The Internet Security

Petri Jäppilä
Petteri Pöyhönen

1. INTRODUCTION

The security of Internet has come more and more important. Companies and private persons are dependent that their computers that are connected to networks are working and information is confident. Security is highly depended on users and administrations will.

1.1 Security classification

Security can be classified by many ways but one way to classify is Internet Engineering Task Force's (IETF) list:

- Confidentiality: Only authorised parties can access to information.
- Authentication: The parties that are using information (sending, receiving, etc) can be identified.
- Integrity: only authorised parties modify Information.
- Non-repudiation: Neither the sender nor receiver of message is able to deny the transmission.
- Access control: The use of target computer can be controlled. Controlling includes that user can access only to information, which is user is authorised.
- Availability: Information must be available to authorised parties when needed. This is the most difficult requirement for security systems.

1.2 Methods to gain security in Internet

Security can be carried out by many means. It can be implemented in all layers of transportation. This study covers some most used methods. The Internet is complicate and in some ways uncontrolled environment. This causes that; it is the easiest way to protect the information transportation is implement the concealment in the endpoints of connection. The endpoints are the only spots that the sender and receiver can secure. Nevertheless more security can be gained also in the net.

Trusted third party scheme is coming more and more important when it's needed to have non-repudation for large number of users. Trusted Third Party must be trusted by sender and receiver.

Public key is certificate from trusted third party which is set available for other users. User itself can have different key for signature and encrypting

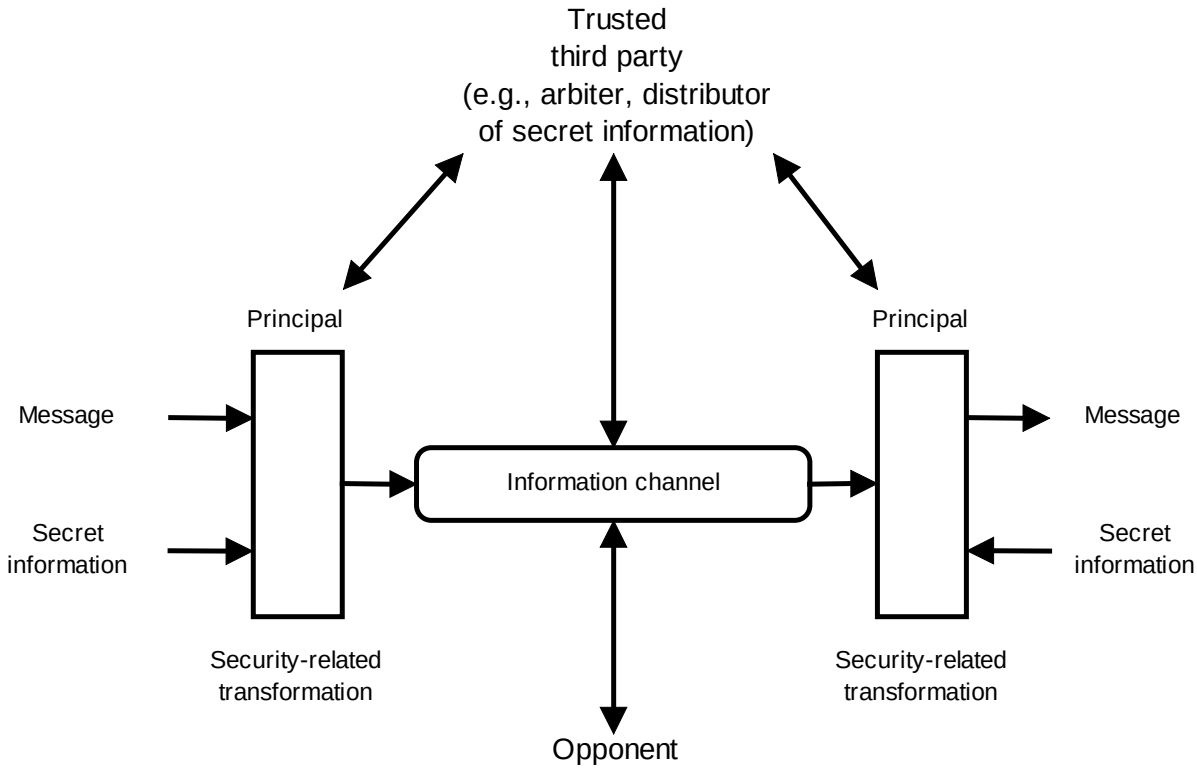

Figure 1. This figure represents trusted third party infrastructure.[1]

