

A basic approach would be:

- Identify what you're trying to protect
- Look at whom you're trying to protect it from
- Define what the potential risks are to any of your Information Assets
- Consider monitoring the process continually in order to be up to date with the latest security weaknesses
- A possible list of categories to look at would be:
  - Hardware: All servers, workstations, personal computers, laptops, removable media (CD's, floppies, tapes, etc.), communication lines, etc.
  - Software: Identify the risks of a potential security problem due to outdated software, infrequent patches and updates to new versions, etc. Also take into account the potential issues with staff installing various file sharing apps (Kazaa, Sharereactor, E-Donkey, etc.), IM (chat) software, entertainment or freeware software coming from unknown and untrustworthy sources.
  - Personnel: Those who have access to confidential information, sensitive data, those who "own", administer or in any way modify existing databases.

4.6 risk management (identifying the threats)

Based on the research conducted on the company's information assets, you should now be able to properly manage all the threats posed by each of your resources.

The purpose of this section is to guide you through the creation of a list outlining various potential threats, something that should also be included in the formal security policy. Each of the following elements will be discussed in depth later in the Security Awareness Program section, thus providing the staff members with a better understanding of each of the topics covered below.

4.6.1 physical/desktop security

- **System Access**: best practices for password creation, passwords aging, minimum password length, characters to be included while choosing passwords, password maintenance, tips for safeguarding (any) accounting data; the dangers to each of these issues must be explained in the security awareness program;
- **Virus Protection**: best practices for malicious code protection, how often the system should be scanned, how often, if not automatically, should Live Update of the software database be done, tips for protection against (any) malicious code(viruses/trojans/worms);
- **Software Installation**: is freeware software forbidden, if allowed, under what conditions, how is software piracy tolerated, are entertainment/games allowed or completely prohibited as well the installation of any other program coming from unknown and untrustworthy sources;
- **Removable Media(CD's, floppy)**: "Acceptable Use" measures (perhaps by way of a AUP – Acceptable Use Policy) need to be established, the dangers of potential malicious code entering the company network or any other critical system need to be explained as well;
- **Encryption**: explain when, how and who must encrypt any of the company's data;
- **System Backups**: the advantage of having backups needs to be explained; who is responsible, and how often should the data be backed up;
- **Maintenance**: the risks of a potential physical security breach need to be briefly explained;
- **Incident Handling**: define what a suspicious event is, to whom it needs to be reported, and what further steps need to be taken;

4.6.2 internet threats

- **Web Browsing**: define what constitutes restricted, forbidden and potentially malicious web sites, provide staff members with brief, and well summarized tips for safer browsing, additionally let them know that their Internet usage is strictly monitored in order to protect company's internal systems;
- **E-mail Use**: define the "acceptable use" criteria of the E-mail system, what is allowed and what is not, the company policy on using the mail system for personal messages, etc. Also briefly explain the potential threats posed by (abusing) the mail system and of the potential problems as far as spreading malicious code is concerned;
- **Instant Messaging (IM) Software (ICQ, AIM, MSN, etc.)**: whether it is allowed or completely forbidden, provide them with short examples of how an attacker might use these programs to penetrate and steal/corrupt/modify company data;
- **Downloading/Attachments**: is downloading allowed or not, useful tips for safer downloading, explanation of trusted and untrustworthy sources, best practices for mail attachments if allowed, discussion of potential threats and dangers, use of virus scanners, etc.

These elements will later be covered in detail in a Security Awareness Program. Staff need to understand why some activities are prohibited, what the impact of certain dangers can have on the company, actions they must follow if and when a potential security problem has been suspected or discovered. By involving staff in a Security Awareness Program staff will not just broaden their knowledge on the information security field, but also learn how to act in a secure manner while using any of the company's information assets.

4.7 security policy violation

In order to realise the importance of a security policy, staff need to be aware and fully understand the consequences of violating the policy, thereby exposing critical systems to a malicious attacker, or causing unintended damage to other companies worldwide. Violations should be handled accordingly; those who in one way or the other violate the security policy should be made aware that they may face being put through a "trial period", which involves also the limited use of some of the company information assets until they can show they are able to act in a secure manner while using the corporate systems. They should also be aware that in some (severe) cases they also may risk being fired or even prosecuted.

Whereas this may seem as overkill to some, appropriate action needs to be taken in every violation case in accordance with the terms of the AUP and the policy, with the focus on reiterating the security basics and not punishment. Otherwise there will most likely be a successful penetration, either due to human error, or misunderstanding the policy.

4.8 the implementation of the policy

When the security policy is all drawn up, revised, updated and agreed upon, the implementation process will follow. This is usually harder than the creation of the policy itself, due the fact that at this stage you also need to coach and educate your staff to behave in a "secure" manner, following each of the core elements pointed in the formal security policy.

The final version of the security policy must be made available to all of your employees having access to any of your information assets. The policy must be easily obtainable at any time, with a copy placed on the internal network and intranet, if applicable.

A proper implementation requires not only educating staff on each of the core elements flagged as critical in the formal Security Policy, but also changing their role in the effort to protect critical company data.

The next section will aim to guide you through the creation process of a basic Security Awareness Program, along with various innovative and interesting ways of educating your staff, using user-friendly & informal lines of communication between the Information Security Office (ISO) members and your employees.

4.9 the process of developing

This section will provide you with the various strategies of building a solid Security Awareness Program. We will discuss various methods, their advantages and disadvantages, and will also give you get a better understanding of the essential steps to building the Program.

At the beginning you must answer yourself the following questions:

- What is the Security Awareness Program supposed to accomplish, and how are you going to draw attention on that?
- Who is your audience, how "educated" they are; is it going to be necessary to divide the program into two parts, one for those who have more knowledge about computers, and one for those who are not much into computers at all?
- How are you going to reach and motivate your audience? More importantly, how are you going to get your audience interested in improving the Information Assets of the company?
- Is the Program going to rely on a formal or an informal way of communication between you and the staff members? In which way are you going to conduct and present it?

4.9.1 the purpose of the program

First of all, you need to explain to staff what the program will be trying to accomplish, how it will aim to improve the operations of the company, and how vital the protection of Information Assets really is. You will need to explain why "Security is everyone's responsibility", and ensure everybody understands it; explain that even if the company has the latest technological improvements like firewalls, intrusion detection systems, etc., an uneducated staff member could easily endanger sensitive information, and render any technical security measure in place, completely and utterly useless.

Another common misunderstanding that you will definitely face while conducting the Program is that the majority of people often tend to think that it is not their responsibility to help improve the security of their company. Generally people are of the (wrong) opinion that only the IT department or Information Security Office(ISO) can and need to take care of issues like these, and that is where generally the buck stops.

4.9.2 addressing the audience

One major problem that I am sure you are going to be facing is the difference in the levels of computer skills (of your audience), which will sometimes force you to pay additional attention to those who are not that much into computers. On the other hand you could also choose to differentiate between those who need security education, and those who don't; the idea is to separate staff having access to any of the company information assets from those who don't (and can't endanger sensitive data in any way), as this will definitely save you a lot of time and resources. It would be a good approach to hold informal meetings with staff in order to talk on a personal level and also conduct several surveys in order to measure their skill level; this way you will know where to focus your attention to.

4.9.3 measuring their security awareness level through surveys

Security Awareness surveys are developed with the idea of measuring the current Awareness level of your staff, but will usually also point out common mistakes and misunderstandings of your employees; which will definitely help you improve the quality of the Program, even before it starts. It is highly recommended to archive the surveys in order to evaluate the effectiveness of the Program over a period of time.

You might also want to indicate to staff members that the survey is completely anonymous, that there is no need to cheat as the main idea is to merely measure the overall security awareness level in the company, and above all that this is just a survey and not an exam. They could answer just the main question without having to answer the "Why do they think so" section, if they don't know what to give here as an answer.

Some sample Security Measuring Survey questions might be:

1. Which of the following passwords is the most secure one, and why do you think so?
   - ○ Abc123456
   - ○ HerculeS
   - ○ HRE42pazoL
   - ○ $safe456TY

   Why do you think so?
2. Which is the most dangerous attachment extension, and why do you think so?
   - ○ *.exe
   - ○ *.com
   - ○ *.bat
   - ○ *.vbs
   - ○ all of the above

   Why do you think so?

3. Your security policy states that the Information Security Office (ISO) would never send you an update to an application, but you have just received one, what would you do next?
   - ○ as it's coming from security@company.com which is our ISO e-mail address I will just run it and have the latest version of the software.
   - ○ as stated in the Security Policy I need to scan all the attachments before running, so I will scan and run after that.
   - ○ I would call the ISO office immediately to request further information.
4. A friend of yours gave you a multimedia CD last night, which you intend to check from your workstation at work; how are you going to do it?
   - ○ he is a friend of mine, and he would never give me any destructive files like viruses, etc. I trust him/her that's why I am going to check it out right away.
   - ○ although he is a friend of mine, it is stated in the security policy that removable media is allowed but its use should be limited to the minimum; I will stick to that and would scan the CD contents and see what's inside before I do so.
   - ○ I would just check the contents of the CD from my personal PC.
5. An ISO office representative asks you (in person) for your password as they misplaced it, and would need it to implement further security measures on your workstation; what would you do?
   - ○ they can't access the workstation without my password, and as it is about improving security, I would give it to them, as they are those responsible for maintaining the security within the organization.
   - ○ I already have the workstation properly secured so I won't give it to them.
   - ○ I won't share my password with somebody even if my manager tries to force me into telling it; I would keep it as secret as possible.

These are some sample questions covering most of the threats pointed out in the Security Policy. It is completely up to you to decide how many questions should be in the survey as well as the aspects they should cover; but it is advisable to consider issuing surveys on a regular basis in order to continuously monitor the level and the effectiveness of the Program.

4.9.4 getting their attention

Staff already have a lot of things to think about, a lot of decisions to make, operate and run through most of the day-to-day business procedures; therefore you need to have a very good strategy to get them motivated & eager to learn how they can improve company security.

Everyone these days is interested in stories about computer security in one way or the other, especially the (high-profile) break-ins, and making use of this, your main aim will be to help understand "attendees" of the program that they are actually going to be the new "gatekeepers" of critical company data (the information assets). You will undoubtedly will get asked questions like "Yeah, it's great to contribute to company security, but what do I get in exchange", which I define as normal questions, to which you must give proper answers.

Your future "students" need to be made aware of and understand how expensive it is for a company to conduct Security Awareness Courses, and to employ security experts, in order to provide 'attainable' services to its customers. Explain to them the damages that could be inflicted to the company, to the company (brand) name, its image, etc., which will inevitably impact on them somehow in return.

On the other hand, draw their attention also to the personal benefits from the whole program and the value of all the knowledge that will be supplied to them. One good example is to mention how all that information will significantly help them increase the security level of their own personal computers at home. The information they will be provided with does not only apply for their PC's at work but applies (in full) for their home PC's as well.

Another important point to keep in mind is the different ways people learn and memorise things, or in other words, deal with information they have just been provided with. Some learn by reading the materials, while others learn more by looking at diagrams, although it is proven that a combination of these methods has maximum effect in the process of understanding the subject. Therefore, you must ensure that your presentation style is such that it appeals to a crowd of people with varying degrees of knowledge and understanding.

Everyone gets bored of reading long materials, no matter how interesting they might be; if there isn't a picture, diagram or anything that brings some sort of variety into the process, people leave it behind. Try to "visualise" every subject that you are talking about by adding plenty of pictures, diagrams, relevant artwork and cartoons.

Cartoons are especially good, as they add an element of humour; people will definitely remember a funny situation representing a far serious procedure. Cartoons are best suited for posters, and most effective when placed all over the company, with their main purpose being a friendly medium to spreading the security awareness program messages (i.e. "Lock your machine when you leave", or "don't share your ID and password with ANYONE", etc).

Humor plays an essential role in the friendly education of the staff members; consider using it as you see fit but don't turn the whole program into a big comedy where everyone laughs and just makes jokes about the word Security. The addition of a little, humorous anecdote to any of your lectures like "I've got a friend who's so paranoid about security that he burns every paper after work, but come on, don't set the fire alarms off, just shred those papers labeled confidential/secret" would do fine.

4.9.5 choosing the approach

There are several approaches that you can follow when educating staff, and this section will point out the one which I define as the best one; a combination of both formal and informal ways of education.

The advantage of the formal method is that it will help staff realize the very importance of the security issue, as they know that these presentations cost a fair amount of resources, effort and money. On the other hand it will highlight the fact that the company is taking security very serious and therefore taking very serious measures to protect its information assets by educating its staff; and all it requires from them is a little time, devotion and understanding the importance of the security issue.

Another highly beneficial point when conducting a formal Awareness Program is the fact that your message, tutorial, presentation will be spread between most, if not all of the staff members; you will reach a lot of people that way, which will save you a lot of time compared to methods like one-on-one sessions, etc.

The informal way of education consists of email reminders, discussions, posters spreading security oriented messages (that are mostly discussed at the Course), screen savers, mouse pads, mugs, stickers, etc. as Security Awareness directors keep finding new and innovative ways of educating staff members. The advantage of this method is that it doesn't push (or, oblige) people in any way, like attending a meeting, listening to lectures, etc.; it is very personalized, user friendly and highly effective due to the fact that it comes very close to their every day life and working procedures within the company (posters, mouse pads, etc.).

Informal discussions are another highly beneficial way to educate and measure the skills of staff where people ask questions, answered by a representative from the ISO; the atmosphere is usually much more informal and calm. This is a highly recommended way of communicating with employees, as it initiates a two-way conversation whereby many points can be covered.

As in many other aspects, you need to find the right balance between the formal and informal ways, as both of these methods have their various advantages and disadvantages. By closely monitoring reactions from staff to the meetings and lectures conducted, you will be able to significantly revise and continuously improve the quality of your Security Awareness Program. Always provide staff with an always-evolving way of education, thus keeping them interested, eager to know and learn, and reducing the chance of boredom, while attending any of the Program's events.

4.10 security threats management

Once you have defined the best way for education, have your plan and strategy ready, measured the computer skills level of your staff, you should start by discussing each of the elements pointed out in the security policy, in detail.

The main purpose of this section is to explore each of these elements in detail and to discuss various threats, providing you with ready made "Best Practices" on various topics along the way. You are encouraged to include parts of this section in your own Security Awareness Course(s), thus providing your staff with a better understanding of the issues covered below.

4.10.1 physical/desktop threats explained

The threats that will be discussed in this section concern the way you use your workstation, access restricted zones in the company, and the way you handle sensitive information. I will cover all the possible threats, discuss their importance in detail, and provide you various effective ways to manage them.

4.10.2 system access

Staff need to be fully aware of their responsibility to keep their User ID and password as secret as possible, and it's all because this is the first line of defense within any system: the identification of the user. Explain to the user that it is completely forbidden to share his/her ID and password with ANYONE, by ANYONE you mean, anyone ranging from the representatives of the Information Security Office (ISO), to their family members. No matter how stupid this might sound to some, they must not do it; even if their manager asks them for their password, they must reject the request. This way, NO ONE can force them into revealing their ID and password, under any circumstances. I know of cases where managers have tried to force (or even, trick) their staff into giving out their passwords for some reason or other in order to evaluate their level of security awareness; to see if they comply with what is stated in the Security Policy, i.e. not to share their ID and password with ANYONE. It is always useful to provide personnel with such "live" examples of how their awareness might, and is being evaluated.

Staff are required not to write any accounting data or ID/password information on loose papers, or sticky (post it) notes, or leave sensitive information on white boards (for example, after a meeting, white boards, and/or flip charts should be cleared off) as this could result in a potential break-in, due to the improper handling of sensitive data. No matter how safe staff might think their password is, they should not be allowed to store them on any of these bits of paper; they must do their best and memorize it instead. Another common mistake that must not be overlooked is the horrifying fact that most of the users tend to hide these notes under the keyboard, or on some "secret" place, as they call it, around their desk; another activity that should be completely forbidden due to obvious reasons. Someone could easily find the "secret" hiding place and get acquainted with vital accounting data.

You must also educate your staff in the way that strong passwords are created. The (secure) ways accounting data must be handled are outlined in the "Password Best Practices" document, which briefly summarizes these two aspects. I have included a sample "Password Creation Best Practices", and a sample "Password Maintenance Best Practices" section below, which will give a overview of what must be taken into account while writing such documents.

4.10.3 password creation best practices

- Passwords must be made up of a mixture of lower-case (small) letters, upper case (capital) letters, numbers, and at least one special character, such as (!@#$%^&*()_+|);
- The minimum length of the password must be at least 8 characters;
- Do not use the same password on several computers and/or services as once revealed, it would compromise the security within all the others in one go.

Good Examples

- Ona327(sA
- @865Dapzl
- 93Sow#-aq

All of these are examples of good passwords, because they fully comply with Password Creation Best Practices; thus containing a mixture of small letters, capital letters, as well as numbers and special characters.

Bad Examples

- aaa123bbb
- abcdefg
- 76543210

The first is a terrible one, and any properly configured cracking program will retrieve it in a matter of minutes, and let's not even mention the second and the third one. The user with the last password (76543210) obviously thought it would be an easy to remember password, as well being a secure one, as it is a long(ish) one; but what the user does not know or realise is the fact that most cracking programs will find it in a matter of seconds (as the password follows a specific numerical pattern). It might be a good idea to incorporate a little demonstration in your Awareness Course at some point providing your staff with the unique opportunity to see how a (password) cracking software operates.

Strong Passwords Creation Tips

- Use the first letters of a quote, song, etc., for example "Something takes a part of me..." would be 'Stpm'
- join two words, include a number, as well as a special character, for example 'run4life#';
- a nice strategy when memorizing passwords might be the following:

  Let's assume your password is Naige453$lZ; first, pronounce it several times in your mind, then ask yourself what your password is, answering this question in the following way: "My password is a mixture of the name Naigel (a foreign friend of mine), several numbers and a dollar sign; my password starts and ends with capital letters, before the last letter of the name (L) there is a dollar sign ($), and before the dollar sign, there are random numbers.

  This is a very useful and helpful trick for anyone who is trying to memorise or remember their password. By repeating (almost explaining) to yourself what your password is describing it the way I suggested above, I am certain that you will not have any problems remembering sophisticated, yet strong passwords.

4.10.4 password maintenance best practices

The proper maintenance of sensitive data such as the User ID and password are a responsibility of every staff member. This section will briefly cover Password Maintenance Best Practices.

- Do NOT share your User ID(s) and password(s) with ANYONE, neither with an ISO representative, help desk staff, family members, nor with your manager(s). No one can force you into revealing your User ID(s) and password(s) under any circumstances, remember that. It is your responsibility to keep the data as secret as possible;
- Do NOT store your User ID(s) and password(s) on any loose bits of paper, sticky (post-it) notes, white boards, flip charts, etc.;
- Do NOT hide your User ID(s) and password(s) under the keyboard, or at any other would be "secret" hiding place. Do your best and memorise it;
- Do change your password(s) following the stated password renewal period in the security policy;
- Before entering your User ID and password, make sure no one is watching you, to avoid the so-called "shoulder surfing" technique.

- Before using your User ID and password on a third-party computer, make sure it is well protected, and free of trojans and key loggers.

4.10.5 virus protection

Based on published papers, expert's predictions, as well as drawing upon personal experiences, I can easily state that viruses will continue to be a very serious threat to critical business data, and will continue to evolve, becoming more sophisticated, dangerous and devastating.

When you start explaining what a virus is, limit it to the facts, for example like how destructive it is, what damage it can cause, the possible financial losses related to a virus outbreak, etc. Don't bother staff with the specific technical information such as ways viruses function, how they hide, and many other topics that will not be of interest to them. Instead, provide those who are most interested, with some external (internet) links to the subject.

Consider explaining what a virus/trojan/worm is, the basic functions of each of these, how to possibly recognize (the operation of) one on your system(s), the potential problems they could cause, and the devastating effects on the whole company. Provide them with live examples, briefly discuss and answer the most simple and frequent questions that come up such as "Can the data corrupted from viruses be recovered", or "What to do once infected with a virus". However, you need to clearly explain that the idea of the presentation is to prevent an infection in the first place, as once infected with a destructive virus, there is not so much you can do, especially if there are no backups of the data.

On the other hand you need to precisely explain what the personal damages would be after a virus infection; damage and/or potential loss of critical business data, documents, projects, business plans, presentations they have been working on, along with any other personal data stored on the computer will be damaged, or, more than likely, be destroyed. By getting to know the devastating effects that viruses may have, staff will be much more aware on the subject, and will more than likely understand the importance of the topic, and the risks for both their company and their home PC's.

Go through the many scenarios of how viruses can get into the company networks, how staff could be tricked into running a virus, the dangers posed by Internet downloads, problems with outdated virus signatures, etc. Also explain the fact that Anti-Virus (AV) scanners are not the best, "fool-proof" solution, and the way they rely on signatures (pattern files). Discuss how useful virus scanners are, and how the effectiveness of preventive measures that are in place depend for a large part on the awareness and vigilance of the users themselves.

Staff need to understand that our main aim is to try and prevent, not to act after we are infected; although there will definitely be infections, we can significantly reduce the risk of infection and limit potential damage by educating staff and making them aware of the dangers posed by malicious code and software (virus/trojan/worm).

The advantages of regular system scanning, as well as the potential problems of not scanning your systems need to highlighted as well; although they know that AV scanners will not detect new viruses, they will at least know that they can reduce the risk, and properly manage the danger.

Scanning the systems using outdated signature files is another common problem that need to be touched upon. Staff should update their Anti-Virus/Anti-Trojan software at least once a week, and if the software allows centralized automatic updates (most do), updates must be scheduled on a regular basis to ensure the software detects the latest viruses/trojans/worms (known to your vendor's lab).

The various ways of getting infected with malicious code should be highlighted as well; let the employees openly ask you questions: see how they react to questions like "How am I getting infected", and then provide them with a better or more complete explanation about the most common as well as specific ways of infections. Below, I have included a sample "Malicious Code Best Practices" section for your convenience ; by no means an exhaustive list, but at least you will be able get an idea of what is considered as dangerous activity.

4.10.6 malicious code best practices

- Do NOT run any files without first scanning them, no matter what the file extension is, i.e. (.exe, .bat, .com, .doc, etc.);
- Do NOT download any files and/or programs from unknown sources; if in doubt, contact the ISO office as soon as possible;
- Do NOT open attachments, even if they were sent by a friend or family member; verify first that indeed, he/she has sent you the file, but nevertheless scan before you open/run anything;
- Do NOT run any programs you have found on diskettes/CD's around your desk if you are not completely sure that they are yours; someone might have placed it there specially for you to "find it and check it out";
- If downloading is allowed, limit it to the minimum; if you need a specific application or something else, always contact the IT department or the ISO office for further information BEFORE you download and installing something;
- Scan (full system scan) your system at least once per week with your default AV scanner software. Be sure to update the virus signatures before you do so, and also consider automating the process by scheduling a Full System Scan for convenient regular scanning in the future;
- Update the signature files as often as possible, so to ensure that the latest malicious software patterns are detected;
- The IT department or the Information Security Office will usually NEVER mail you the latest updates of any software (unless this is preceded by a much publicised, well-advertised, company wide campaign). If you detect suspicious activity, do not delete the e-mail received and contact the Incident Handling or Help Desk team as soon as possible;
- if you have any doubts regarding malicious software (viruses/trojans/worms), contact the ISO, The Help Desk or the IT department immediately. This way you will prevent any potential devastating mishaps, due to inappropriate and erroneous handling of dangerous and harmful incidents.

Everything that is defined as forbidden must be discussed and explained; why it is forbidden or restricted, how it could harm the company or the business, etc. Play out several potential scenarios, thus helping the users grasp the topic in an easy to understand way while trying to touch base on the consequences of all of these dangerous activities.

4.10.7 software installation

Freeware, or any other type of software, obtained or downloaded from unknown or untrustworthy sources could easily affect company security, exposing critical business data and/or corrupting sensitive ones. A lot of users tend to install such programs (from screen savers to games and funny cartoons in Flash) as they put it, for various personal needs and activities; to entertain, have something nice to look at or relax themselves. At the same time, they do not realise the potential threats they are exposing the company systems and networks to, from malicious software (viruses/trojans/worms) to legal actions against the company for installing (possibly) pirated software on the company workstation(s). Thus, you need to familiarise users with the potential problems attached to each of these issues, and also explain the company policy towards installation of any (unauthorised) software on any of the company workstation(s). Files downloaded from the Internet, copied from a CD or a floppy coming from an unknown source, or anything else that has not been reviewed by the Information Security Office or not been scanned for potential malicious code (by the corporate AV systems) could actually be classified as untrustworthy, unknown and dangerous. Freeware applications, due to their nature of origin, are a significant source of threat and should be approached with caution.

Staff members need to be aware of the risks involved, and learn to think twice before they act on issues. This can be stimulated in many ways; by playing out various scenarios on how software downloaded from either the Internet or copied from any removal media could endanger the company, its business, one's privacy, or the use of company bandwidth to commit illegal actions.

It is entirely up to you to decide whether users should be allowed to download and/or install third party programs on their workstations; and implement the appropriate (security) policies and procedures that go with this decision. You will not only need to clearly state the consequences for those who violate any restrictions, but also provide the procedures for obtaining and installing new software.