Encryption can be done in:

- Internet layer (IP)
- Transportation layer (TCP)
- Application layer

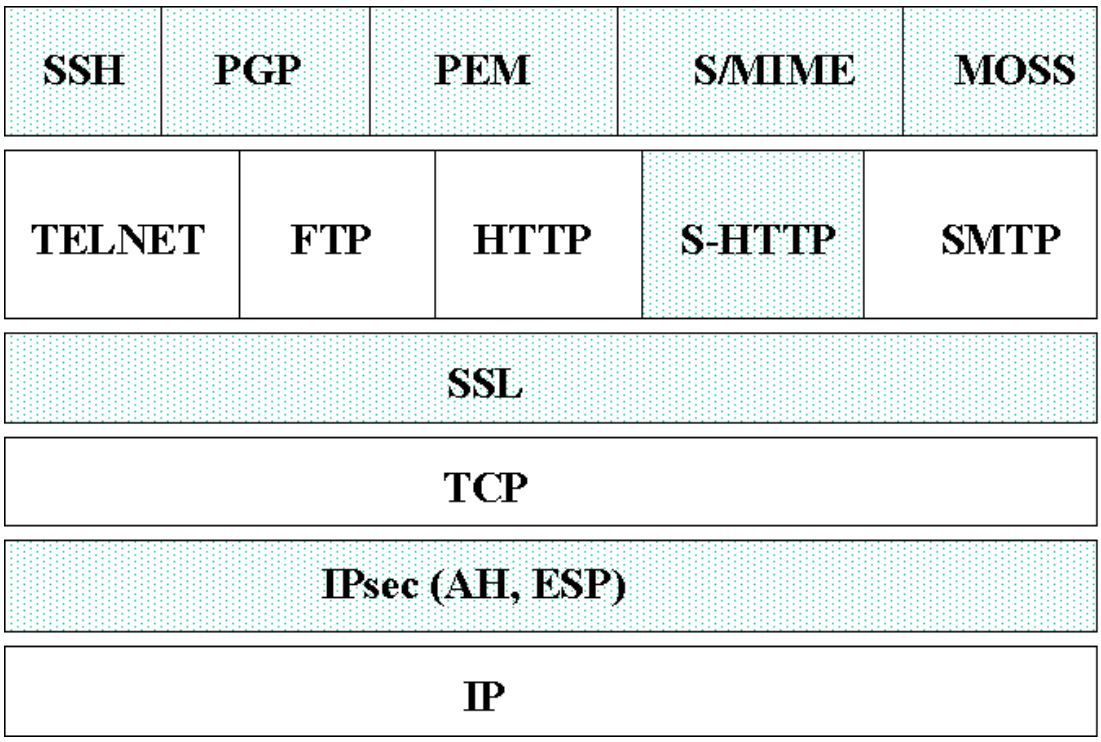

Figure 2. This figure represents how security can be gained in different layers.[2]

1.3 Security attacks

Security can be threaded with passive or active attacks. They can be represented in many ways. Following figure shows threats. Only the picture c represents passive attack.