It is highly recommended that users do not have the ability to install any new programs that might either expose sensitive company information, waste valuable bandwidth, or corrupt critical data. If users need new software installed for business use, they should contact their manager, the IT/IS department, the Help Desk or the ISO (depending on the procedures set out in the policy) instead of undertaking such action themselves.

4.10.8 removable media (CD's, floppies, tapes, etc.)

Removable media such as CD's (Compact Disks), floppies (Floppy Disks) and even tapes (backup/ADR/DAT/DLT tapes) can be defined as another possible entry point for dangerous and malicious files entering the company network or endangering the security of a single workstation. On the other hand, these can also be used to illegally copy sensitive data on, after which it would be easy to walk out of the premises with the stolen information.

Malicious software (viruses/trojans/worms) also use removable media to spread; some take advantage of the auto-run feature of the CD (automatically executing the auto-start file on the CD, which could be a destructive one), others still use "classic" methods like diskettes to get the workstation infected with a malicious program. For best results, removable media devices should be banned entirely (utilising floppy drive-locks, or 'CD-less' workstations; CD's can still be used via CD-Towers, for example). If you need to use removable media in your organisation, then best practices must be established, possible risks and danger scenarios discussed so to reduce malicious programs entering your networks at all points, thus protecting your company from a major disaster.

4.10.9 encryption

Encryption can be defined as another "must implement" measure that will not only keep your sensitive and critical information secured against a potential attacker, but also protect you from a lot of trouble if eventually a security breach does occur. In your security policy and procedures you must clearly define the systems, files and documents that should be encrypted, by whom, and most importantly, using which algorithms. It is strongly recommended to use proven, industry standard algorithms, such as DES, IDEA, Blowfish, or RC5.

4.10.10 system backups

Disaster recovery (DR) plans are essential for the continuity of the business as well as the proper functionality of the current processes. Sooner or later you will inevitably face the problem where a system crashes, no matter of the OS used, but this can be dealt with promptly, if proper backup procedures and disaster recovery plans are in place.

You will have to define the assets that must be backed up on a regular basis, the responsible individuals, best practices and procedures, as well as where the backups should be stored, i.e. a fireproof safe, vault, off-site, etc.

4.10.11 maintenance

The proper maintenance of the PC/workstation is another vital issue that must not be overlooked during the course of the Security Awareness Program. Users' workstations are a significant source of threat to company security, often targeted by the so-called "insiders" snooping around, looking for unprotected workstations. Therefore, you need to educate staff on the aspect of Physical Security as well; again, this can be achieved by running through the possible scenarios, while providing tips for better overall protection.

4.10.12 incident handling

By now your staff should be able to define a potential security problem, while you should be establishing the rules for the course of action to take in case of an incident. In your policy you must clearly state what must be done in various situations; the main idea here should be to minimise and limit damage. Staff should be made aware who is responsible for handling problems, and whom they should contact as soon as they suspect a potential security problem.

4.10.13 internet threats explained

One of the greatest security risks in the company is the Internet connectivity, and its misuse through (uneducated) employees. It is a fact that most employees will surf to sites that are strictly prohibited, and most probably will end up downloading malicious files and/or hostile code from hacker sites somehow. Any of these activities could impact the productivity of your company, especially if you think about the recovery process trying to rectify the mistakes made by staff.

Therefore, it is always a good idea to explain in detail the possible dangers of surfing the Internet; that you don't need to download anything at all to get the computer infected with a virus, trojan or even a worm but just visiting the site is enough to cause a problem. Define what constitutes a "prohibited site", and explain why it is prohibited, including the problems that could occur just by visiting it.

4.10.14 web browsing

Web browsing represents a threat to the security of the workstation, as well as to the whole organisation. Being exposed to the dangers of web browsing is very easy as hostile scripts could be downloaded, and executed automatically; all it takes for example is an outdated version of the web browser.

Staff should be able to make a distinction between sites that are classified as allowed, prohibited or potentially dangerous, and try avoid visiting prohibited ones. Java and ActiveX should be disabled by default (it will not give problems accessing pages), care must be taken with Flash movies, etc. and if ever a hint of a problem occurs, the ISO office must be contacted immediately.

There are web sites in the wild, that could attempt to scan/flood your network, just by visiting them; another variant to this (theoretical, but very possible) scenario is one of your employees using some kind of scanning service to check the security of his/her workstation, thus wasting valuable bandwidth. Something like this will invariably produce more work for the ISO office as well as their systems probably will register the usage of this service as a possible break-in attempt. Online gambling and pornographic web sites should be fully prohibited, and the web usage of staff monitored to ensure they are following the rules and regulations set forth by the Security Awareness Program.

4.10.15 e-mail use

Generally the company E-mail systems are a high risk area due to their constant availability to the outside world, and the risk is often two-fold. The use of e-mail to conduct business, contact clients, and its integration in many other business related processes exposes company mail addresses and (mail) systems to potential attackers. On the other hand, this is also the number one entry point from which most of the malicious programs are entering the company. Therefore, a well-known and proven malicious code protection program is a must have on all the mail gateways, as it will detect, block and/or filter out most of the known dangerous files and hostile scripts trying to enter the company networks.

As with all aspects of IT security, company-wide security can only be improved through the proper education of staff. It is therefore highly recommended that you establish Best Practices for E-mail use, concentrating on the points below.

4.10.16 e-mail use best practices

- If (E-mail) attachments are allowed, the attachment(s) must be scanned before opening as well as confirming with the sender (i.e. via phone) that indeed an attachment has been sent. This will also reduce the risk of running a program that has been e-mailed out automatically (unknown to the originator) via some kind of malicious application that has made use of the mail account(s) and/or mailing system of the sender. If attachments are forbidden, follow the policy and do not download/run any file(s) received as attachments;

- Java and ActiveX must be disabled while reading e-mail in order to manage the risk of auto-executing malicious programs. Just like in the internet browser, certain options of the program can usually be set and locked by way of system policies that automatically set these conditions for all users at logon;
- Do not use the company e-mail accounts for registration purposes of any kind, and do not use it while posting messages in web forums or newsgroups. You may want to create one, special (possibly aliased) account for this purpose only;
- Do not use the company e-mail system for running your own business, excessive personal mailing, sending large attachments, thus wasting valuable bandwidth;
- Do not respond to chain letters, or any other sort of spam using the company e-mail systems; if in doubt, contact the ISO office;
- Never forward any company data to external e-mail accounts (i.e. send a work document to your home email account, so to work on it further from home that evening), without first checking with your manager and/or contacting the ISO office;
- The proper use of the E-mail system should continuously be monitored and the users should be aware that they could be held liable for illegal activities, such as spamming, sending and receiving illegal content, etc.

4.10.17 instant messaging (IM) applications (ICQ, AOL, MSN, etc.)

A lot of users tend to use these programs in order to communicate with friends, send and receive attachments, messages, etc as these applications often try to trick the content blocking gateway at the server level to letting content pass through. However, they do not fully realise the dangers of these programs, and the potential damages that they could cause.

A snapshot from our previous publication 'The Complete Windows Trojans Paper', available from our web site at http://www.frame4.com/publications/index.php, reviews various scenarios of getting infected with a malicious program via ICQ:

- You can never be 100 percent sure who is on the other side of the computer at that particular moment. It could be someone that hacked your friend’s ICQ UIN (Unique Identification Number) and wants to spread some trojans;
- Old versions of ICQ had bugs in the WebServer feature that creates a web site on your computer with your info from the ICQ database. The bug meant that the attacker could have access to EVERY file on your machine ... and you probably realise what could happen if someone has access to your win.ini or some other system file: a trojan installed on your computer in a few minutes;
- Trojan.exe is renamed to Trojan....(150 spaces).txt.exe, the icon changed to a real .txt file; this will definitely get you infected. This bug will most probably be fixed in the newer versions.

No matter the Instant Messaging application you are using, you could always get infected or exploited; through a specific application bug you never heard about or a buggy version you never bothered updating.

When it comes to exchanging information and files no matter where, from whom or how, please be aware that there are certain dangers attached to it; realise the possible dangers of your actions and your naivety, and act accordingly.

4.10.18 downloading

Downloading any data from unknown and untrustworthy sources while using company systems and networks could have a devastating effect on the business processes; you could face a situation of having your data lost, corrupted, or, in certain cases, modified. You should therefore aim to educate staff on the procedures of downloading information in a safe manner; this consists of ensuring downloadingf files only when it is absolutely necessary, scanning of the downloaded files with the corporate Anti-Virus/Anti-Trojan solution before opening it, etc.

For your convenience we have created a summarised "Internet Use Best Practices" section below; again, far from being an exhaustive list, it is aimed at giving you some basic pointers on safe Internet use.

4.10.19 internet use best practices

- Java and ActiveX are blocked by default. Scripts containing Java and ActiveX pose a great danger due to their insecure nature, and the resulting problems could have devastating effects on your computer, not to mention the company. Please do not block, stop or tamper with any measure (i.e. group policy) that is in place to filter out these and if you are having problems purchasing an item or visiting a trusted web site, contact the IT department, Help Desk or the ISO office for assistance;
- Do not visit inappropriate web sites with objectionable content; pornography, gambling, warez (pirated software), hacker/hacking sites, as well as those generally considered as prohibited by your security policy;
- If the use of Instant Messaging (IM) applications is allowed, do not accept any attachments no matter of the file type, extension, or originator;
- Downloading software, files or anything else is prohibited. If you need any applications for your dayto- day business, contact either the IT department, the Help Desk or the ISO office. You will more than likely need to hand in a (software) request form signed by your manager to complete the process. If you do get clearance to download a piece of software, remember to never execute it before scanning them with the corporate Anti-Virus/Anti-Trojan software;
- All internet activity should continuously be monitored and the users should be aware that they could be held liable for visiting prohibited web sites, downloading illegal files and content, as well as face a penalty of having their access to the Internet limited (until they can prove that they are fully aware of the risks created by their actions).

chapter 5 vulnerabilities

The **S**.

5.1 paragraph 1

One

5.2 par 2

One iteration

chapter 6 threats

6.1 overview

A computer-based system has three separate but valuable components: hardware, software, and data.

Each of these assets offers value to different members of the community affected by the system.

A threat to a computing system is a set of circumstances that has the potential to cause loss or harm.

There are two types of network threats:

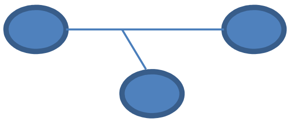
- **Logic attacks** – are known to exploit existing software bugs and vulnerabilities with the intent of crashing a system
- **Resource attacks** – are intended to overwhelm critical system resources such as CPU and RAM

There are many threats to a computer system, including human-initiated and computer-initiated ones. We have all experienced the results of inadvertent human errors, hardware design flaws, and software failures. But natural disasters are threats, too; they can bring a system down when the computer room is flooded or the data center collapses from an earthquake, for example.

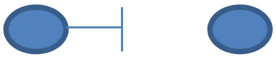
A human who exploits vulnerability perpetrates an attack on the system. An attack can also be launched by another system, as when one system sends an overwhelming set of messages to another, virtually shutting down the second system's ability to function. We can say that a threat is blocked by control of vulnerability.

There are four types of security threats:

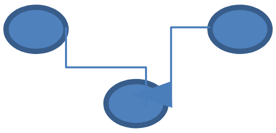
- An **interception** means that some unauthorized party has gained access to an asset. The outside party can be a person, a program, or a computing system. Examples of this type of failure are illicit copying of program or data files, or wiretapping to obtain data in a network. Although a loss may be discovered fairly quickly, a silent interceptor may leave no traces by which the interception can be readily detected.



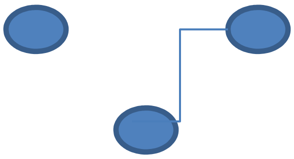
- An **interruption** means that an asset of the system becomes lost, unavailable, or unusable. An example is malicious destruction of a hardware device, erasure of a program or data file, or malfunction of an operating system file manager so that it cannot find a particular disk file.



- If an unauthorized party not only accesses but tampers with an asset, the threat is a **modification**. For example, someone might change the values in a database, alter a program so that it performs an additional computation, or modify data being transmitted electronically. It is even possible to modify hardware. Some cases of modification can be detected with simple measures, but other, more subtle, changes may be almost impossible to detect.



- Finally, an unauthorized party might create a **fabrication** of counterfeit objects on a computing system. The intruder may insert spurious transactions to a network communication system or add records to an existing database. Sometimes these additions can be detected as forgeries, but if skillfully done, they are virtually indistinguishable from the real thing.



A malicious attacker must have three things (MOM acronym):

- **method**: the skills, knowledge, tools, and other things with which to be able to pull off the attack
- **opportunity**: the time and access to accomplish the attack
- **motive**: a reason to want to perform this attack against this system

With an increasing amount of people getting connected to network, the security threats that cause harm are increasing also. Network security is a major part of a network that needs to be maintained because information is being passed between computers and is vulnerable to attack.

The biggest network threats are the following:

- viruses and worms
- trojan horses
- spam
- phishing
- packet sniffers
- spyware
- rootkits
- backdoors
- password attacks
- zombie computers and botnets

6.2 viruses and worms

A **computer virus** is a malware program that, when executed,replicates by inserting copies of itself (possibly modified) into other computer programs, data files, or the boot sectors of the hard drive; when this replication succeeds, the affected areas are then said to be "infected". Viruses often perform some type of harmful activity on infected hosts, such as stealing hard disk space or CPU time, accessing private information, corrupting data, displaying political or humorous messages on the user's screen, spamming their contacts, or logging their keystrokes.

A **computer worm** is a standalone malware computer program that replicates itself in order to spread to other computers. Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it. Unlike a computer virus, it does not need to attach itself to an existing program.Worms almost always cause at least some harm to the network, even if only by consuming bandwidth, whereas viruses almost always corrupt or modify files on a targeted computer.

6.3 trojans

Similar to the mythical wooden horse used by the Greeks to invade Troy, a **Trojan Horse** is “a program in which malicious or harmful code is contained inside apparently harmless programming or data in such a way that it can get control and do its chosen form of damage, such as ruining the file allocation table on your hard disk”. On network, they are even more dangerous. They do not have the ability to self-replicate but to deliver destructive payloads and unload viruses, worms or spyware.

A Trojan often acts as a backdoor, contacting a controller which can then have unauthorized access to the affected computer. While Trojans and backdoors are not easily detectable by themselves, computers may appear to run slower due to heavy processor or network usage. Malicious programs are classified as Trojans if they do not attempt to inject themselves into other files (computer virus) or otherwise propagate themselves (worm).

6.4 spam

SPAM is “flooding the Internet with many copies of the same message, in an attempt to force the message on people who would not otherwise choose to receive it.”

Clicking on links in spam email may send users to phishing web sites or sites that are hosting malware. Spam email may also include malware as scripts or other executable file attachments. Definitions of spam usually include the aspects that email is unsolicited and sent in bulk.

The solution against them is spam filters which come with most of the email clients.

6.5 phishing

Phishing is an email fraud method in which the perpetrator sends out legitimate-looking email in an attempt to gather personal and financial information from recipients. Is one of the worst security threat over a network because a lot of people are vulnerable to giving out information that could cause money theft or identity theft.

Communications purporting to be from popular social web sites, auction sites, banks, online payment processors or IT administrators are commonly used to lure unsuspecting public. Phishing emails may contain links to websites that are infected with malware. Phishing is typically carried out by email spoofing or instant messaging, and it often directs users to enter details at a fake website whose look and feel are almost identical to the legitimate one. Phishing is an example of social engineering techniques used to deceive users, and exploits the poor usability of current web security technologies. Attempts to deal with the growing number of reported phishing incidents include legislation, user training, public awareness, and technical security measures. Many websites have now created secondary tools for applications, like maps for games, but they should be clearly marked as to who wrote them, and users should not use the same passwords anywhere on the internet.

Phishing is a continual threat that keeps growing to this day. The risk grows even larger in social media such as Facebook, Twitter, Myspace etc. Hackers commonly use these sites to attack persons using these media sites in their workplace, homes, or public in order to take personal and security information that can affect the user and the company (if in a workplace environment). Phishing is used to portray trust in the user since the user may not be able to tell that the site being visited or program being used is not real, and when this occurs is when the hacker has the chance to access the personal information such as passwords, usernames, security codes, and credit card numbers among other things.

There are filters designed to prevent this kind of threats, similar to spam filters.

6.6 packet sniffers

A **packet sniffer** is a device or program which allows eavesdropping on traffic traveling between networked computers; it will capture data that is addressed to other machines, saving it later for analysis.

As data streams flow across the network, the sniffer captures each packet and, if needed, decodes the packet's raw data, showing the values of various fields in the packet, and analyzes its content according to the appropriate RFC or other specifications.

So again, personal information is at risk and the solution is to encrypt the data.

6.7 spyware

Spyware is software that aims to gather information about a person or organization without their knowledge and that may send such information to another entity without the consumer's consent, or that asserts control over a computer without the consumer's knowledge.

"Spyware" is mostly classified into four types: system monitors, trojans, adware, and tracking cookies. Spyware is mostly used for the purposes of tracking and storing Internet users' movements on the Web and serving up pop-up ads to Internet users.

Whenever spyware is used for malicious purposes, its presence is typically hidden from the user and can be difficult to detect. Some spyware, such as keyloggers, may be installed by the owner of a shared, corporate, or public computer intentionally in order to monitor users.

It is a ‘sneaky’ program that tracks and reports your computing activity without consent, such as browsing patterns in the more benign case or credit card numbers in more serious ones. It usually comes bundled with free software and automatically installs itself with the program you intended to use.

6.8 rootkits

A **rootkit** is a stealthy type of software, typically malicious, designed to hide the existence of certain processes or programs from normal methods of detection and enable continued privileged access to a computer. The term *rootkit* is a concatenation of "roo" (the traditional name of the privileged account on Unix operating systems) and the word "kit" (which refers to the software components that implement the tool). The term "rootkit" has negative connotations through its association with malware.

Rootkit installation can be automated, or an attacker can install it once they've obtained root or Administrator access. Obtaining this access is a result of direct attack on a system (i.e.), exploiting a known vulnerability (such as privilege escalation) or a password (obtained by cracking or social engineering). Once installed, it becomes possible to hide the intrusion as well as to maintain privileged access. The key is the root or Administrator access. Full control over a system means that existing software can be modified, including software that might otherwise be used to detect or circumvent it.

6.9 backdoors

A **backdoor** in a computer system is a method of bypassing normal authentication, securing unauthorized remote access to a computer, obtaining access to plaintext, and so on, while attempting to remain undetected. The backdoor may take the form of a hidden part of a program; a separate program may subvert the system through a rootkit.

Default passwords can function as backdoors if they are not changed by the user. Some debugging features can also act as backdoors if they are not removed in the release version

6.10 password attacks

Password attacks are attacks by hackers that are able to determine passwords or find passwords to different protected electronic areas. Many systems on a network are password protected and hence it would be easy for a hacker to hack into the systems and steal data. Password cracking is the process of recovering passwords from data that have been stored in or transmitted by a computer system. A common approach (brute-force attack) is to try guesses repeatedly for the password and check them against an available cryptographic hash of the password. There is no solution for to moment to prevent, just to create long and complicated password by using uppercase letters, special characters and numbers.

6.11 zombie computers and botnets

A z**ombie computer** is a computer that has been secretly compromised by hacking tools which allows a third party application to control the computer and its resources remotely

A **botnet** is a number of Internet computers that, although their owners are unaware of it, have been set up to forward transmissions to other computers on the internet. This is a major security threat, because the network can act as a hub that forwards malicious files let's say to other computers.

6.12 threats protection

We seek to protect hardware, software, and data; to make it particularly hard for an intruder to find data useful we scramble the data so that interpretation is meaningless without the intruder knows how the scrambling was done.

Encryption is the formal name for the scrambling process. We take data in their normal, unscrambled state, called cleartext, and transform them so that they are unintelligible to the outside observer; the transformed data are called enciphered text or ciphertext. Using encryption, security professionals can virtually nullify the value of an interception and the possibility of effective modification or fabrication.. Encryption clearly addresses the need for confidentiality of data. Additionally, it can be used to ensure integrity; data that cannot be read generally cannot easily be changed in a meaningful manner. Encryption is the basis of protocols that enable us to provide security while accomplishing an important system or network task. A protocol is an agreed-on sequence of actions that leads to a desired result. For example, some operating system protocols ensure availability of resources as different tasks and users request them. Thus, encryption can also be thought of as supporting availability. That is, encryption is at the heart of methods for ensuring all aspects of computer security. Although encryption is an important tool in any computer security tool kit, we should not overrate its importance. Encryption does not solve all computer security problems, and other tools must complement its use. Furthermore, if encryption is not used properly, it may have no effect on security or could even degrade the performance of the entire system. Weak encryption can actually be worse than no encryption at all, because it gives users an unwarranted sense of protection. Therefore, we must understand those situations in which encryption is most useful as well as ways to use it effectively

Another solution is to use a firewall; a **firewall** is a network security system that controls the incoming and outgoing network traffic based on an applied rule set. A firewall establishes a barrier between a trusted, secure internal network and another network (e.g., the Internet) that is assumed not to be secure and trusted. Firewalls exist both as software to run on general purpose hardware and as a hardware appliance. Many hardware-based firewalls also offer other functionality to the internal network they protect, such as acting as a DHCP server for that network.

chapter 7 passive attacks

There are two main types of passive attack:

- traffic analysis
- Non-evasive eavesdropping and monitoring of transmissions

7.1 traffic analysis

7.1.1 Military roots

Traffic analysis is a key part of signal intelligence and electronic warfare. Michael Hermann, who has served as chair of the UK Joint Intelligence Committee, in his book 'Intelligence Power in Peace and War' describes the value of extracting data from non-textual (to be understood as 'not content') sources: These non-textual techniques can establish targets' locations, order of-battle and movement. Even when messages are not being deciphered, traffic analysis of the target's C3I system and its patterns of behavior provides indications of his intentions and states of mind, in rather the same way as a neurologist develops insights about a silent patient by studying EEG traces from the brain. Traffic analysis was used in military circles even before the invention of wireless communications. Anderson in his book [3] mentions that in the trench warfare of World War I, the earth returns of the telegraph communication of the enemy was used to extract information up to a few hundred yards away from the transmitting station. Traffic analysis though became an extremely potent source of intelligence when wireless communication became popular, particularly in naval and air operations. Ships at see had to balance the value of communicating against the threat of being detected via direction finding if they transmit. When transmitting strict standards, governing call-signs and communication, had to be adhered too in order to minimize the information that traffic analysis could provide. Another example of traffic analysis providing valuable intelligence (by Herman) is the reconstruction of the structure of the network structure of the German Air Force radio in 1941 by the British, confirming that a unit was composed of nine and not twelve planes. This allowed a more accurate estimate of the total strength of their opponent. Identification of radio equipment can also be used to detect accurate movements of units: each transmitter has characteristics such as the unintentional frequency modulations, the shape of the transmitter turn-on signal transient, the precise center of frequency modulation, etc.

7.2 eavesdropping and transmission monitoring

One iteration

chapter 8 active attacks

The **S**.

8.1 paragraph 1

One

8.2 par 2

One iteration

chapter 9 hash functions

9.1 overview

Hash functions take a message of arbitrary length as input and generate a fixed length digest (checksum). The length of the digest depends on the function used, but in general is between 128 and 512 bits.

The hash functions are used in 3 main areas:

- assure the integrity of a message (or of a downloaded file) by attaching the generated digest to the message itself. The receiver recomputes the digest using the received message and compares it against the digest generated by the sender.
- are part of the creation of the digital signature
- password storage – password are (almost) never stored in their original form. What is stored, in general, is a hash of the password. When a user introduces a password, its hash is computed and is compared with the stored hash.

The most used hash functions are those in the MD and the SHA families – namely MD5 and SHA-1 and the newest ones SHA-2 and SHA-3. Another hash function of interest is RipeMD-160. The MD functions generate a 128 bit digest and were designed by the company RSA Security. While MD5 is still widespread, MD4 has been broken and is deemed insecure. SHA-1 and RipeMD-160 are considered safe for now. While SHA-2 is an extension of SHA-1, SHA_3 features a brand new algorithm for computing the hash.

Starting with the newest function, here is a list of hash functions of practical interest.

- SHA-3 uses the Keccak algorithm, a sponge construction in which message blocks are XORed into a subset of the state, which is then transformed as a whole. In the version used in SHA-3, the state consists of a 5×5 array of 64-bit words, 1600 bits total. The standardization process is not finished yet as of April 2015.
- SHA-2 includes significant changes from its predecessor, SHA-1. The SHA-2 family consists of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits: **SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256**.
- SHA-1 – Secure Hash Algorithm. Published by the US Government. Its specification is the object of FIPS 180-1 (April 1995). FIPS stands for Federal Information Processing Standards. Produces a 160 bit digest (5 32-bit words).
- RipeMD-160 – designed as a replacement for the MD series. It produces a digest of 160 bits (or 20 bytes, if you want).
- MD5 – Message Digest Algorithm 5. Developed by RSA Labs. Produces a 128 bit digest. Still in use, especially for message (download) integrity check.
- MD2, MD4 – Older hash algorithms from RSA Data Security. Since they have known flaws, they are only of historic interest.