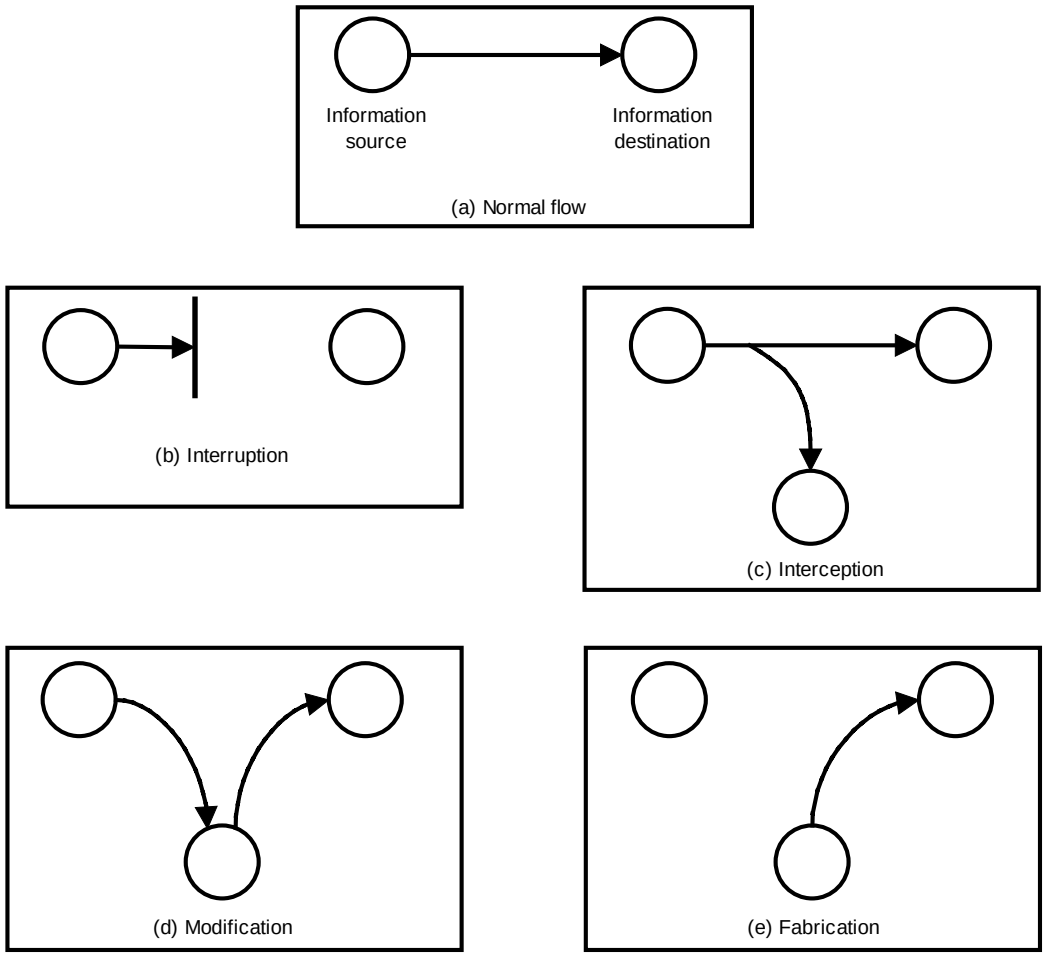

Figure 3. This figure represents security threats.

1.3.1 Passive attacks

Passive attacks are more difficult to detect than active. Passive attacks do not influence directly to data flow. There are two types of passive attacks: traffic analysis and finding out the content of message. Traffic analysis is almost impossible to avoid if messages are carried though commercial Internet. At least the endpoints of messages can been seen.

Traffic analysis is more useful than intuition might first exhibit. It can be used to analyse, if two companies are trading large number of messages, which might conclude in right situation that they are really discussing a merger. In future analyses can be certainly used to estimate competitors Internet commerce.

Extracting the messages is real easy in some circumstances. If in LAN messages are carried without encryption, the enemy can extract all messages, which is really harmful because, for example in Telnet and FTP passwords travel in the clear. The user can’t notice that his or her password is revealed.

1.3.2 Active attacks

Active attacks influence to data flow. Active attacks can interrupt the availability, modify or fabricate the integrity of service. For example the intruder can modify the routing tables to redirect the traffic.

Masquerade: An entity pretends to be a different entity.

Replay: The capture of data unit and retransmission to produce unauthorised effect.

Modification: Some parts of an original message are changed, or messages are delayed or reordered to produce unwanted effect.

Denial of service: Normal use or management of service is disturbed. The messages can be suppressed or another way to degrade the service is overloading or disabling the network.

The denial of services can be caused by E-mail bombing. Bombing can be executed with various ways: sending non-stopping long messages with large, binary attachments, sending mails with forged addresses to newsgroups. Another way to cause denial of service is TCP SYN flooding. In TCP SYN attack a large number of SYN messages are sent server to flood its buffer.

2. CRYPTOGRAPHY

2.1 Introduction

A word cryptology is an old term that was used already by ancient creeks. Word itself consists of two parts, the first one is "krypton", which means hidden and the second one is "logos", which means word. The cryptology is an old method and it was used already by Julius Caesar to secure his communication. It consists of cryptography and cryptanalysis.

Cryptography consists of the study and the practice of encryption and decryption of data, so those specific targets can only access it. These kinds of systems are called as cryptosystems. A definition of cryptography is not so straightforward and depending on source, it may vary (e.g., which is part of cryptography and which is not).

A cryptanalysis is the study and the practice of how to compromise cryptography mechanisms (cryptosystems), e.g., how to decrypt a ciphertext without a key, which is encrypted using DES-algorithm.