9.2 MD5

9.2.1 General description

MD5 is a **block** hash function (the block size is **512 bits**) which has been developed by Rivest in 1991. The input for MD5 is an arbitrary length message or file, while the output is is a fixed length digest. The length of this digest is **128 bits** or 4 words. The formal specification of this hash algorithm is specified in RFC 1321.

9.2.2 algorithm description

OWe begin by supposing that we have a **b-bit message** as input, and that we wish to find its message **digest**. Here b is an arbitrary nonnegative integer; b may be zero, it need not be a multiple of eight, and it may be arbitrarily large. We imagine the bits of the message written down as follows:

m_0 m_1 ... m_{b-1}

The following five steps are performed to compute the message digest of the message.

9.2.3 step 1 - append padding bits

The message is "**padded**" (extended) so that its length (in bits) is congruent to 448, modulo 512. That is, the message is extended so that it is just 64 bits shy of being a multiple of 512 bits long. Padding is always performed, even if the length of the message is already congruent to 448, modulo 512.

Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended.

9.2.4 step 2 - append length

A 64-bit representation of b (the length of the message before the padding bits were added) is appended to the result of the previous step. In the unlikely event that b is greater than 2^64, then only the low-order 64 bits of b are used. (These bits are appended as two 32-bit words and appended low-order word first in accordance with the previous conventions.)

At this point the resulting message (after padding with bits and with the length) has a length that is an exact multiple of 512 bits. Equivalently, this message has a length that is an exact multiple of 16 (32-bit) words. Let M[0 ... N-1] denote the words of the resulting message, where N is a multiple of 16.

9.2.5 step 3 - initialize the MD buffer

A four-word buffer (A,B,C,D) is used to compute the message digest. Here each of A, B, C, D is a 32-bit register. These registers are **initialized** to the following values in hexadecimal, low-order bytes first):

```
word A: 01 23 45 67
word B: 89 ab cd ef
word C: fe dc ba 98
word D: 76 54 32 10
```

9.2.6 step 4 - process message in 16-word blocks

We first define four auxiliary functions that each take as input three 32-bit words and produce as output one 32-bit word.

```
F(X,Y,Z) = (X&Y)|(~X&Z)
G(X,Y,Z) = (X&Z)|(Y&~Z)
H(X,Y,Z) = X^Y^Z
I(X,Y,Z) = Y^(X|~Z)
```

In each bit position F acts as a conditional: if X then Y else Z. The function F could have been defined using + instead of or since XY and not(X)Z will never have 1's in the same bit position.) It is interesting to note that if the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z) will be independent and unbiased.

The functions G, H, and I are similar to the function F, in that they act in "bitwise parallel" to produce their output from the bits of X, Y, and Z, in such a manner that if the corresponding bits of X, Y, and Z are independent and unbiased, then each bit of G(X,Y,Z), H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased. Note that the function H is the bit-wise "xor" or "parity" function of its inputs.

This step uses a 64-element table T[1 ... 64] constructed from the sinus function. Let T[i] denote the i-th element of the table, which is equal to the integer part of 4294967296 times abs(sin(i)), where i is in radians. The elements of the table are given in the appendix.

Below, N is the number of words. Because the last block of 512 bits (16 words) has been padded, N is a multiple of 16 while N/16 is the number of blocks in the padded message.

Do the following:

```
/* Process each 16-word block. */
For i = 0 to N/16 - 1 do

  /* Copy block i into X. */
  For j = 0 to 15 do
    Set X[j] to M[i*16+j].
  end /* of loop on j */

  /* Save A as AA, B as BB, C as CC, and D as DD. */
  AA = A
  BB = B
  CC = C
  DD = D

  /* Round 1. */
  /* Let [abcd k s i] denote the operation
       a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */
  /* Do the following 16 operations. */
  [ABCD  0  7  1]  [DABC  1 12  2]  [CDAB  2 17  3]  [BCDA  3 22  4]
  [ABCD  4  7  5]  [DABC  5 12  6]  [CDAB  6 17  7]  [BCDA  7 22  8]
  [ABCD  8  7  9]  [DABC  9 12 10]  [CDAB 10 17 11]  [BCDA 11 22 12]
  [ABCD 12  7 13]  [DABC 13 12 14]  [CDAB 14 17 15]  [BCDA 15 22 16]

  /* Round 2. */
  /* Let [abcd k s i] denote the operation
       a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */
  /* Do the following 16 operations. */
  [ABCD  1  5 17]  [DABC  6  9 18]  [CDAB 11 14 19]  [BCDA  0 20 20]
  [ABCD  5  5 21]  [DABC 10  9 22]  [CDAB 15 14 23]  [BCDA  4 20 24]
  [ABCD  9  5 25]  [DABC 14  9 26]  [CDAB  3 14 27]  [BCDA  8 20 28]
  [ABCD 13  5 29]  [DABC  2  9 30]  [CDAB  7 14 31]  [BCDA 12 20 32]

  /* Round 3. */
  /* Let [abcd k s t] denote the operation
       a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */
  /* Do the following 16 operations. */
  [ABCD  5  4 33]  [DABC  8 11 34]  [CDAB 11 16 35]  [BCDA 14 23 36]
  [ABCD  1  4 37]  [DABC  4 11 38]  [CDAB  7 16 39]  [BCDA 10 23 40]
  [ABCD 13  4 41]  [DABC  0 11 42]  [CDAB  3 16 43]  [BCDA  6 23 44]
  [ABCD  9  4 45]  [DABC 12 11 46]  [CDAB 15 16 47]  [BCDA  2 23 48]

  /* Round 4. */
  /* Let [abcd k s t] denote the operation
       a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */
  /* Do the following 16 operations. */
```

```
    [ABCD  0  6 49]  [DABC  7 10 50]  [CDAB 14 15 51]  [BCDA  5 21 52]
    [ABCD 12  6 53]  [DABC  3 10 54]  [CDAB 10 15 55]  [BCDA  1 21 56]
    [ABCD  8  6 57]  [DABC 15 10 58]  [CDAB  6 15 59]  [BCDA 13 21 60]
    [ABCD  4  6 61]  [DABC 11 10 62]  [CDAB  2 15 63]  [BCDA  9 21 64]

    /* Then perform the following additions. (That is increment each
       of the four registers by the value it had before this block
       was started.) */
    A = A + AA
    B = B + BB
    C = C + CC
    D = D + DD

  end /* of loop on i */
```

9.2.7 step 5 - output

The message digest produced as output is A, B, C, D. That is, we begin with the low-order byte of A, and end with the high-order byte of D.

This completes the description of MD5.

9.2.8 some examples

The examples below are part of the test suite for MD5 and are specified in RFC 1321.

```
MD5("") = d41d8cd9 8f00b204 e9800998 ecf8427e

MD5("a") = 0cc175b9 c0f1b6a8 31c399e2 69772661

MD5("abc") = 90015098 3cd24fb0 d6963f7d 28e17f72

MD5("message digest") = f96b697d 7cb7938d 525a2f31 aaf161d0

MD5("abcdefghijklmnopqrstuvwxyz") = c3fcd3d7 6192e400 7dfb496c ca67e13b

MD5("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789") =

   d174ab98 d277d9f5 a5611c2c 9f419d9f

MD5 ("123456789012345678901234567890123456789012345678901234567890123456

78901234567890") = 57edf4a2 2be3c955 ac49da2e 2107b67a
```

9.2.9 cryptanalysis

In 1993, Den Boer and Bosselaers gave an early, although limited, result of finding a "pseudo-collision" of the MD5 compression function; that is, two different initialization vectors which produce an identical digest.

In 1996, Dobbertin announced a collision of the compression function of MD5 (Dobbertin, 1996). While this was not an attack on the full MD5 hash function, it was close enough for cryptographers to recommend switching to a replacement, such as WHIRLPOOL, SHA-1 or RIPEMD-160.

The size of the hash—128 bits—is small enough to contemplate a birthday attack. MD5CRK was a distributed project started in March 2004 with the aim of demonstrating that MD5 is practically insecure by finding a collision using a birthday attack.

MD5CRK ended shortly after 17 August 2004, when collisions for the full MD5 were announced by Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu. Their analytical attack was reported to take only one hour on an IBM p690 cluster.

On 1 March 2005, Arjen Lenstra, Xiaoyun Wang, and Benne de Weger demonstrated the construction of two X.509 certificates with different public keys and the same MD5 hash, a demonstrably practical collision.

The construction included private keys for both public keys. A few days later, Vlastimil Klima described an improved algorithm, able to construct MD5 collisions in a few hours on a single notebook computer. On 18 March 2006, Klima published an algorithm that can find a collision within one minute on a single notebook computer, using a method he calls tunneling.

An actual collision can be found at http://www.mathstat.dal.ca/~selinger/md5collision/ . We reproduce it here, since the two messages are not that long. Moreover, the two messages are almost identical.

```
d131dd02c5e6eec4693d9a0698aff95c 2fcab58712467eab4004583eb8fb7f89
55ad340609f4b30283e488832571415a 085125e8f7cdc99fd91dbdf280373c5b
d8823e3156348f5bae6dacd436c919c6 dd53e2b487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080a80d1e c69821bcb6a8839396f9652b6ff72a70

d131dd02c5e6eec4693d9a0698aff95c 2fcab50712467eab4004583eb8fb7f89
55ad340609f4b30283e4888325f1415a 085125e8f7cdc99fd91dbd7280373c5b
d8823e3156348f5bae6dacd436c919c6 dd53e23487da03fd02396306d248cda0
e99f33420f577ee8ce54b67080280d1e c69821bcb6a8839396f965ab6ff72a70
```

The common MD5 digest of these two messages is - 79054025-255fb1a2-6e4bc422-aef54eb4

9.3 SHA-1

9.3.1 general description

The original specification of the algorithm was published in 1993 as the *Secure Hash Standard*, FIPS PUB 180, by US government standards agency NIST (National Institute of Standards and Technology). This version is now often referred to as *SHA-0*. It was withdrawn by NSA shortly after publication and was superseded by the revised version, published in 1995 in FIPS PUB 180-1 and commonly referred to as *SHA-1*. SHA-1 differs from SHA-0 only by a single bitwise rotation in the message schedule of its compression function; this was done, according to NSA, to correct a flaw in the original algorithm which reduced its cryptographic security. However, NSA did not provide any further explanation or identify the flaw that was corrected. Weaknesses have subsequently been reported in both SHA-0 and SHA-1. SHA-1 appears to provide greater resistance to attacks, supporting the NSA's assertion that the change increased the security.

SHA-1 (as well as SHA-0) produces a 160-bit digest from a message with a maximum length of (2^{64} − 1) bits. SHA-1 is based on principles similar to those used by Ronald L. Rivest of MIT in the design of the MD4 and MD5 message digest algorithms, but has a more conservative design.

9.3.2 algorithm description

The SHA-1 is used to compute a message digest for a message or data file that is provided as input. The message or data file should be considered to be a bit string. The length of the message is the number of bits in the message (the empty message has length 0). If the number of bits in a message is a multiple of 8, for compactness we can represent the message in hex. The purpose of message padding is to make the total length of a padded message a multiple of 512. The SHA-1 sequentially processes blocks of 512 bits when computing the message digest. The following specifies how this padding shall be performed. As a summary, a "1" followed by m "0"s followed by a 64-bit integer are appended to the end of the message to produce a padded message of length 512 * n. The 64-bit integer is l, the length of the original message. The padded message is then processed by the SHA-1 as n 512-bit blocks.

Suppose a message has length $l < 2^{64}$. Before it is input to the SHA-1, the message is padded on the right as follows:

a. "1" is appended. **Example:** if the original message is "01010000", this is padded to "010100001".

b. "0"s are appended. The number of "0"s will depend on the original length of the message. The last 64 bits of the last 512-bit block are reserved for the length l of the original message.

Example: Suppose the original message is the bit string

01100001 01100010 01100011 01100100 01100101.

After step (a) this gives

01100001 01100010 01100011 01100100 01100101 1.

Since l = 40, the number of bits in the above is 41 and 407 "0"s are appended, making the total now 448. This gives (in hex)

61626364 65800000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000.

c. Obtain the 2-word representation of l, the number of bits in the original message. If l < 2^{32} then the first word is all zeroes. Append these two words to the padded message.

Example: Suppose the original message is as in (b). Then l = 40 (note that l is computed before any padding). The two-word representation of 40 is hex 00000000 00000028. Hence the final padded message is hex

61626364 65800000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000028.

The padded message will contain 16 * n words for some n > 0. The padded message is regarded as a sequence of n blocks M_1 , M_2, ... , M_n, where each M_i contains 16 words and M_1 contains the first characters (or bits) of the message.

9.3.3 functions used

A sequence of logical functions f_0, f_1,..., f_{79} is used in the SHA-1. Each f_t, 0 <= t <= 79, operates on three 32-bit words B, C, D and produces a 32-bit word as output. f_t(B,C,D) is defined as follows: for words B, C, D,

f_t(B,C,D) = (B AND C) OR ((NOT B) AND D) (0 <= t <= 19)

f_t(B,C,D) = B XOR C XOR D (20 <= t <= 39)

f_t(B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <= 59)

f_t(B,C,D) = B XOR C XOR D (60 <= t <= 79).

9.3.4 constants used

A sequence of constant words K(0), K(1), ... , K(79) is used in the SHA-1. In hex these are given by

K = 5A827999 (0 <= t <= 19)

K_t = 6ED9EBA1 (20 <= t <= 39)

K_t = 8F1BBCDC (40 <= t <= 59)

K_t = CA62C1D6 (60 <= t <= 79)

9.3.5 SHA-1 pseudocode

```
// initialize variables
h0 = 0x67452301
h1 = 0xEFCDAB89
h2 = 0x98BADCFE
h3 = 0x10325476
h4 = 0xC3D2E1F0

// pre-processing:
append the bit '1' to the message
append 0 ≤ k < 512 bits '0', so that the resulting message length (in bits)
   is congruent to 448 ≡ −64 (mod 512)
append length of message (before pre-processing), in bits, as 64-bit big-endian integer

// Process the message in successive 512-bit chunks:
// break message into 512-bit chunks
for each chunk
    break chunk into sixteen 32-bit big-endian words w[i], 0 ≤ i ≤ 15

    // Extend the sixteen 32-bit words into eighty 32-bit words:
    for i from 16 to 79
        w[i] = (w[i-3] xor w[i-8] xor w[i-14] xor w[i-16]) leftrotate 1

    // Initialize hash value for this chunk:
    a = h0
    b = h1
    c = h2
    d = h3
    e = h4

    Main loop:
    for i from 0 to 79
        if 0 ≤ i ≤ 19 then
            f = (b and c) or ((not b) and d)
            k = 0x5A827999
        else if 20 ≤ i ≤ 39
            f = b xor c xor d
            k = 0x6ED9EBA1
        else if 40 ≤ i ≤ 59
            f = (b and c) or (b and d) or (c and d)
            k = 0x8F1BBCDC
        else if 60 ≤ i ≤ 79
            f = b xor c xor d
            k = 0xCA62C1D6

        temp = (a leftrotate 5) + f + e + k + w[i]
        e = d
        d = c
        c = b leftrotate 30
        b = a
        a = temp
    // Add this chunk's hash to result so far:
    h0 = h0 + a
    h1 = h1 + b
    h2 = h2 + c
    h3 = h3 + d
    h4 = h4 + e

Produce the final hash value (big-endian):
digest = hash = h0 append h1 append h2 append h3 append h4
```

9.3.6 one simple operational example

Let the message be the ASCII binary-coded form of "abc", i.e.,

01100001 01100010 01100011.

This message has length l = 24. In step (a) of Section 4, we append "1". In step (b) we append 423 "0"s. In step (c) we append hex 00000000 00000018, the 2-word representation of 24. Thus the final padded message consists of one block, so that n = 1 in the notation of Section 4.

The initial hex values of {H_i} are

H_0 = 67452301
H_1 = EFCDAB89
H_2 = 98BADCFE
H_3 = 10325476
H_4 = C3D2E1F0.

Start processing block 1. The words of block 1 are

```
W[0] = 61626380
W[1] = 00000000
W[2] = 00000000
W[3] = 00000000
W[4] = 00000000
W[5] = 00000000
W[6] = 00000000
W[7] = 00000000
W[8] = 00000000
W[9] = 00000000
W[10] = 00000000
W[11] = 00000000
W[12] = 00000000
W[13] = 00000000
W[14] = 00000000
W[15] = 00000018.
```

The hex values of A,B,C,D,E after pass t of the "for t = 0 to 79" loop (step (d) of Section 7 or step (c) of Section 8) are

```
             A            B            C            D            E

t =  0: 0116FC33    67452301    7BF36AE2    98BADCFE    10325476
t =  1: 8990536D    0116FC33    59D148C0    7BF36AE2    98BADCFE
t =  2: A1390F08    8990536D    C045BF0C    59D148C0    7BF36AE2
t =  3: CDD8E11B    A1390F08    626414DB    C045BF0C    59D148C0
t =  4: CFD499DE    CDD8E11B    284E43C2    626414DB    C045BF0C
```

```
t =  5: 3FC7CA40    CFD499DE    F3763846    284E43C2    626414DB
t =  6: 993E30C1    3FC7CA40    B3F52677    F3763846    284E43C2
t =  7: 9E8C07D4    993E30C1    0FF1F290    B3F52677    F3763846
t =  8: 4B6AE328    9E8C07D4    664F8C30    0FF1F290    B3F52677
t =  9: 8351F929    4B6AE328    27A301F5    664F8C30    0FF1F290
t = 10: FBDA9E89    8351F929    12DAB8CA    27A301F5    664F8C30
t = 11: 63188FE4    FBDA9E89    60D47E4A    12DAB8CA    27A301F5
t = 12: 4607B664    63188FE4    7EF6A7A2    60D47E4A    12DAB8CA
t = 13: 9128F695    4607B664    18C623F9    7EF6A7A2    60D47E4A
t = 14: 196BEE77    9128F695    1181ED99    18C623F9    7EF6A7A2
t = 15: 20BDD62F    196BEE77    644A3DA5    1181ED99    18C623F9
t = 16: 4E925823    20BDD62F    C65AFB9D    644A3DA5    1181ED99
t = 17: 82AA6728    4E925823    C82F758B    C65AFB9D    644A3DA5
t = 18: DC64901D    82AA6728    D3A49608    C82F758B    C65AFB9D
t = 19: FD9E1D7D    DC64901D    20AA99CA    D3A49608    C82F758B
t = 20: 1A37B0CA    FD9E1D7D    77192407    20AA99CA    D3A49608
t = 21: 33A23BFC    1A37B0CA    7F67875F    77192407    20AA99CA
t = 22: 21283486    33A23BFC    868DEC32    7F67875F    77192407
t = 23: D541F12D    21283486    0CE88EFF    868DEC32    7F67875F
t = 24: C7567DC6    D541F12D    884A0D21    0CE88EFF    868DEC32
t = 25: 48413BA4    C7567DC6    75507C4B    884A0D21    0CE88EFF
t = 26: BE35FBD5    48413BA4    B1D59F71    75507C4B    884A0D21
t = 27: 4AA84D97    BE35FBD5    12104EE9    B1D59F71    75507C4B
t = 28: 8370B52E    4AA84D97    6F8D7EF5    12104EE9    B1D59F71
t = 29: C5FBAF5D    8370B52E    D2AA1365    6F8D7EF5    12104EE9
t = 30: 1267B407    C5FBAF5D    A0DC2D4B    D2AA1365    6F8D7EF5
t = 31: 3B845D33    1267B407    717EEBD7    A0DC2D4B    D2AA1365
t = 32: 046FAA0A    3B845D33    C499ED01    717EEBD7    A0DC2D4B
t = 33: 2C0EBC11    046FAA0A    CEE1174C    C499ED01    717EEBD7
t = 34: 21796AD4    2C0EBC11    811BEA82    CEE1174C    C499ED01
t = 35: DCBBB0CB    21796AD4    4B03AF04    811BEA82    CEE1174C
t = 36: 0F511FD8    DCBBB0CB    085E5AB5    4B03AF04    811BEA82
t = 37: DC63973F    0F511FD8    F72EEC32    085E5AB5    4B03AF04
t = 38: 4C986405    DC63973F    03D447F6    F72EEC32    085E5AB5
t = 39: 32DE1CBA    4C986405    F718E5CF    03D447F6    F72EEC32
t = 40: FC87DEDF    32DE1CBA    53261901    F718E5CF    03D447F6
t = 41: 970A0D5C    FC87DEDF    8CB7872E    53261901    F718E5CF
t = 42: 7F193DC5    970A0D5C    FF21F7B7    8CB7872E    53261901
t = 43: EE1B1AAF    7F193DC5    25C28357    FF21F7B7    8CB7872E
t = 44: 40F28E09    EE1B1AAF    5FC64F71    25C28357    FF21F7B7
t = 45: 1C51E1F2    40F28E09    FB86C6AB    5FC64F71    25C28357
t = 46: A01B846C    1C51E1F2    503CA382    FB86C6AB    5FC64F71
t = 47: BEAD02CA    A01B846C    8714787C    503CA382    FB86C6AB
t = 48: BAF39337    BEAD02CA    2806E11B    8714787C    503CA382
t = 49: 120731C5    BAF39337    AFAB40B2    2806E11B    8714787C
t = 50: 641DB2CE    120731C5    EEBCE4CD    AFAB40B2    2806E11B
t = 51: 3847AD66    641DB2CE    4481CC71    EEBCE4CD    AFAB40B2
t = 52: E490436D    3847AD66    99076CB3    4481CC71    EEBCE4CD
t = 53: 27E9F1D8    E490436D    8E11EB59    99076CB3    4481CC71
t = 54: 7B71F76D    27E9F1D8    792410DB    8E11EB59    99076CB3
t = 55: 5E6456AF    7B71F76D    09FA7C76    792410DB    8E11EB59
t = 56: C846093F    5E6456AF    5EDC7DDB    09FA7C76    792410DB
t = 57: D262FF50    C846093F    D79915AB    5EDC7DDB    09FA7C76
t = 58: 09D785FD    D262FF50    F211824F    D79915AB    5EDC7DDB
t = 59: 3F52DE5A    09D785FD    3498BFD4    F211824F    D79915AB
t = 60: D756C147    3F52DE5A    4275E17F    3498BFD4    F211824F
t = 61: 548C9CB2    D756C147    8FD4B796    4275E17F    3498BFD4
t = 62: B66C020B    548C9CB2    F5D5B051    8FD4B796    4275E17F
t = 63: 6B61C9E1    B66C020B    9523272C    F5D5B051    8FD4B796
```

```
t = 64: 19DFA7AC    6B61C9E1    ED9B0082    9523272C    F5D5B051
t = 65: 101655F9    19DFA7AC    5AD87278    ED9B0082    9523272C
t = 66: 0C3DF2B4    101655F9    0677E9EB    5AD87278    ED9B0082
t = 67: 78DD4D2B    0C3DF2B4    4405957E    0677E9EB    5AD87278
t = 68: 497093C0    78DD4D2B    030F7CAD    4405957E    0677E9EB
t = 69: 3F2588C2    497093C0    DE37534A    030F7CAD    4405957E
t = 70: C199F8C7    3F2588C2    125C24F0    DE37534A    030F7CAD
t = 71: 39859DE7    C199F8C7    8FC96230    125C24F0    DE37534A
t = 72: EDB42DE4    39859DE7    F0667E31    8FC96230    125C24F0
t = 73: 11793F6F    EDB42DE4    CE616779    F0667E31    8FC96230
t = 74: 5EE76897    11793F6F    3B6D0B79    CE616779    F0667E31
t = 75: 63F7DAB7    5EE76897    C45E4FDB    3B6D0B79    CE616779
t = 76: A079B7D9    63F7DAB7    D7B9DA25    C45E4FDB    3B6D0B79
t = 77: 860D21CC    A079B7D9    D8FDF6AD    D7B9DA25    C45E4FDB
t = 78: 5738D5E1    860D21CC    681E6DF6    D8FDF6AD    D7B9DA25
t = 79: 42541B35    5738D5E1    21834873    681E6DF6    D8FDF6AD.

    Block 1 has been processed. The values of {Hi} are

H0 = 67452301 + 42541B35 = A9993E36
H1 = EFCDAB89 + 5738D5E1 = 4706816A
H2 = 98BADCFE + 21834873 = BA3E2571
H3 = 10325476 + 681E6DF6 = 7850C26C
H4 = C3D2E1F0 + D8FDF6AD = 9CD0D89D.

Message digest = A9993E36 4706816A BA3E2571 7850C26C 9CD0D89D
```

9.3.7 cryptanalysis

In early 2005, Rijmen and Oswald published an attack on a reduced version of SHA-1—53 out of 80 rounds—which finds collisions with a computational effort of fewer than 2^{80} operations.

In February 2005, an attack by Xiaoyun Wang, Yiqun Lisa Yin, Bayarjargal, and Hongbo Yu was announced. The attacks can find collisions in the full version of SHA-1, requiring fewer than 2^{69} operations. (A brute-force search would require 2^{80} operations.)