A source data that is in understandable format (either for human or for some program) and which will be the input of cryptosystem is called plaintext (or cleantext).

A destination data that is in non-understandable format and which will be the output of cryptosystem is called ciphertext.

Encryption is a mathematical method that transforms a plaintext to a ciphertext. Decryption is inverse method compared with encryption. A key is a parameter of encryption/decryption.

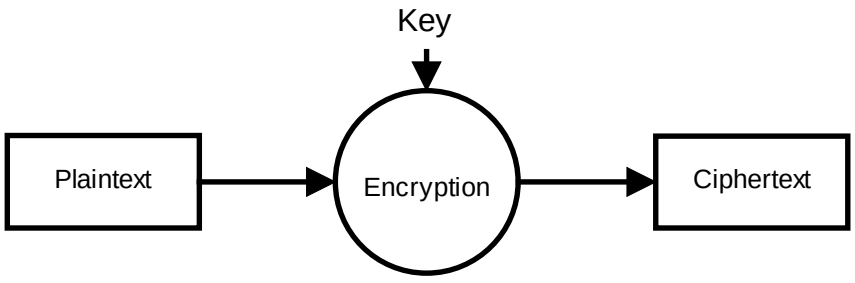

Figure 4. This figure represents basic encryption scheme.

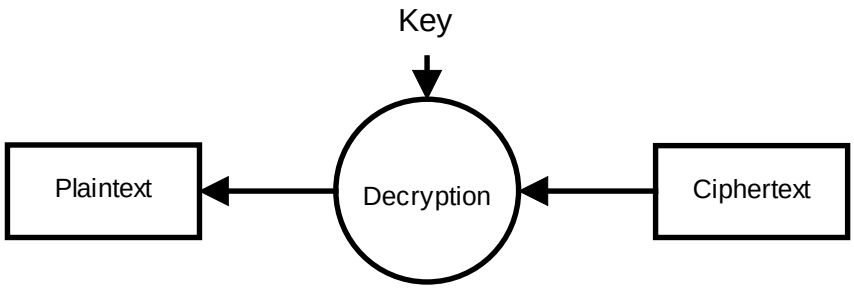

Figure 5. This figure represents basic decryption scheme.

However, a modern cryptography is more than just an encoding and decoding data. Authentication, digital signatures and digital timestamps are also a part of modern cryptography.

Encryption algorithms have usually keys and each ciphertext has its own key. Only using a correct key can decrypt this ciphertext. For example, when Julius Caesar encrypted his data by replacing every "A" letter with "D" letter, every "B" letter with "E" letter, and so on. This algorithm can be called for example "Shift by *n*" and the used key value was 3.

Brute-force search (attack) is a method, which relies on computers' processing power and it just generate all possible values. For example, let say that $f(x)=y$ and y is known and f can be computed, it is possible to find x by trying every possible x. If we use this method against travelling salesman problem, it simply generates all possible routes and compares the distances. This solution will work and it is simple to implement, but at the same time it is not the most efficient. (Travelling salesman problem: Given a set of towns and the distances between them, determine the shortest path starting from a given town, passing through all the other towns and returning to the first town.)

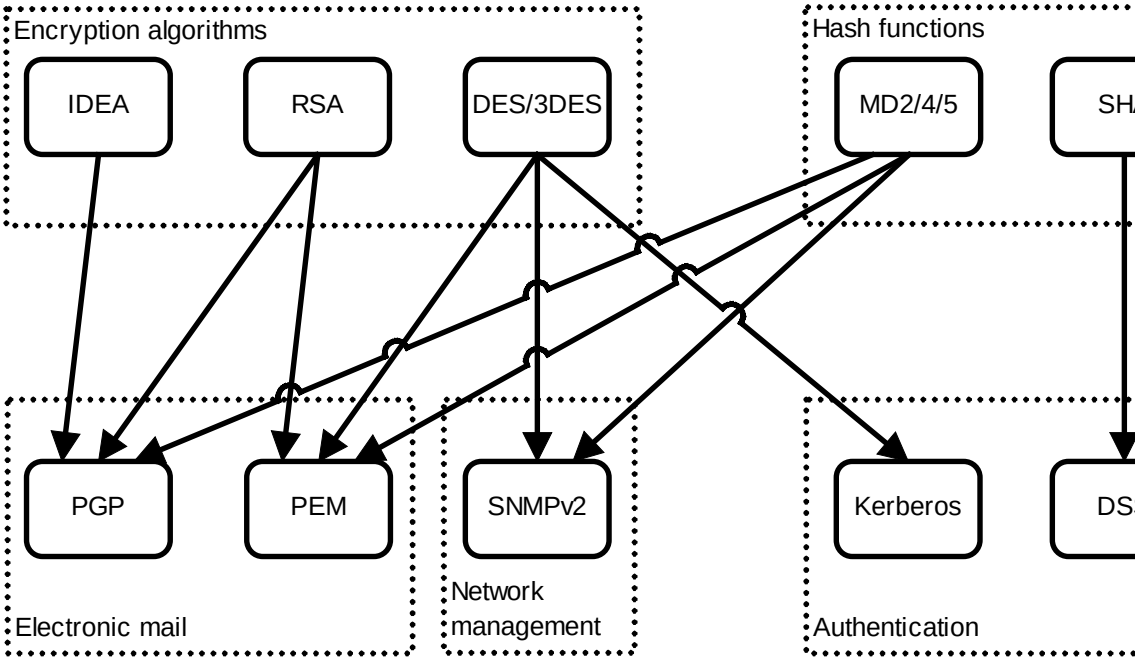

Figure 6. This figure represents what algorithms and functions are used by some common applications.[7]

2.2 Strength of cryptosystem

- the secrecy of cryptosystem should rely on the secrecy of the key rather than the secrecy of the algorithm. A cryptosystem should be so strong that if an algorithm is a public, secrecy can be reached only by keeping secrecy of a key. (Strength of algorithm)

- a key space should be large enough which has straight correlation to the size of a used key (number of bits). (Size of the key)

- a strength of algorithm has a reverse correlation to a number of backdoors of algorithm (backdoor means that based on ciphertext it is possible to find out used key). These backdoors can be seen always as a security risk. The existence of these backdoors might be a one reason why sources of some algorithm are not public and products, which use these kinds of algorithms, may not be the most secure products available. (Possible backdoors of algorithm)

- a cryptosystem should be resistant against all known attacks method. This is an important thing while designing a new cryptosystem. But also sometimes an existing cryptosystem might need upgrading because some

new attack method is appeared and if it is known that our system does not resist that attack.

If our cryptosystem fulfils all these features mentioned above, is it strong enough? Sometimes it can be proved mathematically that a particular cryptosystem is secure, but not in all cases. What is secure enough, depends on our needs. What is the lifetime of data to be secured (2 hours, 1 week, 3 years)? How confidential secured data is and how harmful is if secured data is revealed? It is noticeable that the cryptosystem that is secure today may not be secure after 5 years. This is partly because computers are developing so fast that year after year a cost of single operation of processor is cheaper and cheaper. For example, if a brute-force attack is impractical against our cryptosystem today, after 5 years a breaking of our system might be a piece of cake using same attack method.

Key Size	**Number of Alternative Keys**	**One Encryption/μs**	**10^6 Encryption/μs**
32 bits	$2^{32} = 4{,}3 * 10^9$	2^{31}μs=35,8 minutes	2,15 ms
56 bits	$2^{56} = 7{,}2 * 10^{16}$	2^{55}μs = 1142 years	10,01 h
128 bits	$2^{128} = 3{,}4 * 10^{38}$	2^{127}μs = $5{,}4 * 10^{24}$	$5{,}4 * 10^{18}$ years

Table 1. This table represents time required for exhaustive key search.[7]

2.3 Cipher

A cipher is part of an algorithm and is it's cryptographic ”core”. A block cipher is a symmetric cipher and it encrypts a block of data, size of block might vary, at the time and then goes on to the next block, and so on (e.g., RSA). A product cipher is a block cipher, which iterates a several weak operations (substitution, transposition, modular addition/multiplication and linear transformation) and Shannon introduced it at the first time[3]. Examples of modern product chippers are LUCIFER[4] and DES[5]. Feistel cipher (sometimes called DES-like cipher) is a sub-class of product cipher and they can be recognise that they operate on one half of the ciphertext at each round and then swap the ciphertext halves after each round.[6]