The authors write: "In particular, our analysis is built upon the original differential attack on SHA-0 [*sic*], the near collision attack on SHA-0, the multiblock collision techniques, as well as the message modification techniques used in the collision search attack on MD5. Breaking SHA-1 would not be possible without these powerful analytical techniques." The authors have presented a collision for 58-round SHA-1, found with 2^{33} hash operations. The paper with the full attack description was published in August 2005 at the CRYPTO conference.

In an interview, Yin states that, "Roughly, we exploit the following two weaknesses: One is that the file preprocessing step is not complicated enough; another is that certain math operations in the first 20 rounds have unexpected security problems."

On 17 August 2005, an improvement on the SHA-1 attack was announced on behalf of Xiaoyun Wang, Andrew Yao and Frances Yao at the CRYPTO 2005 rump session, lowering the complexity required for finding a collision in SHA-1 to 2^{63}. On 18 December 2007 the details of this result were explained and verified by Martin Cochran.

Christophe De Cannière and Christian Rechberger further improved the attack on SHA-1 in "Finding SHA-1 Characteristics: General Results and Applications," receiving the Best Paper Award at ASIACRYPT 2006. A two-block collision for 64-round SHA-1 was presented, found using unoptimized methods with 2^{35} compression function evaluations. As this attack requires the equivalent of about 2^{35} evaluations, it is considered to be a significant theoretical break. In order to find an actual collision in the full 80 rounds of the hash function, however, massive amounts of computer time are required. To that end, a collision search for

SHA-1 using the distributed computing platform BOINC began August 8, 2007, organized by the Graz University of Technology. The effort was abandoned May 12, 2009 due to lack of progress.

At the Rump Session of CRYPTO 2006, Christian Rechberger and Christophe De Cannière claimed to have discovered a collision attack on SHA-1 that would allow an attacker to select at least parts of the message.

Cameron McDonald, Philip Hawkes and Josef Pieprzyk presented a hash collision attack with claimed complexity 2^{52} at the Rump session of Eurocrypt 2009. However, the accompanying paper, "Differential Path for SHA-1 with complexity O(2^{52})" has been withdrawn due to the authors' discovery that their estimate was incorrect.

The first public collision (SHAttered) was announced in Feb 2017, by CWI (Centrum Wiskunde & Informatica). Two different pdf files, having the same digest were generated, in about 2^68 SHA_1 evaluations. the attack required the equivalent of 6 500 years of single CPU computations.

9.4 the SHA-2 family

NIST published four additional hash functions in the SHA family, named after their digest lengths (in bits): SHA-224, SHA-256, SHA-384, and SHA-512. The algorithms are collectively known as SHA-2.

The algorithms were first published in 2001 in the draft FIPS PUB 180-2, at which time review and comment were accepted. FIPS PUB 180-2, which also includes SHA-1, was released as an official standard in 2002. In February 2004, a change notice was published for FIPS PUB 180-2, specifying an additional variant, SHA-224, defined to match the key length of two-key Triple DES. These variants are patented in US patent 6829355. The United States has released the patent under a royalty free license.

SHA-256 and SHA-512 are novel hash functions computed with 32- and 64-bit words, respectively. They use different shift amounts and additive constants, but their structures are otherwise virtually identical, differing only in the number of rounds. SHA-224 and SHA-384 are simply truncated versions of the first two, computed with different initial values.

Unlike SHA-1, the SHA-2 functions are not widely used, despite their better security. Reasons might include lack of support for SHA-2 on systems running Windows XP SP2 or older, a lack of perceived urgency since SHA-1 collisions have not yet been found, or a desire to wait until SHA-3 is standardized. SHA-256 is used to authenticate Debian Linux software packages and in the DKIM message signing standard; SHA-512 is part of a system to authenticate archival video from the International Criminal Tribunal of the Rwandan genocide. SHA-256 and SHA-512 are proposed for use in DNSSEC NIST's directive that U.S. government agencies stop most uses of SHA-1 after 2010, and the completion of SHA-3, may accelerate migration away from SHA-1.

Currently, the best public attacks on SHA-2 break 24 of the 64 or 80 rounds.

9.5 SHA-3

9.5.1 general description

SHA-3 (**Secure Hash Algorithm 3**) is the latest member of the Secure Hash Algorithm family of standards, released by NIST on August 5, 2015. Although part of the same series of standards, SHA-3 is internally different from the MD5-like structure of SHA-1 and SHA-2.

SHA-3 is a subset of the broader cryptographic primitive family **Keccak**, designed by Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche, building upon RadioGatún. Keccak's authors have proposed additional uses for the function, not (yet) standardized by NIST, including a stream cipher, an authenticated encryption system, a "tree" hashing scheme for faster hashing on certain architectures, and AEAD ciphers Keyak and Ketje.

Keccak is based on a novel approach called sponge construction. Sponge construction is based on a wide random function or random permutation, and allows inputting ("absorbing" in sponge terminology) any amount of data, and outputting ("squeezing") any amount of data, while acting as a pseudorandom function with regard to all previous inputs. This leads to great flexibility.

9.5.2 history

An open competition for a new SHA-3 function was formally announced in the *Federal Register* on November 2, 2007. "NIST is initiating an effort to develop one or more additional hash algorithms through a public competition, similar to the development process for the Advanced Encryption Standard (AES)." Submissions were due October 31, 2008 and the proclamation of a winner and publication of the new standard took place in 2012.

NIST selected 51 entries for the Round 1, and 14 of them advanced to Round 2.

The following hash function submissions have been accepted for Round Two.

- BLAKE
- Blue Midnight Wish
- CubeHash
- ECHO (France Telecom)
- Fugue
- Grøstl (Knudsen et al.)
- Hamsi
- JH
- Keccak (Keccak team, Daemen et al.)
- Luffa
- Shabal
- SHAvite-3
- SIMD
- Skein (Schneier et al.)

In October 2012 the Keccak algorithm has been declared the winner..

9.5.3 algorithm description

SHA-3 uses the sponge construction, in which data is "absorbed" into the sponge, then the result is "squeezed" out. In the absorbing phase, message blocks are XORed into a subset of the state, which is then transformed as a whole using a permutation function *f*. In the "squeeze" phase, output blocks are read from the same subset of the state, alternated with the state transformation function *f*. The size of the part of the state that is written and read is called the "rate" (denoted *r*), and the size of the part that is untouched by input/output is called the "capacity" (denoted *c*). The capacity determines the security of the scheme. The maximum security level is half the capacity.

9.5.4 some examples

SHA3-256("") = a7ffc6f8bf1ed76651c14756a061d662f580ff4de43b49fa82d80a4b80f8434a

SHA3-384("") = 0c63a75b845e4f7d01107d852e4c2485c51a50aaaa94fc61995e71bbee983a2ac3713831264adb47fb6bd1e058d5f004

SHA3-512("") = a69f73cca23a9ac5c8b567dc185a756e97c982164fe25859e0d1dcc1475c80a615b2123af1f5f94c11e3e9402c3ac558f500199d95b6d3e301758586281dcd26

9.5.5 cryptanalysis

There is a general result (Grover's algorithm) that quantum computers can perform a structured preimage attack in sqrt(2^d), while a classical brute-force attack needs 2^*d*. A structured preimage attack implies a second preimage attack and thus a collision attack. A quantum computer can also perform a birthday attack, thus break collision resistance, in 2^{d/3} (although that is disputed). Noting that the maximum strength can be , this gives the following upper bounds on the quantum security of SHA-3:

Instance	***Security strengths in bits***			
	Collision (Brassard et al.)	***Collision (Bernstein)***	***Preimage***	***2nd preimage***
SHA3-224(*M*)	74⅔	112	112	112
SHA3-256(*M*)	85⅓	128	128	128
SHA3-384(*M*)	128	192	192	192
SHA3-512(*M*)	170⅔	256	256	256
SHAKE128(*M*, *d*)	min(*d*/3,128)	min(*d*/2,128)	≥min(*d*/2,128)	min(*d*/2,128)
SHAKE256(*M*, *d*)	min(*d*/3,256)	min(*d*/2,256)	≥min(*d*/2,256)	min(*d*/2,256)

chapter 10 encryption systems

10.1 what is encryption?

Encryption is the conversion of electronic data into another form, called ciphertext, which cannot be easily understood by anyone except authorized parties.

Computer encryption is based on the science of cryptography, which has been used as long as humans have wanted to keep information secret. Before the digital age, the biggest users of cryptography were governments, particularly for military purposes.

The Greek historian Plutarch wrote, for example, about Spartan generals who sent and received sensitive messages using a scytale, a thin cylinder made out of wood. The general would wrap a piece of parchment around the scytale and write his message along its length. When someone removed the paper from the cylinder, the writing appeared to be a jumble of nonsense. But if the other general receiving the parchment had a scytale of similar size, he could wrap the paper around it and easily read the intended message.

The Greeks were also the first to use ciphers, specific codes that involve substitutions or transpositions of letters and numbers.

As long as both generals had the correct cipher, they could decode any message the other sent. To make the message more difficult to decipher, they could arrange the letters inside the grid in any combination.

Most forms of cryptography in use these days rely on computers, simply because a human-based code is too easy for a computer to crack. Ciphers are also better known today as algorithms, which are the guides for encryption -- they provide a way in which to craft a message and give a certain range of possible combinations. A key, on the other hand, helps a person or computer figure out the one possibility on a given occasion.

Computer encryption systems generally belong in one of two categories:

1. Symmetric-key encryption
2. Public-key encryption

10.2 types of encryption

10.2.1 symmetric key encryption

In symmetric-key schemes, the encryption and decryption keys are the same. Thus communicating parties must have the same key before they can achieve secret communication.

10.2.2 public key encryption

In public-key encryption schemes, the encryption key is published for anyone to use and encrypt messages. However, only the receiving party has access to the decryption key that enables messages to be read. Public-key encryption was first described in a secret document in 1973; before then all encryption schemes were symmetric-key (also called private-key).

A publicly available public key encryption application called Pretty Good Privacy (PGP) was written in 1991 by Phil Zimmermann, and distributed free of charge with source code; it was purchased by Symantec in 2010 and is regularly updated.

10.3 DES – Data Encryption Standard

The Data Encryption Standard (DES) is an outdated symmetric-key method of data encryption.

DES works by using the same key to encrypt and decrypt a message, so both the sender and the receiver must know and use the same private key. Once the go-to, symmetric-key algorithm for the encryption of electronic data, DES has been superseded by the more secure Advanced Encryption Standard (AES) algorithm.

10.3.1 general description

DES is a symmetric encryption algorithm which uses a common, 64-bit long key, for both encryption and decryption. Out of the 64 bits of the key, 56 bits are independent, while the remaining 8 bits are parity bits.

It is a block algorithm, the size of the block being 64 bits.

DES was approved as a federal standard in November 1976, and published on 15 January 1977 as FIPS 46, authorized for use on all unclassified data. It was subsequently reaffirmed as the standard in 1983, 1988 (revised as FIPS-46-1), 1993 (FIPS-46-2), and again in 1999 (FIPS-46-3), the latter prescribing "Triple DES" (see below). On 26 May 2002, DES was finally superseded by the Advanced Encryption Standard (AES), following a public competition. On 19 May 2005, FIPS 46-3 was officially withdrawn, but NIST has approved Triple DES through the year 2030 for sensitive government information.

10.3.2 history

Originally designed by researchers at IBM in the early 1970s, DES was adopted by the U.S. government as an official Federal Information Processing Standard (FIPS) in 1977 for the encryption of commercial and sensitive yet unclassified government computer data. It was the first encryption algorithm approved by the U.S. government for public disclosure. This ensured that DES was quickly adopted by industries such as financial services, where the need for strong encryption is high. The simplicity of DES also saw it used in a wide variety of embedded systems, smart cards, SIM cards and network devices requiring encryption like modems, set-top boxes and routers.

10.3.3 algorithm description

DES takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bitstring of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits.

The key is nominally stored or transmitted as 8 bytes, each with odd parity. According to ANSI X3.92-1981 (Now, known as ANSI INCITS 92-1981), section 3.5:

One bit in each 8-bit byte of the *KEY* may be utilized for error detection in key generation, distribution, and storage. Bits 8, 16,..., 64 are for use in ensuring that each byte is of odd parity.

Like other block ciphers, DES by itself is not a secure means of encryption, but must instead be used in a mode of operation. FIPS-81 specifies several modes for use with DES. Further comments on the usage of DES are contained in FIPS-74.

Decryption uses the same structure as encryption, but with the keys used in reverse order. (This has the advantage that the same hardware or software can be used in both directions.)

10.3.4 cryptanalysis

For any cipher, the most basic method of attack is brute force - trying every possible key in turn. The length of the key determines the number of possible keys, and hence the feasibility of this approach. For

DES, questions were raised about the adequacy of its key size early on, even before it was adopted as a standard, and it was the small key size, rather than theoretical cryptanalysis, which dictated a need for a replacement algorithm. As a result of discussions involving external consultants including the NSA, the key size was reduced from 128 bits to 56 bits to fit on a single chip.

The EFF's US$250,000 DES cracking machine contained 1,856 custom chips and could brute force a DES key in a matter of days - the photo shows a DES Cracker circuit board fitted with several Deep Crack chips.

In academia, various proposals for a DES-cracking machine were advanced. In 1977, Diffie and Hellman proposed a machine costing an estimated US$20 million which could find a DES key in a single day. By 1993, Wiener had proposed a key-search machine costing US$1 million which would find a key within 7 hours. However, none of these early proposals were ever implemented—or, at least, no implementations were publicly acknowledged. The vulnerability of DES was practically demonstrated in the late 1990s. In 1997, RSA Security sponsored a series of contests, offering a $10,000 prize to the first team that broke a message encrypted with DES for the contest. That contest was won by the DESCHALL Project, led by Rocke Verser, Matt Curtin, and Justin Dolske, using idle cycles of thousands of computers across the Internet. The feasibility of cracking DES quickly was demonstrated in 1998 when a custom DES-cracker was built by the Electronic Frontier Foundation (EFF), a cyberspace civil rights group, at the cost of approximately US$250,000 (see EFF DES cracker). Their motivation was to show that DES was breakable in practice as well as in theory: "*There are many people who will not believe a truth until they can see it with their own eyes. Showing them a physical machine that can crack DES in a few days is the only way to convince some people that they really cannot trust their security to DES.*" The machine brute-forced a key in a little more than 2 days search.

The COPACOBANA machine, built in 2006 for US$10,000 by the Universities of Bochum and Kiel, Germany,[16] contains 120 low-cost FPGAs and could perform an exhaustive key search on DES in 9 days on average. The photo shows the backplane of the machine with the FPGAs.

The only other confirmed DES cracker was the COPACOBANA machine built in 2006 by teams of the Universities of Bochum and Kiel, both in Germany. Unlike the EFF machine, COPACOBANA consists of commercially available, reconfigurable integrated circuits. 120 of these Field-programmable gate arrays (FPGAs) of type XILINX Spartan3-1000 run in parallel. They are grouped in 20 DIMM modules, each containing 6 FPGAs. The use of reconfigurable hardware makes the machine applicable to other code breaking tasks as well. The figure shows a full-sized COPACOBANA. One of the more interesting aspects of COPACOBANA is its cost factor. One machine can be built for approximately $10,000. The cost decrease by roughly a factor of 25 over the EFF machine is an impressive example for the continuous improvement of digital hardware. Adjusting for inflation over 8 years yields an even higher improvement of about 30x. Since 2007, SciEngines GmbH, a spin-off company of the two project partners of COPACOBANA has enhanced and developed successors of COPACOBANA. In 2008 their COPACOBANA RIVYERA reduced the time to break DES to less than one day, using 128 Spartan-3 5000's.

10.4 AES – Advanced Encryption Standard

Advanced Encryption Standard (**AES**) is an encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as **Rijndael.** Each AES cipher has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor, the Data Encryption Standard (DES).

AES was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001 after a 5-year standardization process in which fifteen competing designs were presented and evaluated before Rijndael was selected as the most suitable (see Advanced Encryption Standard process for more details). It became effective as a Federal government standard on May 26, 2002 after approval by the Secretary of Commerce. It is available in many different encryption packages. AES is the first publicly accessible and open cipher approved by the NSA for top secret information (see Security of AES, below).

The Rijndael cipher was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, and submitted by them to the AES selection process. Rijndael (pronounced [rɛindaːl]) is a portmanteau of the names of the two inventors.

10.4.1 general description

AES is based on a design principle known as a Substitution permutation network. It is fast in both software and hardware. Unlike its predecessor, DES, AES does not use a Feistel network.

AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits.

AES operates on a 4×4 array of bytes, termed the *state* (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field.

The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

10.4.2 history

The origins of AES date back to 1997 when the National Institute of Standards and Technology (NIST) announced that it needed a successor to the aging Data Encryption Standard (DES) which was becoming vulnerable to brute-force attacks.

AES has been adopted by the U.S. government. It supersedes the Data Encryption Standard (DES), which was published in 1977. The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data.

In the United States, AES was announced by the NIST as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001. This announcement followed a five-year standardization process in which fifteen competing designs were presented and evaluated, before the Rijndael cipher was selected as the most suitable

10.4.3 algorithm description

AES is based on a design principle known as a substitution–permutation network, and is efficient in both software and hardware. Unlike its predecessor DES, AES does not use a Feistel network. AES is a variant of Rijndael, with a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. By contrast, Rijndael *per se* is specified with block and key sizes that may be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits.

AES operates on a 4 × 4 column-major order array of bytes, termed the *state*.[note 3] Most AES calculations are done in a particular finite field.

For instance, 16 bytes, are represented as this two-dimensional array:

The key size used for an AES cipher specifies the number of transformation rounds that convert the input, called the plaintext, into the final output, called the ciphertext. The number of rounds are as follows:

- 10 rounds for 128-bit keys.
- 12 rounds for 192-bit keys.
- 14 rounds for 256-bit keys.

Each round consists of several processing steps, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

Main functions

1. KeyExpansion – round keys are derived from the cipher key using the AES key schedule. AES requires a separate 128-bit round key block for each round plus one more.

2. Initial round key addition:

1.AddRoundKey – each byte of the state is combined with a byte of the round key using bitwise xor.

3. 9, 11 or 13 rounds:

1. SubBytes – a non-linear substitution step where each byte is replaced with another according to a lookup table.

2.ShiftRows – a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

3.MixColumns – a linear mixing operation which operates on the columns of the state, combining the four bytes in each column.

4.AddRoundKey

4.Final round (making 10, 12 or 14 rounds in total):

1.SubBytes

2.ShiftRows

3.AddRoundKey

10.4.4 cryptanalysis

Until May 2009, the only successful published attacks against the full AES were side-channel attacks on some specific implementations. In 2009, a new related-key attack was discovered that exploits the simplicity of AES's key schedule and has a complexity of 2^119. In December 2009 it was improved to 2^99.5. This is a follow-up to an attack discovered earlier in 2009 by Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolić, with a complexity of 2^96 for one out of every 2^35 keys. However, related-key attacks are not of concern in any properly designed cryptographic protocol, as a properly designed protocol (i.e., implementational software) will take care not to allow related keys, essentially by constraining an attacker's means of selecting keys for relatedness.

Another attack was blogged by Bruce Schneier on July 30, 2009, and released as a preprint on August 3, 2009. This new attack, by Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir, is against AES-256 that uses only two related keys and 2^39 time to recover the complete 256-bit key of a 9-round version, or 2^45 time for a 10-round version with a stronger type of related subkey attack, or 2^70 time for an 11-round version. 256-bit AES uses 14 rounds, so these attacks are not effective against full AES.

The practicality of these attacks with stronger related keys has been criticized, for instance, by the paper on chosen-key-relations-in-the-middle attacks on AES-128 authored by Vincent Rijmen in 2010.

In November 2009, the first known-key distinguishing attack against a reduced 8-round version of AES-128 was released as a preprint. This known-key distinguishing attack is an improvement of the rebound, or the start-from-the-middle attack, against AES-like permutations, which view two consecutive rounds of permutation as the application of a so-called Super-S-box. It works on the 8-round version of AES-128, with a time complexity of 248, and a memory complexity of 232. 128-bit AES uses 10 rounds, so this attack is not effective against full AES-128.

10.5 the Diffie-Hellman key exchange algorithm

Diffie-Hellman key exchange, also called exponential key exchange, is a method of digital encryption that uses a number raised to specific powers to produce decryption keys that are never directly transmitted, making the task of a would-be code breaker mathematically overwhelming.

To implement Diffie-Hellman, the two end users Alice and Bob, while communicating over a channel they know to be private, mutually agree on positive whole numbers p and q, such that p is a prime number and q is a generator of p. The generator q is a number that, when raised to positive whole-number powers less than p, never produces the same result for any two such whole numbers. The value of p may be large but the value of q is usually small.

Once Alice and Bob have agreed on p and q in private, they choose positive whole-number personal keys a and b, both less than the prime-number modulus p. Neither user divulges their personal key to anyone; ideally they memorize these numbers and do not write them down or store them anywhere. Next, Alice and Bob compute public keys a* and b* based on their personal keys according to the formulas

a* = q^a mod p

and

b* = q^b mod p

The two users can share their public keys a* and b* over a communications medium assumed to be insecure, such as the Internet or a corporate wide area network (WAN). From these public keys, a number x can be generated by either user on the basis of their own personal keys. Alice computes x using the formula

10.6 RSA

RSA is a cryptosystem for public-key encryption, and is widely used for securing sensitive data, particularly when being sent over an insecure network such as the Internet.

RSA was first described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman of the Massachusetts Institute of Technology. Public-key cryptography, also known as asymmetric cryptography, uses two different but mathematically linked keys, one public and one private. The public key can be shared with everyone, whereas the private key must be kept secret. In RSA cryptography, both the public and the private keys can encrypt a message; the opposite key from the one used to encrypt a message is used to decrypt it. This attribute is one reason why RSA has become the most widely used asymmetric algorithm: It provides a method of assuring the confidentiality, integrity, authenticity and non-reputability of electronic communications and data storage.

Many protocols like SSH, OpenPGP, S/MIME, and SSL/TLS rely on RSA for encryption and digital signature functions. It is also used in software programs -- browsers are an obvious example, which need to establish a secure connection over an insecure network like the Internet or validate a digital signature. RSA signature verification is one of the most commonly performed operations in IT.

10.7 the encryption nowadays

Few organizations today have access to truly private and secure networks; instead, they share network infrastructure with other organizations. As a result, information traveling over these public or virtual private networks is often vulnerable to interception. Quite rightly, many of today’s data privacy requirements and standards include, as a baseline level of protection, a mandate to protect data in motion. While organizations can choose to encrypt selected data at the application level or within databases or other storage environments, the bulk protection of data flowing over a network provides a blunt but very effective instrument for adding an extra layer of security. Network encryption guards against regulated data inadvertently being sent in the clear and also provides valuable protection for all other classes of data that perhaps do not justify dedicated protection but nonetheless are still considered sensitive. Although network-level encryption is a relatively mature technology, organizations need to make several choices when deciding what kind of network encryption to deploy

Standalone network encryption platforms are particularly valuable for high-speed connections between data centers. Globally interconnected organizations and service providers require the combination of optimized bandwidth, unshakeable resilience, and security for critical systems such as storage area networks (SANs), transaction systems, and cloud computing. The ability to secure these backbone connections as transparently as possible becomes a critical success factor for enterprises and a valuable differentiator for network service providers.

10.7.1 which networks should have their traffic encrypted?

Most networks are 'open' to some degree, but some are much more open than others. Internal wired networks might be considered vulnerable only for the most sensitive data, since they still suffer from the threat of insider attacks, whereas backbone networks and wide area network (WAN) connections usually deserve more consideration as they typically use shared pipes from external service providers. In almost all settings, organizations will want to encrypt traffic over wireless local area networks (LANs), wireless WANs, and, of course, the Internet. This page focuses mainly on WAN encryption.

10.7.2 should traffic be encrypted at Layer 2 or Layer 3 in the OSI Network Model?

At stake in this choice are overhead and the potential waste of bandwidth. Applying encryption at Layer 3, using well-known protocols such as IPsec, creates the need to preserve routing information used by equipment throughout the network. This imposes a significant overhead, ultimately affecting capacity and latency. Layer 2 encryption operates at a lower layer and is independent of the routing information and flow-management techniques that exist at Layer 3, and is more efficient in most cases. That said, IPsec, remains the most common form of network encryption for all but high-speed data-center-to-data-center connections where bandwidth and latency are most critical.

10.7.3 are your security needs best served by embedded or standalone encryption?

Since network-level encryption is a relatively mature technology, it is commonly available as an embedded or native feature of routing or switching equipment. Standalone encryption platforms provide an alternative to embedded encryption—one that delivers a higher level of assurance and benefits from purpose-built key management capabilities. Standalone encryption platforms are independently certified against security benchmarks such as FIPS 140 and Common Criteria, offer tamper resistance, and offer features that enable organizations to enforce a strong separation of duties between network administrators and security officers.

10.8 risks associated with network security

Attackers can "eavesdrop" on unencrypted data traveling over a network, not only impacting privacy but potentially opening the potential to modify or substitute data as a way to stage more sophisticated attacks.

Because industry mandates often require protection for data in motion, organizations that do not implement this protection risk fines, embarrassing data breach disclosure statements, and resulting damage to their reputation.

Depending on the application, encryption capabilities embedded in routers and switches may not offer the combination of security and performance you need.

10.9 example of one country which uses encryption systems

The National Security Agency (NSA) is an intelligence organization of the United States government, responsible for global monitoring, collection, and processing of information and data for foreign intelligence and counterintelligence purposes - a discipline known as Signals intelligence (SIGINT). NSA is concurrently charged with protection of U.S. government communications and information systems against penetration and network warfare.

The large number of encryption systems that NSA has developed can be grouped by application.

10.9.1 record traffic encryption

During World War II, written messages (known as record traffic) were encrypted off line on special, and highly secret, rotor machines and then transmitted in five letter code groups using Morse code or teletypewriter circuits, to be decrypted off-line by similar machines at the other end. The SIGABA rotor machine, developed during this era continued to be used until the mid-1950s, when it was replaced by the KL-7, which had more rotors.

The KW-26 ROMULUS was a second generation encryption system in wide use that could be inserted into teletypewriter circuits so traffic was encrypted and decrypted automatically. It used electronic shift registers instead of rotors and became very popular (for a COMSEC device of its era), with over 14,000 units produced. It was replaced in the 1980s by the more compact KG-84, which in turn was superseded by the KG-84-interoperable KIV-7.

10.9.2 fleet broadcast

U.S. Navy ships traditionally avoid using their radios to prevent adversaries from locating them by direction finding. The Navy also needs to maintain traffic security, so it has radio stations constantly broadcasting a stream of coded messages. During and after World War II, Navy ships copied these fleet broadcasts and used specialized call sign encryption devices to figure out which messages were intended for them. The messages would then be decoded off line using SIGABA or KL-7 equipment.

The second generation KW-37 automated monitoring of the fleet broadcast by connecting in line between the radio receiver and a teleprinter. It, in turn, was replaced by the more compact and reliable third generation KW-46.

10.9.3 internet

NSA has approved a variety of devices for securing Internet Protocol communications. These have been used to secure the Secret Internet Protocol Router Network (SIPRNet), among other uses.

The first commercial network layer encryption device was the Motorola Network Encryption System (NES). The system used the SP3 and KMP protocols defined by the NSA Secure Data Network System (SDNS) and were the direct precursors to IPsec. The NES was built in a three part architecture that used a small cryptographic security kernel to separate the trusted and untrusted network protocol stacks.

The SDNS program defined a Message Security Protocol (MSP) that was built on the use X.509 defined certificates. The first NSA hardware built for this application was the BBN Safekeeper. The Message Security Protocol was a precursor to the IETF Privacy Enhance Mail (PEM) protocol. The BBN Safekeeper provided a high degree of tamper resistance and was one of the first devices used by commercial PKI companies. iteration

10.10 some examples of the encryption on the internet

10.10.1 HTTPS

HTTPS is a communications protocol for secure communication over a computer network, with especially wide deployment on the Internet. Technically, it is not a protocol in and of itself; rather, it is the result of simply layering the Hypertext Transfer Protocol (HTTP) on top of the SSL or TLS protocol, thus adding the security capabilities of SSL/TLS to standard HTTP communications. The main motivation for HTTPS is to provide authentication of the visited website and to protect the privacy and integrity of exchanged data.The security of HTTPS is therefore that of the underlying TLS, which uses long-term public and secret keys to exchange a short term session key to encrypt the data flow between client and server. X.509 certificates are used to guarantee one is talking to the partner with whom one wants to talk. As a consequence, certificate authorities and a public key infrastructure are necessary to verify the relation between the owner of a certificate and the certificate, as well as to generate, sign, and administer the validity of certificates. While this can be more beneficial than verifying the identities via a web of trust, the 2013 mass surveillance disclosures made it more widely known that certificate authorities are a weak point from a security standpoint, allowing man-in-the-middle attacks. Another important property in this context is perfect forward secrecy (PFS), so the short-term session key cannot be derived from the long-term asymmetric secret key; however, PFS is not widely adopted.

10.10.2 TLS and SSL

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols designed to provide communications security over a computer network.

SSL is the secure communications protocol of choice for a large part of the Internet community. There are many applications of SSL in existence, since it is capable of securing any transmission over TCP. Secure HTTP, or HTTPS, is a familiar application of SSL in e-commerce or password transactions. According to the Internet Draft of the SSL Protocol, the point of the protocol "is to provide privacy and reliability between two communicating applications."

The protocol release further explains that three points combine to provide connection security. These points are:

- Privacy - connection through encryption
- Identity authentication – identification through certificates
- Reliability –dependable maintenance of a secure connection through message integrity checking.

10.10.3 PGP

Pretty Good Privacy (PGP) is a data encryption and decryption computer program that provides cryptographic privacy and authentication for data communication. PGP is often used for signing, encrypting, and decrypting texts, e-mails, files, directories, and whole disk partitions and to increase the security of e-mail communications. It was created by Phil Zimmermann in 1991.

PGP and similar software follow the OpenPGP standard (RFC 4880) for encrypting and decrypting data.

The program PGP it's a free software for non-commercial use and accessible in www.pgpi.com. It's the most used system all over the world by particular users and also big companies. The newest versions of the software are very easy to use and communicates with the most used email softwares (Outlook, Netscape Mail, Eudora, etc.).

10.10.4 SET

Secure Electronic Transaction (SET) was a communications protocol standard 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 enabled users to employ the existing credit card payment infrastructure on an open network in a secure fashion. However, it failed to gain attraction in the market. VISA now promotes the 3-D Secure scheme.ne iteration

chapter 11 the public key infrastructure

11.1 what is it?

A public key infrastructure (PKI) is a set of **roles**, **policies**, and **procedures** needed to **create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption**. The purpose of a PKI is to facilitate the secure electronic transfer of information for a range of network activities such as e-commerce, internet banking and confidential email. It is required for activities where simple passwords are an inadequate authentication method and more rigorous proof is required to confirm the identity of the parties involved in the communication and to validate the information being transferred.

In cryptography, a PKI is an arrangement that binds public keys with respective identities of entities (like persons and organizations). The binding is established through a process of registration and issuance of certificates at and by a certificate authority (CA). Depending on the assurance level of the binding, this may be carried out by an automated process or under human supervision.

The PKI role that assures valid and correct registration is called registration authority (RA). An RA is responsible for accepting requests for digital certificates and authenticating the entity making the request. In a Microsoft PKI, a registration authority is usually called a subordinate CA.

11.2 the infrastructure components

A public key **infrastructure** (PKI) is a system for the creation, storage, and distribution of digital certificates which are used to verify that a particular public key belongs to a certain entity. The PKI creates digital certificates which map public keys to entities, securely stores these certificates in a central repository and revokes them if needed.

A PKI consists of:

- A certificate authority (CA) that stores, issues and signs the digital certificates
- A registration authority which verifies the identity of entities requesting their digital certificates to be stored at the CA
- A central directory—i.e., a secure location in which to store and index keys
- A certificate management system managing things like the access to stored certificates or the delivery of the certificates to be issued.
- A certificate policy

11.3 PKI usage

PKIs of one type or another, and from any of several vendors, have many uses, including providing public keys and bindings to user identities which are used for:

- Encryption and/or sender authentication of e-mail messages (e.g., using OpenPGP or S/MIME)
- Encryption and/or authentication of documents (e.g., the XML Signature or XML Encryption standards if documents are encoded as XML)
- Authentication of users to applications (e.g., smart card logon, client authentication with SSL). There's experimental usage for digitally signed HTTP authentication in the Enigform and mod_openpgp projects
- Bootstrapping secure communication protocols, such as Internet key exchange (IKE) and SSL. In both of these, initial set-up of a secure channel (a "security association") uses asymmetric key—i.e.,

public key—methods, whereas actual communication uses faster symmetric key—i.e., secret key—methods.

Mobile signatures are electronic signatures that are created using a mobile device and rely on signature or certification services in a location independent telecommunication environment

11.4 public and private keys

A public key is an element that allows an entity to be identified in the public electronic space.

- Protocol developed by Netscape for the transmission of private documents via the Internet.
- Use a cryptographic system that uses two keys to encrypt data: a public key known to everyone and a private (secret) key known only to the key owner

11.5 digital certificates

One The structure of a digital certificate is specified in the X.509 standard. In version 3 of this standard, the structure is as follows:

- version - the version of the X.509 standard used to create the certificate
- series - identifier generated by the authority that issued the certificate
- identifier for the certificate algorithm - the combination of hash function and the algorithm used by the issuer to sign the certificate
- name of the issuer - the name of the certification authority (CA) that issued the certificate
- validity period - the start and expiration date of the certificate
- the name of the certificate owner
- information about the public key of the certificate owner
- the unique identifier of the issuer
- extensions (optional)
- the digital signature created by the issuer (CA), using the algorithm specified above

Below is an example - the digital certificate of uvt.ro

- version – `V3`
- series – `03 2e f6 9c a0 65 75 89 d9 59 5b 9e da 22 07 c1 c8 d6`
- certificate algorithm ientifier – `sha256RSA`
- issuer’s name – `CN = Let's Encrypt Authority X3, O = Let's Encrypt, C = US`
- validity period – start: `Saturday, September 28, 2019 6:53:27 AM,` end: `Friday, December 27, 2019 6:53:27 AM`
- the name of certificate’s owner – `uvt.ro`
- the public key – `30 82 01 0a 02 82 01 01 00 db 7a 0d 3e 33 b0 64 0d aa 5c 93 27 bb c9 09 65 7b de 53 f2 c8 4f 53 be b2 c4 c1 ee 2a 85 94 aa 78 c4 03 fe ad 3c 71 1d 14 78 5e c7 62 7b d5 c4 1b d8 93 53 b1 86 c4 3c 3c 5d 1f 34 af 38 bd e0 af d6 dc 60 bf 31 35 27 f9 49 09 4f 89 bc 3f bd 3a a4 85 79 62 0d 3d`

```
5e 98 76 28 19 6a 03 1b 36 0e 8b c8 2f 39 37 52 20 89 95 99 56 22 8e 6c dd
6a 4a df 1b d2 4a 00 53 ee 79 0a 87 f3 ce db a6 e7 a3 8f 6e 44 65 46 cf 67
db 48 3b 69 12 b2 fb d9 01 97 d6 d0 f6 74 af ba 67 78 cb 85 a1 28 cc f5 7c
62 5d fa 56 93 d0 4f 53 80 0d dd 2c 68 32 11 e0 cc 7b 24 b6 cd 2f 42 5a 79
ea 58 cc 30 36 68 8b 61 db 34 fa 18 51 b1 b6 45 a2 70 a1 2b 86 8e df 5b fe
83 48 33 97 ff 97 f6 df c1 04 25 f5 1a 7f 58 d6 5a 5f c7 c9 53 28 6b ed 7d
11 f1 46 41 d5 42 12 ab 39 5f 10 49 e4 8e 0a 0c 08 87 c3 02 03 01 00 01
```

- issuer’s unique identifier – `64 3c 2a 7f 0a 5d 97 c8 7e 5e 4d 94 eb af 5f f6 e0 82 d5 8a`
- extensions (optional)
- the digital signature of the issuer – `cd cc de 94 da 53 f4 02 bc 33 3a a4 f0 10 b1 7f b7 82 ac cc`

11.6 cerification authorities (CAs)

11.6.1 What is a certification authority?

A certification authority (CA) is an entity (a trusted service provider, or a certification service provider) that issues public key certificates, better known as electronic or digital certificates. The CA is also a crucial part of the public key infrastructure (PKI) because it uses a PKI to issue or revoke public key certificates and to provide verifiable statements about their status.

A digital certificate is a digital document that can contain information related to the identity of its owner, the issuer and the certificate itself. Having real identity attributes in a digital certificate is optional; Alternatively, it may comprise only a pseudonym.

The certificate allows the user to authenticate and access various sensitive online services (such as accessing web banking applications, public service platforms, etc.). Digital certificates also serve companies and service providers who guarantee that the person accessing their services is the right one. [THAT]

11.6.2 What does a certification authority do?

A certification authority is responsible for the entire life cycle of a digital certificate, namely:

- Activating a certificate. A certificate can be activated upon release. In general, before issuing a qualified certificate, the user's identity must be verified to ensure that the person is the person claiming to be.
- Reactivate a certificate (if suspended)
- Re-typing. The Board of Directors can also deal with the re-introduction of the certificate. Certificates have a limited validity period, depending on their types and PKI policies. For example, LuxTrust certificates are valid for a period of 3 years.
- Revocation of a certificate: The CA is also authorized to revoke a certificate that makes it irreversibly unusable. This can happen if there are suspicions that the certificate has been compromised or at the request of the authorities. A certificate can also be revoked if the CA finds that information in the certificate becomes outdated or inaccurate.

11.6.3 how to become a certification authority?

If a company wants to become a certification authority, then it needs to procure a certified PKI, define the appropriate processes and create an appropriate certificate policy (CP) and statement of practice (CPS). It must then be successfully recognized and certified by the competent authorities. When you are a certification authority, compliance with regulations is essential. In fact, the competent authorities are regularly audited and supervised by the authorities to ensure that they comply with current legislation. In the event of an offense, the competent authorities are subject to serious consequences which may lead to the suspension of their services.

11.7 open source implementations

- OpenSSL is the simplest form of CA and tool for PKI. It is a toolkit, developed in C, that is included in all major Linux distributions, and can be used both to build your own (simple) CA and to PKI enable applications. (Apache licensed)
- EJBCA is a full featured, Enterprise grade, CA implementation developed in Java. It can be used to set up a CA both for internal use and as a service. (LGPL licensed)
- OpenCA is a full featured CA implementation using a number of different tools. OpenCA uses OpenSSL for the underlying PKI operations.
- XCA is a graphical interface, and database. XCA uses OpenSSL for the underlying PKI operations.
- (Discontinued) TinyCA was a graphical interface for OpenSSL.
- XiPKI, CA and OCSP responder. OSGi-based, written in Java.

11.8 crtificate types

Not all certification authorities offer the same type of certificate. The difference comes from the set of verification measures performed. Thus, we have several types of certificates:

- Domain validation (DV) - only the domain is verified for authentication
- Wildcard - the basic domain and all its subdomains are included in a single certificate
- Extended validation (EV)
- Unified communications (UC) - for encrypting emails or other types of communications. May include multiple domains.
- Subject Alternative Name (SAN) - the root domain and its associated domains can be included in a single certificate
- Organization validation (OV)

11.9 registration authorities (RAs)

A registration authority can be defined as a trusted secondary entity. This can be a bank, an educational institution, a company, etc. [RA]

The role of the registration authorities is to clearly identify the future owner of the certificate and to manage the data involving this identification.

What does a Registration Authority do? We list the main attributes:

- identifies the certificate applicant through a procedure involving the verification of an official identity document
- ensures the concordance of personal data with those that will appear in the certificate
- administers the life cycle of the certificate in case of a request for revocation or renewal

11.10 the PKI in Romania

In Romania, the Ministry of Communications and Information Society (MCSI) exercises the attributions of supervisory body for qualified trust service providers established in Romania, as well as taking measures, as appropriate, in connection with unqualified trust service providers established in Romania. Romania, in accordance with Regulation (EU) no. 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93 / EC.

The status of qualified trust service provider is granted by the Supervisory Body, according to Regulation (EU) no. 910/2014 and OMCSI no. 449/2017.

Electronic signatures issued by entities on the Trusted List published at EU level are valid in all EU Member States.

In Romania, the certification authorities (accredited by the government) are:

- Trans Sped SRL
- Digisign SA
- Certsign SA
- Alfatrust Certification SA
- Centrul de Calcul SA
- SRI – Institutul pentru tehnologii avansate (?)

For an up-to-date list at EU level, use the link:- https://webgate.ec.europa.eu/tl-browser/

11.11 examples from the outside world

At EU level, the list of trusted certification authorities is the Trusted List, the link being the one in the previous paragraph. At country level, the number of trusted certification authorities varies between 1 (Cyprus and Denmark, for example) and 34 (Spain). Spain is rather the exception, the next countries with a large number of trusted certification authorities being France and Italy (22 each), but the average is 5-6.

In the United States, the network of certification authorities consists of root, intermediate, and issuing authorities. [USPKI]

Their work is governed by a set of rules, known as the Federal Common Policy Certification Authority (FCPCA), which serves as the root and trust anchor for intermediary and registration authorities operated by the Executive Branch Agencies of the Executive Branch. the federal government.

At the same time, the certifying authorities are divided (horizontally) into several categories, namely:

- PKI Shared Service Provider (SSP) - an entity subordinate to the FCPCA
- Private certification authorities - they can also provide services to the federal government
- Other government certification authorities - administered by state, local, tribal, territorial or international administrations
- Link CAs
- Older Federal Agencies (Legacy) - agencies that invested and conducted their own PKIs and Cases before 2004

Ensuring policy coherence in the area of certification authorities is achieved through an authority called the Federal Bridge Certification Authority (FBCA).

In China, the agency responsible for .cn domains (CNNIC - China Internet Network Information Center) provides services as a Certification Authority, although the market share of this authority is below 0.1%. (according to [SSLMS])

11.12 main CAs and SSL certificates providers

Top five (conform [TOP5])

- Let's Encrypt – if it's free, with pleasure. It offers 2048-bit RSA encryption and has the support of companies such as Automatic, Mozilla, Juices, Facebook, Chrome. Support for ECDSA being implemented. Free certificates, instant, free renewal, high compatibility. DV, SAN, UC type certificates.
- Comodo
- Symantec
- Digicert
- Geotrust

In terms of market share for SSL certificates used by websites, things look a little different, according to [SSL-MS]:

- IdenTrust – 52.2%
- DigiCert Group – 19.3%
- Sectigo – 17.3%
- GoDaddy Group – 6.8%
- GlobalSign – 2.9%

chapter 12 the digital signature standard

12.1 what is a digital signature?

Digital signatures are essential in today's world to verify who the sender of a document is. A digital signature is represented in a computer as a string of binary digits. The signature is computer using a set of rules and parameters (algorithm) such that the identity of the person signing the document as well as the originality of the data can be verified. The signature is generated by the use of a private key. A private key is known only to the user. The signature is verified makes use of a public key which corresponds to the private key. With every user having a public/private key pair, this is an example of public-key cryptography. Public keys, which are known by everyone, can be used to verify the signature of a user. The private key, which is never shared, is used in signature generation, which can only be done by the user.

Digital signatures are used to detect unauthorized modifications to data. Also, the recipient of a digitally signed document in proving to a third party that the document was indeed signed by the person who it is claimed to be signed by. This is known as nonrepudiation, because the person who signed the document cannot repudiate the signature at a later time. Digital signature algorithms can be used in e-mails, electronic funds transfer, electronic data interchange, software distribution, data storage, and just about any application that would need to assure the integrity and originality of data.

12.2 what is DSS?

Digital Signature Standard (DSS) is the digital signature algorithm (DSA) developed by the U.S. National Security Agency (NSA) to generate a digital signature for the authentication of electronic documents. DSS was put forth by the National Institute of Standards and Technology (NIST) in 1994, and has become the United States government standard for authentication of electronic documents. DSS is specified in Federal Information Processing Standard (FIPS) 186.

This Standard specifies algorithms for applications requiring a digital signature, rather than a written signature. A digital signature is represented in a computer as a string of bits. A digital signature is computed using a set of rules and a set of parameters that allow the identity of the signatory and the integrity of the data to be verified. Digital signatures may be generated on both stored and transmitted data.

Signature generation uses a private key to generate a digital signature; signature verification uses a public key that corresponds to, but is not the same as, the private key. Each signatory possesses a private and public key pair. Public keys may be known by the public; private keys are kept secret. Anyone can verify the signature by employing the signatory's public key. Only the user that possesses the private key can perform signature generation. A hash function is used in the signature generation process to obtain a condensed version of the data to be signed; the condensed version of the data is often called a message digest. The message digest is input to the digital signature algorithm to generate the digital signature.

The hash functions to be used are specified in the Secure Hash Standard (SHS), FIPS 180. FIPS approved digital signature algorithms shall be used with an appropriate hash function that is specified in the SHS. The digital signature is provided to the intended verifier along with the signed data. The verifying entity verifies the signature by using the claimed signatory's public key and the same hash function that was used to generate the signature. Similar procedures may be used to generate and verify signatures for both stored and transmitted data.

In the picture below (provided by tutorialspoint.com) is shown the scheme for creating and verifying a digital signature.

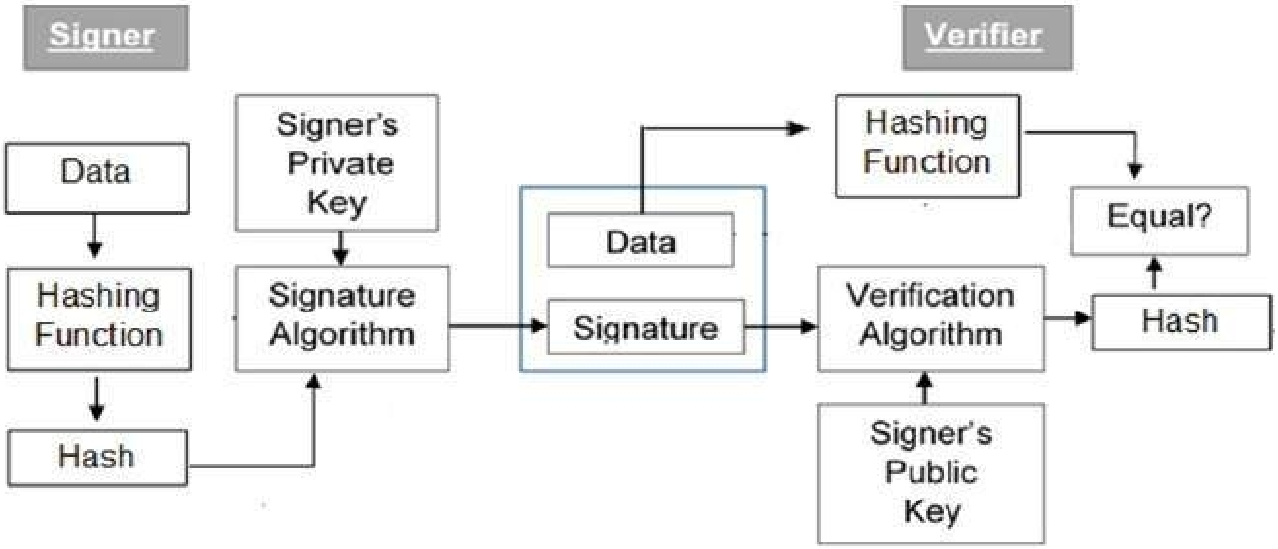


Fig. 12.1 Digital signature creation and verification scheme

12.3 the digital signature algorithm - DSA

DSA refers to a standard for digital signatures. It was introduced in 1991 by the National Institute of Standards and Technology (NIST) as a better method of creating digital signatures. Along with RSA, DSA is considered one of the most preferred digital signature algorithms used today.

DSA is one the main three signature algorithms. the other two are :

- RSA
- ECDSA – Elliptic Curve Digital Signature Algorithm

Unlike DSA, most digital signature types are generated by signing message digests with the private key of the originator. This creates a digital thumbprint of the data. Since just the message digest is signed, the signature is generally much smaller compared to the data that was signed. As a result, digital signatures impose less load on processors at the time of signing execution, use small volumes of bandwidth, and generate small volumes of cipher text intended for cryptanalysis.

DSA, on the other hand, does not encrypt message digests using private key or decrypt message digests using public key. Instead, it uses unique mathematical functions to create a digital signature consisting of two 160-bit numbers, which are originated from the message digests and the private key. DSAs make use of the public key for authenticating the signature, but the authentication process is more complicated when compared with RSA.

The digital signature procedures for RSA and DSA are usually regarded as being equal in strength. Because DSAs are exclusively used for digital signatures and make no provisions for encrypting data, it is typically not subject to import or export restrictions, which are often enforced on RSA cryptography.

12.3.1 DSA procedures

The algorithm consists of three procedures:

- key generation and distribution
- signing
- signature verification.

A DSA digital signature is calculated using a set of domain parameters, a private key x, a secret number k specific to each message, the data to be signed, and a hash function.

The digital signature is verified using the same domain parameters, the public key y associated with the private key x, the data to be verified as well as the hash function used in the generation.

The digital signature (in the case of the DSA algorithm) is generated as follows:

1. a hash function (such as SHA1, SHA256, MD5) is applied to the data / message / file to be signed.

2. a random number K is generated

3. the private key of the signatory is used

The result of applying the digital signature algorithm is a pair of integers r + s of length (in general) of 40-64 bytes.

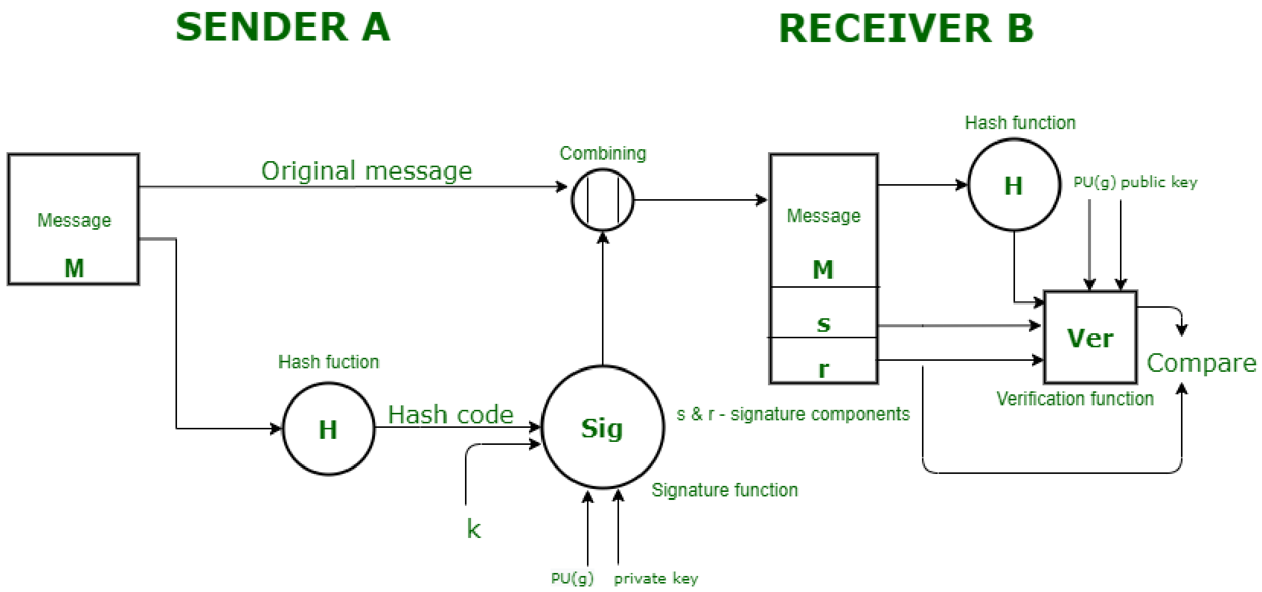
Verification of the digital signature is done by applying the same algorithm (verification part):

1. the same hash function applies to the received data / message / file

2. the received digital signature (r + s) is used

3. the public key of the signatory is used

In the picture below one can see the steps involved in the signature generation and verification processes in the DSA case.



12.3.2 DSA Parameters

- **p** = a prime modulus, where $2^{L-1} < p < 2^L$ for $512 \le L \le 1024$ and L is a multiple of 64. So L will be one member of the set {512, 576, 640, 704, 768, 832, 896, 960, 1024}.

- **q** = a prime divisor of p-1, where $2^{159} < q < 2^{160}$

- **g** = $h^{(p-1)/q}$ mod p, where h is any integer with 1 < h < p -1 such that $h^{(p-1)/q}$ mod p > 1 (g has order q mod p)

- **x** = a randomly integer with 0 < x < q

- **y** = g^x mod p

- **k** = a randomly generated integer with 0 < k < q

12.3.3 generation of primes p and q

Now we have to know how to generate p and q. It will be describe step by step:

The prime generation scheme starts by using the SHA and a user supplied SEED to construct a prime, q, in the range $2^{159} < q < 2^{160}$. Once this is accomplished, the same SEED value is used to construct an X in the range $2^{L-1} < X < 2^L$. The prime, p, is then formed by rounding X to a number congruent to 1 mod 2q as described below. An integer x in the range 0 ≤ x < 2 g may be converted to a g-long sequence of bits by using its binary expansion as shown below:

$x = x_1 * 2^{g-1} + x_2 * 2^{g-2} + ... + x_{g-1} * 2 + x_g \rightarrow \{ x_1,..., x_g \}$.

Conversely, a g-long sequence of bits $\{ x_1,..., x_g \}$ is converted to an integer by the rule:

$\{ x_1,..., x_g \} \rightarrow x_1 * 2^{g-1} + x_2 * 2^{g-2} + ... + x_{g-1} * 2 + x_g$.

Note that the first bit of a sequence corresponds to the most significant bit of the corresponding integer and the last bit to the least significant bit.

12.3.4 steps

As specified in - http://www.ijettcs.org/Volume4Issue2/IJETTCS-2014-12-22-136.pdf

Let L -1 = n* 160 + b, where both b and n are integers and 0 ≤ b < 160.

Step1. Choose an arbitrary sequence of at least 160 bits and call it SEED. Let g be the length of SEED in bits.

Step2. Compute U = SHA-1[SEED] XOR SHA-1[(SEED+ 1) mod 2 g].

Step3. Form q from U by setting the most significant bit (the 2 159 bit) and the least significant bit to 1. In terms of Boolean operations, q = U OR 2159 OR 1. Note that 2159 < q < 2160 .

Step4. Use a robust primality testing algorithm to test whether q is prime 1 .

Step5. If q is not prime, go to step 1.

Step6. Let counter = 0 and offset = 2. Step7.

For k = 0,..., n let Vk = SHA-1[(SEED + offset + k) mod 2g]. 1 A robust primality test is one where the probability of a non-prime number passing the test is at most 2-80

Step8. Let W be the integer W = V0 + V1* 2160 + ... + Vn-1* 2(n-1)* 160 + (Vn mod 2b) * 2 n* 160 and let X = W + 2L-1 . Note that 0 ≤ W < 2L-1 and hence 2L-1 ≤ X < 2L .

Step9. Let c = X mod 2q and set p = X -(c -1). Note that p is congruent to 1 mod 2q.

Step10. If p < 2L-1 , then go to step 13.

Step11. Perform a robust primality test on p.

Step12. If p passes the test performed in step 11, go to step 15.

Step13. Let counter = counter + 1 and offset = offset + n + 1.

Step14. If counter ≥ 212 = 4096 go to step 1, otherwise (i. e. if counter < 4096) go to step 7.

Step15. Save the value of SEED and the value of counter for use in certifying the proper generation of p and q.

- g = h(p-1)/ q mod p, where h is any integer with 1 < h < p -1 such that h(p-1)/ q mod p > 1 (g has order q mod p)
- x = a randomly or pseudorandomly generated integer with 0 < x < q

- y = gx mod p
- k = a randomly or pseudorandomly generated integer with 0 < k < q

The parameters p, q, and g are made public. The users will have the private key, x, and the public key y. The parameters x and k are used for signature generation and must be kept private and k will be randomly or pseudorandomly generated for each signature. This part seems to be straightforward so far. The signature of the message M will be a pair of the numbers r and s which will be computed from the following equations.

- r = (gk mod p) mod q
- s = (k-1 (SHA(M) + xr)) mod q

k -1 is the multiplicative inverse of k (mod q). The value of SHA(M) is a 160-bit string which is converted into an integer according to the SHS standard. Then the signature is sent to the verifier.

12.3.5 verification

Before getting the digitally signed message the receiver must know the parameters p, q, g, and the sender's public key y.

We will let M′, r′, s′ be the received versions of M, r, and s. To verify the signature the verifying program must check to see that 0 < r′ < q and 0 < s′ < q and if either fails the signature should be rejected. If both of the conditions are satisfied then we will compute

12.4 sensitivity

With DSA, the entropy, secrecy, and uniqueness of the random signature value k is critical. It is so critical that violating any one of those three requirements can reveal the entire private key to an attacker. Using the same value twice (even while keeping k secret), using a predictable value, or leaking even a few bits of k in each of several signatures, is enough to break DSA.

In December 2010, a group calling itself *fail0verflow* announced recovery of the ECDSA private key used by Sony to sign software for the PlayStation 3 game console. The attack was made possible because Sony failed to generate a new random k for each signature.

This issue can be prevented by deriving k deterministically from the private key and the message hash, as described by RFC 6979. This ensures that k is different for each H(m) and unpredictable for attackers who do not know the private key x.

chapter 13 authentication techniques

The Network/computer security hinges on two very simple goals:

- Keeping unauthorized persons from gaining access to resources
- Ensuring that authorized persons can access the resources they need

Access permissions works only if you are able to verify the identity of the user who is attempting to access the resources. That's where authentication comes in.

Authentication is the process of confirming the identification of a user (or in some cases, a machine) that is trying to log on or access resources.

It is easy to confuse authentication with another element of the security plan: authorization. While authentication verifies the user's identity, authorization verifies that the user in question has the correct permissions and rights to access the requested resource. As you can see, the two work together. Authentication occurs first, then authorization.

13.1 authentication occurrences

- Logon authentication
  - It is required by most network operating systems.
  - Users need to authenticate in order to log onto the network.
- Network access authentication
  - Verifies the user's identity to each network service that the user attempts to access.
  - Authentication process is, in most cases, transparent to the user once he or she has logged on. Otherwise, the user would have to reenter the password or provide other credentials every time he or she wanted to access another network service or resource.
- IPSec authentication
  - Provides a means for users to encrypt and/or sign messages that are sent across the network to guarantee confidentiality, integrity, and authenticity.
  - An important consideration is that both the sending and receiving computers must be configured to use a common authentication method or they will not be able to engage in secured communications.
- Remote authentication
  - Remote users can be authenticated via a Remote Authentication Dial-In User Service (RADIUS) or the Internet Authentication Service (IAS).
- Single Sign-On (SSO)
  - Single Sign-On (SSO) is a feature that allows a user to use one password (or smart card) to authenticate to multiple servers on a network without reentering credentials.

Users don't have to remember multiple passwords or keep going through the authentication process over and over to access different resources.

13.2 means to provide authentication credentials

- **Password authentication**
    - To log onto a computer or network, you enter a user account name and the password assigned to that account. This password is checked against a database that contains all authorized users and their passwords.
    - To preserve the security of the network, passwords must be “strong,” that is, they should contain a combination of alpha and numeric characters and symbols, they should not be words that are found in a dictionary, and they should be relatively long (eight characters or more). In short, they should not be easily guessed.
    - Password authentication is vulnerable to a password “cracker” who uses a brute force attack (trying every possible combination until hitting upon the right one) or who uses a protocol “sniffer” to capture packets if passwords are not encrypted when they are sent over the network.
- **Smart card authentication**
    - Smart cards are credit card-sized devices that hold a small computer chip, which is used to store public and private keys and other personal information used to identify a person and authenticate him or her to the system.
    - Logging onto the network with a smart card requires that you physically insert the card into (or slide it through) a reader and then enter a Personal Identification Number (PIN) in much the same way that you use an ATM card to access an automatic teller machine.
    - Smart cards use cryptography-based authentication and provide stronger security than a password because in order to gain access, the user must be in physical possession of the card and must know the PIN.
- **Biometric authentication**
    - An even more secure type of authentication than smart cards

Biometric authentication involves the use of biological statistics that show that the probability of two people having identical biological characteristics such as fingerprints is infinitesimally small; thus, these biological traits can be used to positively identify a person.

13.3 protocols used for authentication

An authentication protocol is a type of cryptographic protocol with the purpose of authenticating entities wishing to communicate securely.

There are different authentication protocols such as:

- Kerberos
- Microsoft CHAP
- Secure Socket Layer (SSL)
- Host Identity Protocol (HIP)
- Microsoft Network LAN Manager (NTLM)
- Password Authentication Protocol (PAP)
- Extensible Authentication Protocol (EAP)
- Shiva Password Authentication Protocol (SPAP)

- Protected Extensible Authentication Protocol (PEAP)
- Remote Authentication Dial-In User Service (RADIUS)
- Challenge-Handshake Authentication Protocol (CHAP)

chapter 14 kerberos

14.1 overview

Kerberos is a secure method for authenticating a request for a service in a computer network. It was developed at MIT and the name is taken from Greek mythology: Kerberos was a three-headed dog who guarded the gates of Hades. The three heads of Kerberos are:

1. Key Distribution Center (KDC)
2. Client user
3. Server with the desired service to access

The KDC is installed as part of the domain controller and performs two service functions: the Authentication Service (AS) and the Ticket-Granting Service (TGS). As exemplified in Figure 1, three exchanges are involved when the client initially accesses a server resource:

1. AS Exchange
2. TGS Exchange
3. Client/Server (CS) Exchange

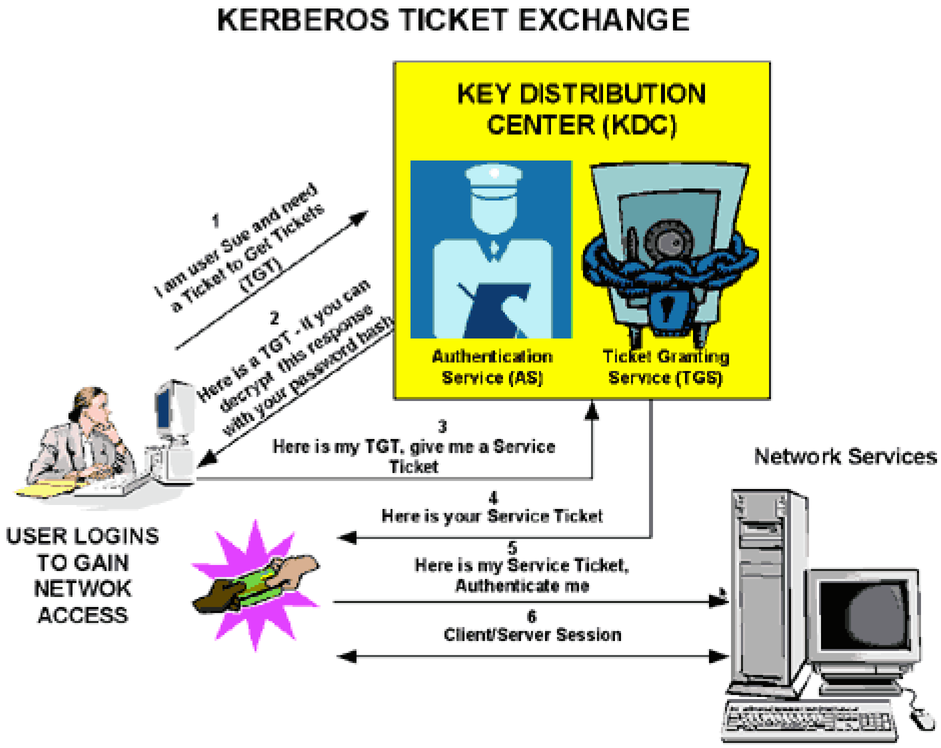

KERBEROS TICKET EXCHANGE
KEY DISTRIBUTION CENTER (KDC)
Authentication Service (AS)
Ticket Granting Service (TGS)
1 I am user Sue and need a Ticket to Get Tickets (TGT)
2 Here is a TGT - if you can decrypt this response with your password hash
3 Here is my TGT, give me a Service Ticket
4 Here is your Service Ticket
5 Here is my Service Ticket, Authenticate me
6 Client/Server Session
USER LOGINS TO GAIN NETWOK ACCESS
Network Services

Figure 14.1

A Kerberos realm defines what Kerberos manages in terms of who can access what. This realm holds the client, the service or host wanted to request and the KDC. It is created by the admin and encompasses all that is available to access.

14.2 authentication service (AS) exchange

When initially logging on to a network, users must negotiate access by providing a **log-in name** and **password** in order to be verified by the AS portion of a KDC within their domain. The KDC has access to Active Directory user account information.

Once successfully authenticated, the user is granted a **Ticket to Get Tickets (TGT)** that is valid for the local domain. The TGT has a default lifetime of 10 hours and may be renewed throughout the user's log-on session without requiring the user to re-enter his password. The TGT is cached on the local machine in volatile memory space and used to request sessions with services throughout the network.

If the KDC approves the client's request for a TGT, the reply (referred to as the AS reply) will include two sections:

- a TGT encrypted with a key that only the KDC (TGS) can decrypt
- a session key encrypted with the user's password hash to handle future communications with the KDC.

Because the client system cannot read the TGT contents, it must blindly present the ticket to the TGS for service tickets. The TGT includes:

- time to live parameters
- authorization data
- a session key to use when communicating with the client
- the client's name.

14.3 TGS exchange

The user presents the TGT to the TGS portion of the KDC when desiring access to a server service. The TGS on the KDC authenticates the user's TGT and creates a ticket and session key for both the client and the remote server. This information, known as the service ticket, is then cached locally on the client machine.

The TGS receives the client's TGT and reads it using its own key. If the TGS approves of the client's request, a service ticket is generated for both the client and the target server. The client reads its portion using the TGS session key retrieved earlier from the AS reply. The client presents the server portion of the TGS reply to the target server in the client/server exchange coming next.

14.4 client/server exchange

Once the client user has the client/server service ticket, he can establish the session with the server service. The server can decrypt the information coming indirectly from the TGS using its own long-term key with the KDC. The service ticket is then used to authenticate the client user and establish a service session between the server and client. After the ticket's lifetime is exceeded, the service ticket must be renewed to use the service.

The client blindly passes the server portion of the service ticket to the server in the client/server request to establish a client/server session. If mutual authentication is enabled, the target server returns a time stamp encrypted using the service ticket session key. If the time stamp decrypts correctly, not only has the client authenticated himself to the server, but the server also has authenticated itself to the client.

14.5 remote access - referral tickets

A TGT and a service ticket are required to successfully log on to a local system but are also needed to access services on remote computers. The AS and TGS functions are separate within the KDC. This permits the user to use the TGT obtained from an AS in his domain to obtain service tickets from a TGS in other domains.

This is accomplished through referral tickets. Once a trust has been established between two domains, referral tickets can be granted to clients requesting authorization for services in other domains. When there is a trust established between the two domains, an inter-domain key based on the trust password becomes available for authenticating KDC functions. This can best be explained by example of a user/client seeking services in another domain.

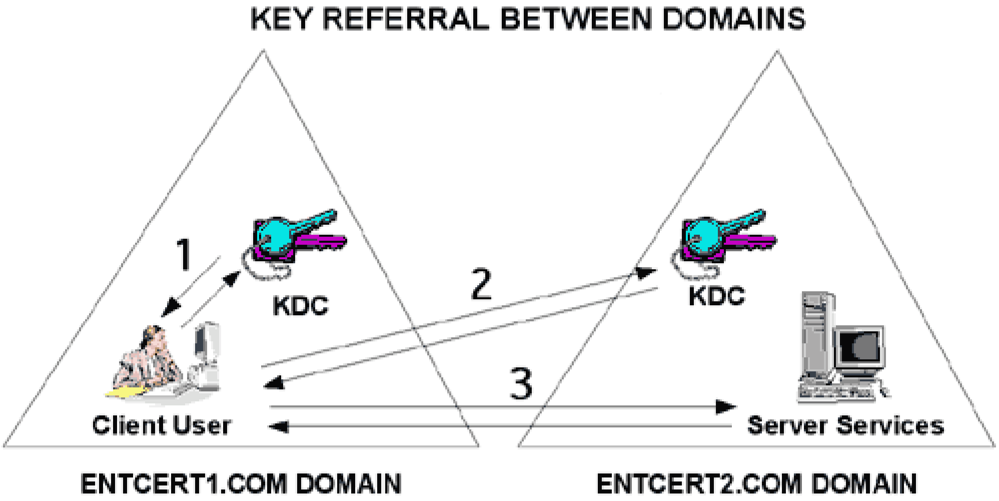


Figure 14.2

As illustrated in Figure 14.2, a user client in Entcert1.com requests authority for a server in Entcert2.com. He utilizes referral tickets. The numbers in Figure 2 correspond to the following numbered explanations:

1. The client contacts its domain KDC TGS using a TGT. The KDC recognizes a request for a session with a foreign domain server and responds by returning a referral ticket for the KDC in the foreign domain.
2. The client contacts the KDC of the foreign domain with the referral ticket. This ticket is encrypted with the inter-domain key. Given that the decryption works, the TGS service for the foreign domain returns a service ticket for the server service in Entcert2.com.
3. The client performs the client/server exchange with the server and begins the user session with the service.

14.6 kerberos - history

Kerberos was developed by MIT to protect network services provided by Project Athena (a joint project of MIT, DEC and IBM to produce a campus-wide distributed computing environment for educational purposes, launched in 1983 and still in use in 2020).

The protocol is based on the earlier Needham-Scroeder symmetric key protocol.

Versions 1-3 were used only internally by MIT.

Version 4 was published in late 1980s and targeted Project Athena.

Version 5 was formalized as RFC 1510 in 1993 and it was superseded by RFC 4120 in 2005.

In 2005, the IETF Kerberos work group updated the specifications in different areas:

- Encryption and Checksum Specifications (RFC 3961).
- Advanced Encryption Standard (AES) Encryption for Kerberos 5 (RFC 3962).
- A new edition of the Kerberos V5 specification "The Kerberos Network Authentication Service (V5)" (RFC 4120). This version obsoletes RFC 1510, clarifies aspects of the protocol and intended use in a more detailed and clearer explanation.
- A new edition of the Generic Security Services Application Program Interface (GSS-API) specification "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2." (RFC 4121).

MIT makes an implementation of Kerberos freely available, under copyright permissions similar to those used for BSD.

14.7 kerberos - usage

Windows 2000 and later versions use Kerberos as their default authentication method. Some Microsoft additions to the Kerberos suite of protocols are documented in RFC 3244 "Microsoft Windows 2000 Kerberos Change Password and Set Password Protocols". RFC 4757 documents Microsoft's use of the RC4 cipher. While Microsoft uses and extends the Kerberos protocol, it does not use the MIT software.

Many Unix-like operating systems, including FreeBSD, OpenBSD, Apple's macOS, Red Hat Enterprise Linux, Oracle's Solaris, IBM's AIX, HP-UX and others, include software for Kerberos authentication of users or services. A variety of non-Unix like operating systems such as z/OS, IBM i and OpenVMS also feature Kerberos support. Embedded implementation of the Kerberos V authentication protocol for client agents and network services running on embedded platforms is also available from companies

chapter 15 IPsec and Ipv6 security features

The **S**.

15.1 paragraph 1

One

15.2 par 2

One iteration

chapter 16 secure communication – vpn, tls, ssh

Secure communication is when two entities are communicating and do not want a third party to listen in. For that they need to communicate in a way not susceptible to eavesdropping or interception. Internet communications that are based on the Transfer Control Protocol/Internet Protocol (TCP/IP), such as the Hypertext Transfer Protocol (HTTP), Telnet, and File Transfer Protocol (FTP), are not secure because all communication occurs in plaintext. Confidential or sensitive information that is transmitted with these protocols can easily be intercepted and read unless the information is protected by encryption technology.

Considering the recent events, related to NSA surveillance and the deliberate weakening of cryptographic systems, secure communication becomes that much more important.

The primary way the NSA eavesdrops on internet communications is in the network. They have invested in enormous programs to automatically collect and analyze network traffic. They gain access to data through different “secret partnerships” with UK/USA telecommunication providers and not only (tapping undersea cables, intercepting satellite communications and so on). The agency can quickly filter the data, looking for “interesting” traffic ("Interesting" can be defined in many ways: by the source, the destination, the content, the individuals involved, and so on. This data is funneled into the vast NSA system for future analysis).

Secure Web communication protocols provide a way to authenticate clients and servers on the Web and to protect the confidentiality of communication between clients and servers. A variety of secure communication standards that use public key technology have been developed, including Secure Hypertext Transfer Protocol (SHTTP), IP Security (IPSec), PPTP, and L2TP. The leading general-purpose, secure Web communication protocols are SSL 3.0 and the open TLS protocol that is based on SSL. The SSL and TLS protocols are widely used to provide secure channels for confidential TCP/IP communication on the Web.

16.1 VPN

A virtual private network (VPN) extends a private network across a public network, such as the Internet. It enables a computer or network-enabled device to send and receive data across shared or public networks as if it were directly connected to the private network, while benefiting from the functionality, security and management policies of the private network. A VPN is created by establishing a virtual point-to-point connection through the use of dedicated connections, virtual tunneling protocols, or traffic encryption. Major implementations of VPNs include OpenVPN and Ipsec.

Businesses use VPNs to connect remote datacenters, and individuals can use VPNs to get access to network resources when they're not physically on the same LAN (local area network), or as a method for securing and encrypting their communications when they're using an untrusted public network.

A VPN alone is just a way to increase your security and access resources on a network you're not physically connected to. What you choose to do with a VPN is a different story. The most common uses are:

- **Students/workers** need to access resources on their network while they’re at home or traveling.
- **To download files securely**. VPNs are the only way to stay safe when using something like BitTorrent or other Peer To Peer sharing service.
- **To surpass geographically restricted content or services** such as Netflix (video streaming service) or Pandora/Spotify (music streaming services)

Most VPNs rely on tunneling to create a private network that reaches across the Internet. Tunneling is the process of **placing an entire packet within another packet** before it's transported over the Internet. That outer packet protects the contents from public view and ensures that the packet moves within a virtual tunnel.

This layering of packets is called encapsulation. Computers or other network devices at both ends of the tunnel, called tunnel interfaces, can encapsulate outgoing packets and reopen incoming packets. Users at both ends have to configure the tunnel interfaces they’re responsible for to use a tunneling protocol (also known as encapsulation protocol). **A tunneling protocol is a standardized way to encapsulate packets**.

The purpose of the tunneling protocol is to add a layer of security that protects each packet on its journey over the Internet. We have to note that the packet is traveling with the same transport protocol it would have used without the tunnel. A more colorful way to look at the relationship between the protocols would be to think of tunneling as having a package sent to you by a shipping company. The vendor who is sending you the computer packs the computer (**passenger protocol**) in a box (**tunneling protocol**). Shippers then place that box on a shipping truck (**transport protocol**) at the vendor's warehouse (**one tunnel interface**). The truck (**transport protocol**) travels over the highways (**Internet**) to your home (**the other tunnel interface**) and delivers the computer. You open the box (**tunneling protocol**) and remove the computer (**passenger protocol**).

There is no standard that all VPNs follow in terms of their setup but some of the most common equipment used are:

- **Network access server** - a NAS is responsible for setting up and maintaining each tunnel in a remote-access VPN.
- **Firewall** - A firewall provides a strong barrier between your private network and the Internet. IT staff can set firewalls to restrict what type of traffic can pass through from the Internet onto a LAN, and on what TCP and UDP ports.
- **AAA Server** - The acronym stands for the server's three responsibilities: authentication, authorization and accounting. For each VPN connection, the AAA server confirms who you are (authentication), identifies what you're allowed to access over the connection (authorization) and tracks what you do while you're logged in (accounting). An example of AAA server is **RADIUS** (Remote Authentication Dial-in User Service). This protocol isn't just for dial-up users. When a RADIUS server is part of a VPN, it handles authentication for all connections coming through the VPN's NAS.

On a side note, over the passing of time and because of its popularity, companies have developed dedicated VPN devices that business can purchase.

A few examples**:** **VPN Concentrator** (replace AAA server installed on a generic server), **VPN-enabled/VPN-optimized Router**, **VPN-enabled Firewall** and **VPN Client**.

Encryption is the process of encoding data so that only a computer with the right decoder will be able to read and use it. An encryption key tells the computer what computations to perform on data in order to encrypt or decrypt it. The most common forms of encryption are symmetric-key encryption or public-key encryption: **Symmetric-key encryption** (all users have the same key that is used for both encryption and decryption) and **public-key encryption** (each computer has a public/private key. A computer uses the private key to encrypt while another uses the corresponding public key to decrypt the data).

In a VPN environment, both end of the tunnel encrypt data entering and decrypt it at the other end. But a VPN needs more than just keys to apply the encryption mechanisms. (Protocols come in) A site-to-site VPN could use either Internet protocol security protocol (IPSec) or generic routing encapsulation (GRE). GRE provides the framework for how to package the passenger protocol for transport over the Internet protocol (IP). This framework includes information on what type of packet you're encapsulating and the connection between sender and receiver.

IPSec is a widely used protocol for securing traffic on IP networks, including the Internet. IPSec can encrypt data between various devices, including router to router, firewall to router, desktop to router, and desktop to server. IPSec consists of two sub-protocols which provide the instructions a VPN needs to secure its packets:

- **Encapsulated Security Payload** (**ESP**) encrypts the packet's payload (the data it's transporting) with a symmetric key.
- **Authentication Header** (**AH**) uses a hashing operation on the packet header to help hide certain packet information (like the sender's identity) until it gets to its destination

In a **remote- access VPN**, tunneling typically relies on **Point-to-point Protocol** (PPP) which is part of the native protocols used by the Internet. More accurately, though, remote-access VPNs use one of three protocols based on PPP: **L2F** (Layer 2 Forwarding), **PPTP** (Point-to-point Tunneling Protocol) and **L2TP** (Layer 2 Tunneling Protocol)

16.2 TLS

One problem when you administer a network is securing data that is being sent between applications across an untrusted network. You can use TLS/SSL to authenticate servers and clients and then use it to encrypt messages between the authenticated parties.

History

Netscape Communications created the original specification of secure socket layer (SSL) in 1994, when it became apparent that there was no way to securely transfer reliable protocols across the internet, without fear of interference or snooping of traffic. The first specification, version 1.0 was so heavily criticised by the cryptographic community for the implementation of weak cryptographic algorithms that it was never realised for public use.

The Netscape Communications department revised the specification and released a much improved version 2.0 in February 1995, as described by (Shostack, 1995) the second version of the protocol requested the use of the MD5 hash function and required the use of MD5 for all cipher types, the MD5 algorithm is defined in (RFC 1321).

While SSL version 2.0 was considered a fairly strong and robust protocol, it did have some areas where it was vulnerable. So in 1996 the next iteration of the protocol version 3.0, which was designed by both Netscape and Paul Kocher, was released. As described by (Gibson, 2009) version 3 addressed the implementation of the weak MD5 hash that was implemented in version 2 by producing both an MD5 hash and a SHA-1 hash and XOR'ing the result together to create a hybrid hash that was dependant on both algorithms.

As the protocol was now gaining such traction on the internet, the Internet Engineering Task Force (IETF) took responsibility for the protocol and renamed it to transport Layer Security (TLS) to avoid bias towards any particular company.

Finally the latest and most current version of the TLS standard, 1.2 and was released in August 2008 and has a number of improvements as documented in (RFC 5246), including the removal of older cipher suites like DES and IDEA and the inclusion of the SHA256 cipher suites.

In the authentication process, a TLS/SSL client sends a message to a TLS/SSL server, and the server responds with the information that the server needs to authenticate itself. The client and server perform an additional exchange of session keys, and the authentication dialog ends. When authentication is completed, SSL-secured communication can begin between the server and the client using the symmetric encryption keys that are established during the authentication process.

For servers to authenticate to clients, TLS/SSL does not require server keys to be stored on domain controllers or in a database, clients confirm the validity of a server's credentials with a trusted root certification authority's (CA's) certificates.

TLS and SSL are most widely recognized as the protocols that provide secure HTTP (HTTPS) for Internet transactions between Web browsers and Web servers. TLS/SSL can also be used for other application level protocols, such as File Transfer Protocol (FTP), Lightweight Directory Access Protocol (LDAP), and Simple Mail Transfer Protocol (SMTP). TLS/SSL enables server authentication, client authentication, data encryption, and data integrity over networks such as the World Wide Web.

The four protocol layers of the SSL protocol (Record Layer, ChangeCipherSpec Protocol, Alert Protocol, and Handshake Protocol) encapsulate all communication between the client machine and the server.

Record Layer

The record layer formats the Alert, ChangeCipherSpec, Handshake and application protocol messages. This formatting provides a header for each message and a hash, generated from a Message Authentication Code (MAC) at the end. The fields that comprise the five-byte header of the Record Layer are: Protocol Definition (1 byte), Protocol Version (2 bytes) and the Length (2 bytes). The protocol messages that follow the header cannot be longer than 16,384 bytes, as specified by the SSL protocol.

ChangeCipherSpec Protocol

The ChangeCipherSpec layer is composed of one message that signals the beginning of secure communications between the client and server. Though the ChangeCipherSpec Protocol uses the Record Layer format, the actual ChangeCipherSpec message is only one byte long, and signals the change in communications protocol by having a value of ‘1’.

Alert Protocol

This protocol sends errors, problems or warnings about the connection between the two parties. This layer is formed with two fields: the Severity Level and Alert Description.

Severity Level

The Severity Level sends messages with a ‘1’ or ‘2’ value, depending on the level of concern. A message with a value of ‘1’ is a cautionary or warning message, suggesting that the parties discontinue their session and reconnect using a new handshake. A message with a value of ‘2’ is a fatal alert message, and requires that the parties discontinue their session.

Alert Description

The Alert Description field indicates the specific error that caused the Alert Message to be sent from a party. This field is one byte, mapped to one of twelve specific numbers, and can take on one of the following meanings. Those descriptions that always follow a “fatal” alert message are underlined.

16.3 the handshake protocol in TLS

Messages passed back and forth between the user’s browser (client) and web application (server) establish a handshake that begins a secure connection. The following steps are how a SSL handshake is performed. The messages that compose this handshake are: ClientHello, ServerHello, ServerKeyExchange, ServerHelloDone, ClientKeyExchange, ChangeCipherSpec, Finished, ChangeCipherSpec, Finished.

ClientHello

The first message is the ClientHello. Since the client machine is requesting the secure communication session, this message involves a set of options that the client is willing to use in order to communicate with the server. The option categories are: Version of SSL to be used, CipherSuites supported by the client, and CompressionMethods used by the client. Other information that is included in this message is a 32-byte RandomNumber that assists the client in establishing encrypted communications, and a SessionID field that is blank. This message is generated by the client in the web e-mail example when our user wants to check her email and clicks on the “secure connection” option that is made available on many websites.

ServerHello

The second message of the SSL handshake is the ServerHello. In this message, the server makes choices based on the ClientHello message. The server returns five fields, just like the ClientHello message, but fills in the SessionID, and makes firm decisions on the Version of SSL to be used, the CompressionMethod and CipherSuite. The date and time stamp replaces four bytes of the RandomNumber field to avoid repeated random values, and Thomas adds that “the remaining bytes should be created by a cryptographically secure random number generator.”

ServerKeyExchange

Now that the server has made decisions for the transmission of data, information must be passed between the parties to determine how data will be encrypted. Since no algorithm has been previously agreed upon, this information is sent with no encryption. This means that all communication for this segment must already be in the public domain. The server’s public key is used to encrypt a separate session key to be maintained for this secure communication. Both the client and server will use this same key to encrypt data to be transmitted. To ensure that the communicating parties are who they claim to be, digital certificates are used to provide electronic identification. Digital certificates combine the public key and connect it to the name of the certificate owner. Additionally, these certificates contain public keys to certification authorities like RSA Security or VeriSign and an expiration date so that the person receiving the digital certificate can verify the link between the certificate owner and the certification authority. The certificate only contains the public key, and should never include the private key, else the private key would be compromised, and the entire purpose of having the digital certificate would be voided.

ServerHelloDone

Once the Server has completed the ServerKeyExchange message, the client receives a ServerHelloDone message to indicate that the server is through with its messages. It is similar to a two-way radio conversation when the sending party says "OVER" to announce that he is done sending a message, and signals the receiving party to acknowledge the message that was sent.

ClientKeyExchange

Since SSL does not require a client to have public and private keys in order to establish a SSL session, the ClientKeyExchange message contains information about the key that the client and server will use to communicate. This is the point where the "man in the middle" attack is mitigated since a masquerader must know the server's private key in order to decrypt this message. This message completes the negotiation processes between the client and the server.

ChangeCipherSpec

The two ChangeCipherSpec messages signal the change of data transmission from an insecure state to a secure state. As each computer sends the ChangeCipherSpec message, it changes its side of the connection into the agreed-upon secure state.

Finished

The two messages signaling the final messages of the SSL handshake ensure that three things are verified before the initial handshake is complete. These are:

- Key Information
- Contents of all previous SSL handshake messages exchanged by the systems
- A special value indicating whether the sender is a client or server

At the end of this handshake process, the user will see a lock icon in the corner of the browser to indicate that a secure protocol has been agreed upon, and is in use by the browser and the web e-mail server.

Message Authentication

Once this information is checked, the communication can continue, appending a message authentication algorithm to the end of each message. Message Authentication is performed by using "an algorithm that uses cryptographic technology to create a digital summary of information so that if the information is altered, the summary (known as a hash) will also change." (Thomas, 186) MD5 and SHA are common hash functions used in SSL communications.

Resuming a Disconnected Session

If an Alert message disconnects a sessions before the parties are through communicating, that session can be resumed if the client sends a HelloRequest to the server with the properly encrypted SessionID information. The server then determines if the SessionID is valid, exchanges ChangeCipherSpec and Finished messages with the client machine, and secure communication can resume.

TLS/SSL provides numerous benefits to clients and servers over other methods of authentication, including:

- **Strong authentication, message privacy, and integrity -** TLS/SSL can help to secure transmitted data using encryption. TLS/SSL also authenticates servers and, optionally, authenticates clients to prove the identities of parties engaged in secure communication. It also provides data integrity through an integrity check value. In addition to protecting against data disclosure, the TLS/SSL security protocol can be used to help protect against masquerade attacks, man-in-the-middle or bucket brigade attacks, rollback attacks, and replay attacks.
- **Interoperability** - TLS/SSL works with most Web browsers and on most operating systems and Web servers, including the Microsoft Windows operating system, UNIX, Novell, Apache, Netscape

Enterprise Server, and Sun Solaris. It is often integrated in news readers, LDAP servers, and a variety of other applications.

- **Algorithm flexibility** - TLS/SSL provides options for the authentication mechanisms, encryption algorithms, and hashing algorithms that are used during the secure session.
- **Ease of use** - Because you implement TLS/SSL beneath the application layer, most of its operations are completely invisible to the client. This allows the client to have little or no knowledge of the security of communications and still be protected from attackers.

There are a few limitations to using TLS/SSL, including: **Increased processor load** (This is the most significant limitation to implementing TLS/SSL. Cryptography, specifically public key operations, is CPU-intensive. As a result, performance varies when you are using SSL. Unfortunately, there is no way to know how much performance you will lose. The performance varies, depending on how often connections are established and how long they last. TLS uses the greatest resources while it is setting up connections) and **Administrative overhead** (A TLS/SSL environment is complex and requires maintenance; the system administrator must configure the system and manage certificates)

16.4 SSH

One Secure Shell, or SSH, is a cryptographic (encrypted) network protocol for initiating text-based shell sessions on remote machines in a secure way.

This allows a user to run commands on a machine's command prompt without them being physically present near the machine. It also allows a user to establish a secure channel over an insecure network in a client-server architecture, connecting an SSH client application with an SSH server.

History

SSH1 and the SSH-1 protocol were developed in 1995 by Tatu Ylönen, a researcher at the Helsinki University of Technology in Finland. After his university network was the victim of a password-sniffing attack earlier that year, Ylönen whipped up SSH1 for himself. When beta versions started gaining attention, however, he realized that his security product could be put to wider use.

In July 1995, SSH1 was released to the public as free software with source code, permitting people to copy and use the program without cost. By the end of the year, an estimated 20,000 users in 50 countries had adopted SSH1.

Also in 1995, Ylönen documented the SSH-1 protocol as an Internet Engineering Task Force (IETF) Internet Draft, which essentially described the operation of the SSH1 software after the fact. It was a somewhat ad hoc protocol with a number of problems and limitations discovered as the software grew in popularity. These problems couldn't be fixed without losing backward compatibility, so in 1996, SCS introduced a new, major version of the protocol, SSH 2.0 or SSH-2, that incorporates new algorithms and is incompatible with SSH-1. In response, the IETF formed a working group called SECSH (Secure Shell) to standardize the protocol and guide its development in the public interest. The SECSH working group submitted the first Internet Draft for the SSH-2.0 protocol in February 1997.

In 1998, SCS released the software product "SSH Secure Shell" (SSH2), based on the superior SSH-2 protocol. However, SSH2 didn't replace SSH1 in the field, for two reasons. First, SSH2 was missing a number of useful, practical features and configuration options of SSH1. Second, SSH2 had a more restrictive license.

This situation promises to change, however, as a result of two developments: a loosening of the SSH2 license and the appearance of free SSH-2 implementations.

On Unix-like systems, the list of authorized public keys is typically stored in the home directory of the user that is allowed to log in remotely, in the file ~/.ssh/authorized_keys. This file is respected by ssh only if it is not writable by anything apart from the owner and root. The ssh-keygen utility produces the public and private keys, always in pairs. SSH also supports password-based authentication that is encrypted by automatically generated keys. In this case the attacker could imitate the legitimate server side, ask for the password, and obtain it (man-in-the-middle attack). However, this is possible only if the two sides have never authenticated before, as SSH remembers the key that the server side previously used. The SSH client raises a warning before accepting the key of a new, previously unknown server. Password authentication can be disabled.

SSH provides multiple mechanisms for authenticating the server and the client. Two of the commonly used authentication mechanisms are password based, and key based authentication. Although password based authentication is also secure, it’s advisable to use key based authentication instead.

SSH protocol version 2 is the default protocol used these days. This is due to some major advancements in version 2 compared to version 1. The workflow of the SSH login is almost same as that of version 1, however there are some major changes done in the protocol level. Some of these changes include improved encryption standards, Public key certification, much better message authentication codes, reassignment of session key(which is altered every hour) etc.

Some of the most important characteristics of SSH are:

- **Privacy communication** - means that the connection, which provides a remote shell login, must be encrypted to prevent eavesdropping.
- **Integrity check** – There must be a mechanism that checks whether the data sent by either sides was not altered or tampered with.
- **Identity of both server and client must be provided to each other**.
- SSH Tunneling
- TCP port forwarding

How does SSH work

The SSH protocol employs a client-server model to authenticate two parties and encrypt the data between them.

The server component listens on a designated port for connections. It is responsible for negotiating the secure connection, authenticating the connecting party, and spawning the correct environment if the credentials are accepted.

The client is responsible for beginning the initial TCP handshake with the server, negotiating the secure connection, verifying that the server's identity matches previously recorded information, and providing credentials to authenticate.

An SSH session is established in two separate stages. The first is to agree upon and establish encryption to protect future communication. The second stage is to authenticate the user and discover whether access to the server should be granted.

Negotiating Encryption for the Session

When a TCP connection is made by a client, the server responds with the protocol versions it supports. If the client can match one of the acceptable protocol versions, the connection continues. The server also provides its public host key, which the client can use to check whether this was the intended host.

At this point, both parties negotiate a session key using a version of something called the Diffie-Hellman algorithm. This algorithm (and its variants) make it possible for each party to combine their own private data with public data from the other system to arrive at an identical secret session key.

The session key will be used to encrypt the entire session. The public and private key pairs used for this part of the procedure are completely separate from the SSH keys used to authenticate a client to the server.

The basis of this procedure for classic Diffie-Hellman is:

1. Both parties agree on a large prime number, which will serve as a seed value.
2. Both parties agree on an encryption generator (typically AES), which will be used to manipulate the values in a predefined way.
3. Independently, each party comes up with another prime number which is kept secret from the other party. This number is used as the private key for this interaction (different than the private SSH key used for authentication).

4. The generated private key, the encryption generator, and the shared prime number are used to generate a public key that is derived from the private key, but which can be shared with the other party.
5. Both participants then exchange their generated public keys.
6. The receiving entity uses their own private key, the other party's public key, and the original shared prime number to compute a shared secret key. Although this is independently computed by each party, using opposite private and public keys, it will result in the same shared secret key.
7. The shared secret is then used to encrypt all communication that follows.

The shared secret encryption that is used for the rest of the connection is called binary packet protocol. The above process allows each party to equally participate in generating the shared secret, which does not allow one end to control the secret. It also accomplishes the task of generating an identical shared secret without ever having to send that information over insecure channels.

The generated secret is a symmetric key, meaning that the same key used to encrypt a message can be used to decrypt it on the other side. The purpose of this is to wrap all further communication in an encrypted tunnel that cannot be deciphered by outsiders.

After the session encryption is established, the user authentication stage begins.

Authenticating the User's Access to the Server

The next stage involves authenticating the user and deciding access. There are a few different methods that can be used for authentication, based on what the server accepts.

The simplest is probably password authentication, in which the server simply prompts the client for the password of the account they are attempting to login with. The password is sent through the negotiated encryption, so it is secure from outside parties.

Even though the password will be encrypted, this method is not generally recommended due to the limitations on the complexity of the password. Automated scripts can break passwords of normal lengths very easily compared to other authentication methods.

The most popular and recommended alternative is the use of SSH key pairs. SSH key pairs are asymmetric keys, meaning that the two associated keys serve different functions.

The public key is used to encrypt data that can only be decrypted with the private key. The public key can be freely shared, because, although it can encrypt for the private key, there is no method of deriving the private key from the public key.

Authentication using SSH key pairs begins after the symmetric encryption has been established as described in the last section. The procedure happens like this:

1. The client begins by sending an ID for the key pair it would like to authenticate with to the server.
2. The server check's the authorized_keys file of the account that the client is attempting to log into for the key ID.
3. If a public key with matching ID is found in the file, the server generates a random number and uses the public key to encrypt the number.
4. The server sends the client this encrypted message.
5. If the client actually has the associated private key, it will be able to decrypt the message using that key, revealing the original number.
6. The client combines the decrypted number with the shared session key that is being used to encrypt the communication, and calculates the MD5 hash of this value.
7. The client then sends this MD5 hash back to the server as an answer to the encrypted number message.
8. The server uses the same shared session key and the original number that it sent to the client to calculate the MD5 value on its own. It compares its own calculation to the one that the client sent

back. If these two values match, it proves that the client was in possession of the private key and the client is authenticated.

A short list of differences between SSH v1 and SSH v2:

- Diffie-Hellman key is used instead of the server key for sharing the session key in v2 protocol
- No Rhosts support in SSH v2
- SSH protocol version 1 only allows negotiation of the symmetric encryption algorithm, all other things are hard corded(mac, compression etc)
- SSH 2 supports certificates for public keys used
- A SSH 2 server can dictate the client to use multiple authentication methods in a single session to succeed. However SSH v1 only supports one method per session
- SSH version 2 allows the change of session key periodically.

chapter 17 secure storage

The ever-increasing amount of valuable digital data both at home and in business needs to be protected, since its irrevocable loss is unacceptable. Cloud storage services promise to be a solution for this problem. In this study we have examined the security mechanisms of seven cloud storage services: CloudMe, CrashPlan, Dropbox, Mozy, TeamDrive, Ubuntu One, and, Wuala.

We identified three categories of security requirements:

Transport Security (to secure communication between client and server)

- communication confidentiality and integrity
- server authentication
- suitable cryptography

Encryption (to disable the provider to examine stored data)

- client-side encryption of data
- client-side encryption of file names
- non-deterministic generation of encryption keys
- suitable cryptography

Secure File Sharing (to protect documents shared by a closed group)

- clear description which flavor of sharing is used
- obfuscated link
- no indexing by external search engines
- reversible sharing
- uninvited users are excluded by cryptographic means

17.1 transport security

Cloud storage providers usually provide client software which assists users in setting up their synchronization or backup schemes on the local devices. The actual transmission of all data with the remote storage servers is also handled by the client software. The server must authenticate itself to the client and all communication should be encrypted and its integrity ensured.

It is important to use appropriate cryptographic functions. All primitives, like symmetric and asymmetric encryption functions and hash functions should be up to date. This includes the algorithms as well as their parameters, like key lengths. If keys are generated this should be done by a secure high-entropic key generator.

Algorithms and protocols should always be public, as stated by Kerckho's principle". Keeping these things secret is always a risk, that does not increase security but decreases it dramatically. Developing a cryptographic protocol is a very difficult task. In the past, even protocols designed by well-respected experts have failed. So it is in most cases a bad idea to invent a new algorithm for a well known problem, especially if a widely accepted solution is available.

The standard protocol TLS offers an established solution for transport security. There should be severe reasons for replacing it by something else for the same ask.

17.2 encryption

The main reason to use a cloud storage provider, for both individuals and companies, is to always have a backup of valuable data which is off-premises yet easily accessible. The data itself should be protected in such a way that even in the event of a successful attack, the contents of the stored data remain confidential. To this end, all data needs to be stored on the remote servers in encrypted form.

There are several cryptographically secure encryption schemes available which can be used freely. Cloud storage providers often offer a general encryption of all data stored on their servers using a company key which is known only to them. This may prevent data theft from external attackers, but does not protect against any attacks which include theft of the encryption key or internal attacks conducted by personnel who are able to gain access to these keys. Therefore, all data should be encrypted on the client system before the data is transmitted into the cloud using a key unknown to the service provider.

Standalone software may be used to encrypt all data on the client system, but this has drawbacks: The software has to be installed, administrated and operated on all client systems in addition to the client software of the cloud storage provider. The key used to encrypt the data needs to be distributed to all devices which are used to access the stored data. In the event that this key is lost, the data can never be decrypted again. As a precautionary measure, all keys used to encrypt data could be integrated into some kind of key escrow system to guard against data loss.

All keys that are used for encryption should be generated at (pseudo) random. This requirement ensures that two cryptograms of the same clear text are different.

17.3 secure file sharing

Sharing files appears in three different flavors: Sharing files with other subscribers of the same service, Sharing files with a closed group of non-subscribers, Sharing files with everybody.

In any case the service should describe clearly which flavor of sharing is used.

The files that are being shared should only be accessible to the closed user group that was decided by the sharing user. It should also be possible to revert sharing for each individual file. A list of files currently being shared by the user could be accessed in the web interface or in the client application. It should be possible to deal with different access rights and at any time the sharing user should be able to grant, edit or remove individual access rights. If client-side encryption is used, sharing files should not weaken the security level. In particular, the cloud storage service provider should not be able to read shared files. If client-side encryption is used, a uninvited user should be excluded by cryptographic means from the closed user group. In particular, this means that an encryption key that is known by the uninvited user can no longer be used for the encryption of new files.

Sharing files with everybody has the security requirement to hide informations about user names. If there is no client-side encryption with keys individual to the user the service knows which clients share files even though the clients do not use file sharing feature.

17.4 CloudMe

Transport Security

- CloudMe does not encrypt the data transferred between the server and the client.

Encryption

- CloudMe does not encrypt the files that are stored on the server. Since the communication between the client and the server is also not encrypted, attackers are able to intercept every file a user uploads to the service.

Sharing

- The required password length of one character is not enough to guard against any attacks.

17.5 CrashPlan

Transport Security

- CrashPlan does not use SSL/TLS to secure the communication between a client and the CrashPlan server, instead a self-made, unpublished protocol is used.
- The communication between the client and other backup destinations is not secured by SSL/TLS. This is a disadvantage if these destinations are outside of an intranet.

Encryption

- The key used to encrypt the files is chosen at random during the installation of the software and will be referred to as data key.
- CrashPlan provides multiple options to encrypt files, which are explained in high detail on the CrashPlan website: Securing the data encryption key with the account password, Securing data encryption key with a private password or Using an exclusively local stored data encryption key.
- With the default option, it is possible for CrashPlan to decrypt and access the data stored on their servers, since both the data key and the password used to secure the data key are known to CrashPlan.
- With the second option, CrashPlan can't access the encrypted data unless the user is using the web restore function, where he has to enter his private password which is then used to unlock the data key.
- Using the third option, the private data key has to be entered when using the web restore function. The user is responsible to store this key in a secure and safe way.

Sharing

- CrashPlan does not support file sharing or file publication. When using the friend feature to store files on the computers of other users, these files are stored encrypted and are not accessible to the other users.

17.6 Dropbox

Transport Security

- Dropbox uses TLS to encrypt the communication between the client application and the server. The communication between the browser and the web interface is encrypted by using HTTPS.

Encryption

- Dropbox uses AES-256 to encrypt data stored on its servers. The data will not be encrypted at the client; instead Dropbox encrypts the data after the upload on the server-side using its own encryption key.
- While the encryption of data in transit meets the requirements, Dropbox has not optimally implemented the encryption of the stored data. Since Dropbox itself encrypts the data on the server-side, users cannot be sure by cryptographic means that all stored data is highly confidential.

Sharing

- Sharing files with subscribers meets our previously described requirements. Dropbox has some problems when sharing files with non-subscribers / everybody.
- Using a simple script which iterated through possible URL combinations we were able to search for the existence of specific files inside the Public folder.

17.7 Mozy

Transport Security

- Mozy uses TLS to encrypt the communication between the client application and the server. The communication between the browser and the web interface is encrypted by using HTTPS.

Encryption

- All data is generally encrypted at the client, before being transferred to the Mozy server. The user can select between two encryption methods. The default is to use an 448-bit Blowfish encryption key which is provided by and therefore known to Mozy. As an alternative, the user can use a personal 256-bit AES encryption key.

Sharing

- Mozy does not offer to share files with other people.

17.8 TeamDrive

Transport Security

- The web interface uses HTTPS to secure the communication between browser and server. The communication between clients and the TeamDrive server uses HTTP, enhanced by a self-made, unpublished protocol.

Encryption

- TeamDrive uses AES-256 for file encryption. The data is encrypted at the client before it is transmitted to the server. Every space uses an individual AES key for file encryption. These AES keys are not based on a password and are not known to TeamDrive, therefore TeamDrive is not able to access any data stored by its users on their servers.

Sharing

- Sharing files is supported by cryptographic means. When sharing files with another subscriber, the TeamDrive server sends the public key of the invitee. The inviting user encrypts the AES key of the space with this key. In doing so, he trusts that the received key is authentic. An invitation including the encrypted space key is sent to the invitee. After decrypting the space key with his secret key the invitee can access and decrypt all files inside the space.

17.9 Ubuntu One

Transport Security

- Ubuntu One uses SSL/TLS to encrypt the communication between the clients and the server. The communication between the browser and the web interface is encrypted by using HTTPS.

Encryption

- Ubuntu One does neither encrypt data using the client software nor on the server. Thus, the data itself is not protected against unauthorized access from attackers who successfully circumvent authentication security of the service. Ubuntu makes the missing encryption very clear in their FAQ.

Sharing

- Sharing files with subscribers. The sharing of files between registered Ubuntu One users meets our requirements. Sharing files with everybody: The URL of a published file consists of a mix of numbers and upper- and lower-case characters. The URL does not contain a user name which impedes information gathering.

17.10 Wuala

Transport Security

- Wuala uses a proprietary client / server-communication protocol instead of the standardized and well-known SSL/TLS protocol to secure the communication between a client and the Wuala server. According to Wuala, integrity checks are used to protect transmitted data in transit. In combination with the convergent encryption scheme employed by Wuala, the absence of encryption during transmit allows attackers to sniff exchanged messages and attempt information gathering attacks.

Encryption

- The idea behind Wuala's encryption scheme is an untrusted file system that is secured by cryptographic methods. The employed system is an implementation of a folder tree structure for cryptographic file systems called Cryptree that has been published by Grolimund from ETH Zurich. The trust anchor is a symmetric root key r which is derived from the user's password. Wuala calculates individual keys for every directory and individual keys for every file. All of them are accessible via r. They can be given to partners in order to share data.

Sharing

- Security of shared files depends on the invitee. Sharing between registered users meets all mandatory requirements. The files are not readable by Wuala. When sharing files with another subscriber, the Wuala server sends the public key of the invitee to the inviting user. He encrypts a key for the invitee. The result is sent via Wuala to the invitee. In doing so, the inviting user trusts that the received keys are authentic. Sharing files with non sub-subscribers is based on secret web links. Knowing the link is equal to having the right to access the file. The value included in the URL is sufficiently large, appears to be random and is only valid for this folder. The files shared with this method are not indexed by search engines, and sharing can be reversed anytime.

17.11 Conclusion

The study shows that most of the analyzed cloud storage providers are aware of the extreme importance of data security and privacy. Nevertheless, none of the examined cloud storage providers meets all mandatory security requirements.

Transport Security was a problem for CrashPlan, TeamDrive and Wuala because they deny the usage of SSL/TLS. Instead they use unpublished, self-made protocols { a very error-prone approach. CloudMe does not take any measure to protect the security of files during transmission.

Encryption was a problem for CloudMe, Dropbox and Ubuntu One because they do not use client-side encryption, thus the provider is able to read the data. Mozy does not encrypt filenames. The convergent encryption scheme used by Wuala enables attacks by a server-side attacker.

Sharing of data was a problem for CloudMe, Dropbox, TeamDrive and Wuala. Problems occur if files are shared with non-subscribers on the principle of a long, unpredictable URL. CloudMe does not obfuscate this URL adequately. Dropbox gives an unclear description with regard to sharing details, TeamDrive is weak when uninviting a group member and Wuala enables information gathering by including the user name in public URLs. CloudMe does not prevent search engines from accessing the workspace.

This study, as presented by the Fraunhofer Institute for Secure Information Technology, is not meant to nominate the best cloud storage service that fits all needs of any possible user. This is impossible. Instead, we want to give some advice, that may help selecting a service for a particular use case. First of all, evaluate your use case, make clear, which problem you want to solve by using a storage service. Align your requirements to the features of the examined services. In addition to concrete security requirements it is recommendable to observe some extra aspects. It is worthwhile to consider using more than one service to reduce the impacts of service downtime. Further, calculation of the time to recover all data from the cloud is recommended. Depending on the individual amount of data, this may take several days. Having a plan for a provider change in the future reduces the dependency on a particular provider (provider lock-in). This will be relevant, for example, if the chosen provider is getting to expensive or is not longer compliant with governmental rules.

chapter 18 P2P security

18.1 introduction

Peer-to-Peer (P2P) networking is a fairly popular concept. Networks such as BitTorrent and eMule make it easy for people to find what they want and share what they have. For one thing, sharing files on your computer with anonymous and unknown users on the general public Internet goes against many of the basic principles of securing your computer. For one thing, sharing files on your computer with anonymous and unknown users on the general public Internet goes against many of the basic principles of securing your computer.

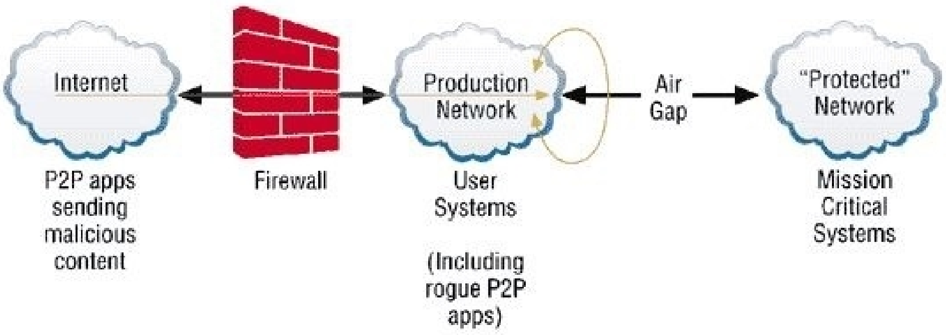
However, in order to share files on your computer and sometimes in order for you to access files on other computers within a P2P network such as BitTorrent, you must open a specific TCP port through the firewall for the P2P software to communicate. In effect, once you open the port you are no longer protected from malicious traffic coming through it.

More technically, a P2P network is a special type of computer network that exhibits self-organization, symmetric communication, and distributed control. The network is self-organizing in that there is typically no centralization of resources. As a result, link capacity is typically distributed throughout peers in the network, and as a result control is distributed, as well.

18.2 need for security

In these turbulent times you would think that P2P security would be the least of the world's problems. However corporate fraud and loss of revenue due to attacks on their internal networks has brought P2P to the forefront in the IT world. Napster was the headliner but since its high profile court case more and more P2P applications have been causing the corporate world headaches, which it could do without. With better security protocols this headache could be turned into a valuable asset for the corporate world and for the world.

The diagram on the next page illustrates the gaps in security when using P2P applications. We can see that we are letting these applications get inside our networks. The security of our “secure” network is now in jeopardy.



Following on from this, is the question of what must we protect ourselves against. We must outline the elements that our important to use, before we address the issue of the security. The main points of this are connection control, access control, operation control, anti-virus, and of course the protection of the data stored on our machines.

18.3 security mechanisms

One All security mechanisms deployed today are based on either symmetric/secret key or asymmetric/public key cryptography, or sometimes a combination of the two. Here we will introduce the basic aspects of the secret key and public key techniques and compare their main characteristics.

Secret Key Techniques:
Secret key techniques are based on the fact that the sender and recipient share a secret, which is used for various cryptographic operations, such as encryption and decryption of messages and the creation and verification of message authentication data. This secret key must be exchanged in a separate out of bound procedure prior to the intended communication (using a PKI for example).

Public Key Techniques:
Public Key Techniques are based on the use of asymmetric key pairs. Usually each user is in possession of just one key pair. One of the pair is made publicly available, while the other is kept private. Because one is available there is no need for an out of band key exchange, however there is a need for an infrastructure to distribute the public key authentically. Because there is no need for pre-shared secrets prior to a communication, public key techniques are ideal for supporting security between previously unknown parties.

Asymmetric Key Pairs:
Unlike a front door key, which allows its holder to lock or unlock the door with equal facility, the public key used in cryptography is asymmetric. This means just the public key can encrypt a message with relative ease but decrypt it, if at all, with considerable difficulty.

Besides being one-way functions, cryptographic public keys are also trapdoor functions- the inverse can be computed easily if the private key is known.

18.4 protocols

Mechanisms for establishing strong, cryptographically verifiable identities are very important. These are industry standard authorization protocols that allow peers to ensure that they are speaking with the intended remote system.

Secure Sockets Layer (SSL) protocol:
For protection of information transmitted over a P2P network, some P2P’s employ the industry-standard Secure Sockets Layer (SSL) protocol. This guarantees that files and events sent will arrive unmodified, and unseen, by anyone other than the intended recipient. Moreover, because both peers use SSL both sides automatically prove who they are to each other before any information is transferred over the network. The protocol provides mechanisms to ensure tamperproof, confidential communications with the right counterpart, using the same, well-proven techniques used by all major website operators to protect consumer privacy and financial information transmitted on the Internet.

IPSec technologies:
Most VPNs (virtual private networks) use IPSec technologies, the evolving framework of protocols that has become the standard for most vendors. IPSec is useful because it is compatible with most different VPN hardware and software, and is the most popular for networks with remote access clients. IPSec requires very little knowledge for clients, because the authentication is not user-based, which means a token (such as Secure ID or Crypto Card) is not used. Instead, the security comes from the workstation's IP address or its certificate (e.g. X.509), establishing the user's identity and ensuring the integrity of the network. An IPSec tunnel basically acts as the network layer protecting all the data packets that pass through, regardless of the application.

Public Key Infrastructure (PKI) An industry standard:
A full-featured X.509 Public Key Infrastructure (PKI) over a Secure Sockets Layer (SSL) network backbone - the combination of X.509 PKI authentication and SSL transport encryption is the established cryptographic standard for Internet e-commerce.

Use of X.509 PKI authentication allows security certificates from Endeavors, or from any other recognized X.509 certificate authority, to be used to establish the true identity of any peer device when it comes on-line. Use of SSL point-to-point security encryption enables each pair of peers that communicate with each other to have a unique key for that pairing. The advantage of SSL encryption is that when a peer goes off-line from a community, all its unique pairing keys become invalid, but no pairing keys between other members of the community are affected.

What about VPN Security?

The key word in "virtual private networks" is private. The last thing a business wants is to have sensitive corporate information end up in the hands of some hacker, or worse, the competition. Fortunately, VPNs are widely considered extremely secure, despite using public networks.

Why are they secure?

In order to authenticate the VPNs users, a firewall will be necessary. All VPNs require configuration of an access device, either software- or hardware-based, to set up a secure channel. A random user cannot simply log in to a VPN, as some information is needed to allow a remote user access to the network, or to even begin a VPN handshake. When used in conjunction with strong authentication, VPNs can prevent intruders from successfully authenticating to the network, even if they were able to somehow capture a VPN session.

18.5 the future of P2P security

The constant running theme in the security of P2P is that of trust. Trust in the other users who we interact with, and trust within the software vendors who supply us with the necessary applications. If we could have more faith in this trust, or feel a greater sense of security, maybe the development of P2P would grow even faster than it is already doing.

Many proposals are already being studied. People are acknowledging that security is an area P2P must address, if it is to be accepted by consumers.

Users Gaining Their Own Trust:

One very interesting idea recently proposed, is that of users gaining trust within the P2P community. All users would be assigned a unique digital signature, like IP, but per user and not per machine. Associated with this digital signature would be a level of trust. Trust levels would vary from say zero, to twenty. Depending on a users behaviour in the past, their trust level would either be promoted on the grounds of valid use of the network, of demoted with acts of malice and misuse.

The proposed plan states that all users trust level would begin at a rather low level. This is merely to combat unwanted users creating new accounts, and abusing the new high trust level immediately. Users would have to be active on the network for some time (say one/two months), before their trust level would be pushed up a level. Users could also keep a local record of other known users, to which they may want to share a local trust level, and bypass the global trust policy.

This proposal has many hurdles to jump of course. It is merely an idea to be developed. The problem that it overcomes is that of the centralized managing authority. Instead, the users of the network are the authority. If the general public continuously try to demote a user, he/she will eventually lose all their privileges, and become silenced from other users. This idea also rewards genuine users, for their efforts in keeping the network policed, and for their good behaviour on the network.

The idea is possibly a bit too naive, as we all know that must humans(especially adolescent ones), will do the exact opposite of what they are meant to do, if given no choice. In other words, people do not like to be told what to do.

Biometrics:

Biometrics involves the use of a person’s unique characteristics to authenticate them. Traits that are commonly utilized include a person’s facial image, signature, fingerprint or retinal pattern. One key feature of biometrics is that the user is no longer required to remember any passwords or store any key data, a major weakness in conventional authentication systems.

Ultimately, the technology could find its strongest role as an integrated and complementary piece of a larger authentication system, perhaps in combination with the cryptographic certificates mentioned above, rather than a stand-alone single point of defense.

In the future, many experts foresee biometrics both playing a key role in enabling public key infrastructure deployment by protecting public and private keys and residing in smart card technology in an effort to support personalized e-commerce.

Quantum Key Cryptography:

For the short term, The US Government is adopting a new encryption standard called Advanced Encryption Standard (AES), which will eventually replace DES. "When approved, the AES will be a public algorithm designed to protect sensitive government information well into the 21st century." If that's true, what will be used after AES?

One idea currently being proposed is the notion of Quantum Cryptography. Many modern encryption systems depend on the difficulty in mounting brute force attacks on secret keys, due to processing and time constraints. Although still at the theoretical stage, the performance improvements given by a hypothetical quantum computer would render many algorithms useless.

Obviously new encryption algorithms would be needed. Quantum encryption uses photon state as the key for encoding information. According to the Heisenberg uncertainty principle, it's impossible to discover both the momentum and position of a particle at any given instant in time. Therefore, in theory, an intruder can't discover secret keys based on particle state information; the intruder would need the actual particle to decipher any data encrypted with a key.
Unfortunately this concept is, for the moment, incredibly complex to implement. IBM scientists constructed the first working prototype of a quantum key distribution (QKD) system in the late 80’s. Back then they could transmit quantum signals just under half a meter through open air. Today, fiber optic cables can transmit the signal up to 31 miles. This still isn't very far, but it is definitely good progress. And although we might not see QKD come to market for quite some time, the technology sounds incredibly promising.

18.6 key points to consider when using a P2P network

1. **Don't Use P2P on a Corporate Network** : At least, don't ever install a P2P client or use P2P network file sharing on a corporate network without explicit permission- preferably in writing. Having other P2P users downloading files from your computer can clog the company's network bandwidth. That is the best-case scenario. You may also inadvertently share company files of a sensitive or confidential nature. All of the other concerns listed below are also a factor.
2. **Beware The Client Software:** Installing the software might cause system crashes or problems with your computer in general. Another factor is that the client software is typically hosted from every participating user's machine and could potentially be replaced with a malicious version.
3. **Don't Share Everything :** Many users unknowingly designate the root "C:" drive as their shared files folder which enables everyone on the P2P network to see and access virtually every file and folder on the entire hard drive, including critical operating system files.
4. **Scan Everything :** You should treat all downloaded files with the utmost suspicion. As mentioned earlier, you have virtually no way of ensuring that what you downloaded is what you think it is or that it doesn't also contain some sort of Trojan or virus.

18.7 conclusion

It is obvious from the above that security is a crucial issue when it comes to designing and implementing P2P systems. At the moment it is probably the main inhibiting factor for the growth of P2P. It is vital that users become confident in the ability of the security measures being utilized to protect them, in order for P2P technology to reach its full potential. At the moment, security measures in general are failing to inspire consumer confidence, a problem that must be addressed immediately.

chapter 19 personal profiles, data verification

The **S**.

19.1 paragraph 1

One

19.2 par 2

One iteration

chapter 20 electronic vote

The **S**.

20.1 paragraph 1

One

20.2 par 2

One iteration

chapter 21 electronic payments

The **S**.

21.1 paragraph 1

One

21.2 par 2

One iteration

chapter 22 smart cards

22.1 definition

A smart card, or smart card is the generic name of a small electronic circuit integrated into a small object of small size, made of plastic, metal or other materials that look like a bank card, it is generally used to control a person's access to certain resources.

This card usually contains information that identifies a person, authentication data, generic data, or specific commands for some applications.

It is usually used with a card reader such as POS and ATM for electronic payments and. Cash withdrawals if we talk about bank cards, or phone/tablet if we talk about a card. SIM.

22.2 classification

Multiple type of smart cards appeared during the last three decades:

- Bank cards that help people to make transactions of billions of dollars every day

  - SIM cards facilitate billions of conversations and calls but they also ensure the entrance to social media

  - Badges of identification of identity. Used by the people that need access to companies or institutions with the purpose of security.

  - National health cards, staying permit or electronic passport, identification cards and authentication cards for authorities of citizens.

- Smart card market

  - SIM card with a market value of 52%

  - Credit or debit cards with a market value of 32%

  - Government cards or medical cards with a market value of 6%

  - Electronics producers: mobile phones, tablets, GPS devices with a market value of 5%

  - Cards made by mobile phones operators, cards for public transport and parking, cards for TVs with a market value of 5%

- Other types

  - Memory cards or microprocessor cards

  - With contact or contactless

22.3 smart cards history

Roland Moreno patented the memory card in 1974. Since 1977, Bull CP8, SGS Thomson and Schlumberger were the first three producers who started creating cards.

In March 1979, Michael Hugon from Bull CP8 was the first who designed and created a card that had a microprocessor which combined local memory and the processor. He invented the smart card.

- 1979 – early evolution for banking sector
- 1995 – first SIM card-sized

- 1999 – first national ID card (Finland)
- 2001 – Department of Defense of USA creates the first military card which contained the credentials for the physical control and security authentication
- 2003 – Micro-SIM launches
- 2005 – First electronic passport (Norway)
- 2012 – Introduction of Micro-SIM
- 2018 – first contact less bio-metric card. E SIM (1mm or 0.039 thickness)
- 2019 – first 5G SIM

22.4 technical details

Smart cards can have the following characteristics:

- Similar dimensions as a credit card (ID-1 of standard ISO/IEC 7810 with the dimension of 85.6x53.98mm or ID-000 25x15mm for SIM cards)
- Integrated security system which can prevent the data alteration from the card
- It is administered by a system which, in a safe manner can exchange data and update the data from the card-size
- It can communicate with the ATMs, POS, DIP readers, etc
- They are made out of plastic usually but there are other material options, such as metal.

These cards have two ways of communicating with the reader: contact or contactless; cards that need contact usually have an area of 1cm2 which interacts with the reader and the contactless use an antenna for data exchange with a transfer rate of 106-848 kbit/s.

None of these types contain a power source, they rely on the reader device in order to power up.

There is also a third type which is a hybrid card, which implements both technologies in order to provide a diversity of higher redundancy.

22.5 smart cards and electronic commerce

Smart cards can be used in electronic commerce, over the Internet, though the business model used in current electronic commerce applications still cannot use the full potential of the electronic medium. An advantage of smart cards for electronic commerce is their use customize services. For example, in order for the service supplier to deliver the customized service, the user may need to provide each supplier with their profile, a boring and time-consuming activity. A smart card can contain a non-encrypted profile of the bearer, so that the user can get customized services even without previous contacts with the supplier.

22.6 advantages

The first main advantage of smart cards is their flexibility. Smart cards have multiple functions which simultaneously can be an ID, a credit card, a stored-value cash card, and a repository of personal information such as telephone numbers or medical history. The card can be easily replaced if lost, and, the requirement for a PIN (or other form of security) provides additional security from unauthorised access to information by others. At the first attempt to use it illegally, the card would be deactivated by the card reader itself.

The second main advantage is security. Smart cards can be electronic key rings, giving the bearer ability to access information and physical places without need for online connections. They are encryption devices, so that the user can encrypt and decrypt information without relying on unknown, and therefore potentially untrustworthy, appliances such as ATMs. Smart cards are very flexible in providing authentication at different level of the bearer and the counterpart. Finally, with the information about the user that smart cards can provide to the other parties, they are useful devices for customizing products and services.

Other general benefits of smart cards are:

- Portability
- Increasing data storage capacity
- Reliability that is virtually unaffected by electrical and magnetic fields.

22.7 disadvantages

The plastic or paper card in which the chip is embedded is fairly flexible. The larger the chip, the higher the probability that normal use could damage it. Cards are often carried in wallets or pockets, a harsh environment for a chip and antenna in contactless cards. PVC cards can crack or break if bent/flexed excessively. However, for large banking systems, failure-management costs can be more than offset by fraud reduction.

The production, use and disposal of PVC plastic is known to be more harmful to the environment than other plastics. Alternative materials including chlorine free plastics and paper are available for some smart applications.

If the account holder's computer hosts malware, the smart card security model may be broken. Malware can override the communication (both input via keyboard and output via application screen) between the user and the application. Man-in-the-browser malware could modify a transaction, unnoticed by the user. Banks like Fortis and Belfius in Belgium and Rabobank in the Netherlands combine a smart card with an unconnected card reader to avoid this problem. The customer enters a challenge received from the bank's website, a PIN and the transaction amount into the reader. The reader returns an 8-digit signature. This signature is manually entered into the personal computer and verified by the bank, preventing point-of-sale-malware from changing the transaction amount.

Smart cards have also been the targets of security attacks. These attacks range from physical invasion of the card's electronics, to non-invasive attacks that exploit weaknesses in the card's software or hardware. The usual goal is to expose private encryption keys and then read and manipulate secure data such as funds. Once an attacker develops a non-invasive attack for a particular smart card model, he or she is typically able to perform the attack on other cards of that model in seconds, often using equipment that can be disguised as a normal smart card reader. While manufacturers may develop new card models with additional information security, it may be costly or inconvenient for users to upgrade vulnerable systems. Tamper-evident and audit features in a smart card system help manage the risks of compromised cards.

chapter 23 biometrics

The **S**.

23.1 paragraph 1

One

23.2 par 2

One iteration

chapter 24 crypto currencies

The **S**.

24.1 paragraph 1

One

24.2 par 2

One iteration

-

Biblography

[CA] – Certification Authority - https://www.luxtrust.com/what-is-a-certification-authority-ca/

[CISSP] – S. Harris, F. Maymi, Certified Information Systems Security Professional Exam Guide, 8-th edition, Mc Graw Hill Education, 2019

[NS-WIT] – What is a Network Security policy? - https://www.algosec.com/security-policy/

[PIL-SEC] – The five pillars of information security - https://www.siaonline.org/what-are-the-5-pillars-of-information-security/

[PKIS] – Totul despre PKI - https://www.ssl2buy.com/wiki/public-key-infrastructure

[RA] – Registration Authority - https://www.luxtrust.com/what-is-a-registration-authority-ra/

[RSA-W] – RSA Wiki - https://simple.wikipedia.org/wiki/RSA_algorithm

[TOP5] – Most popular SSL Certificate Authorities - https://premium.wpmudev.org/blog/ssl-certificate-authorities-reviewed/

[USPKI] – Ghid pentru PKI - https://fpki.idmanagement.gov/ca/