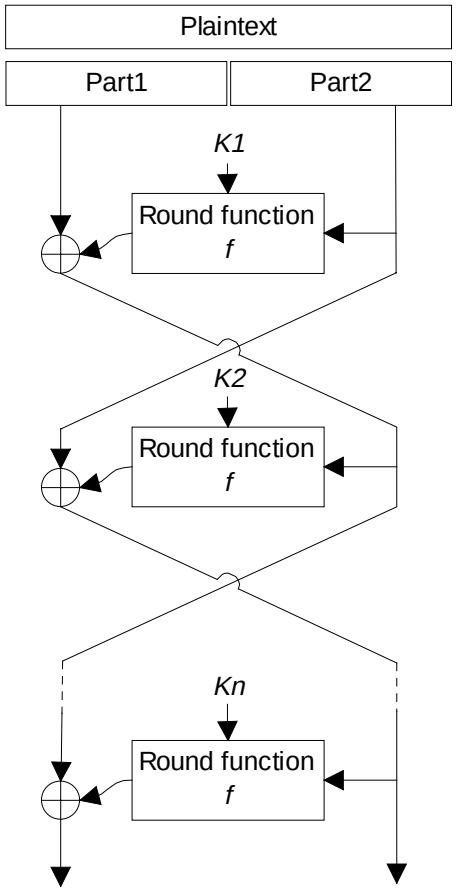

Figure 7. This figure represents functionality of Feistel cipher.

A following table represents the main parameters of some product cipher:

Cipher	**Block length**	**key size**	**Number of rounds**
LUCIFER	128	64	16
DES	64	56	16
IDEA	64	128	8

2.4 Secret Key Cryptography (symmetric)

Secret key models are also called as conventional encryption model. In this model the same key is used to encrypt and to decrypt a data. Because of this, a sender and a receiver must have a same key.

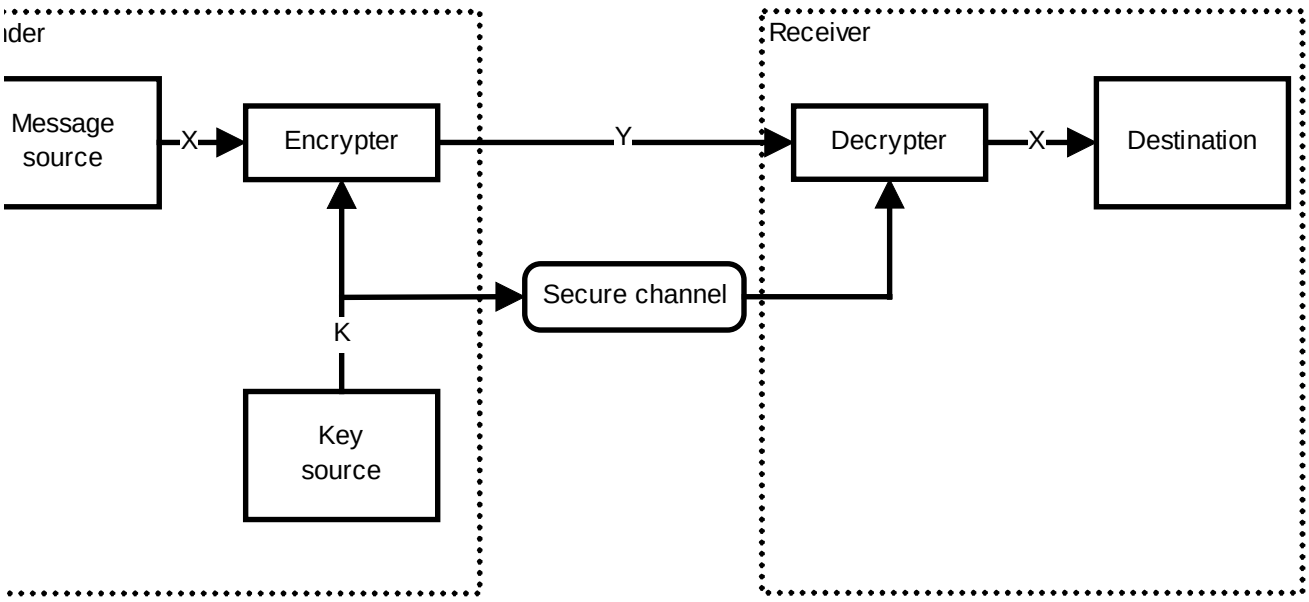

Figure 8. This figure represents a functionality of symmetric model.[7]

In figure above, X is plaintext, Y is ciphertext and K is the key used by encryption and by decryption. Secure channel is some undeclared media for distribute key K and this key distribution is one common problem while using symmetric encryption models. The lack of security in key distribution scheme exposes the security of the hole cryptosystem. There are several schemes for key distribution and in some of them a third party is involved.

Examples of symmetric cryptosystems:

- 3DES
- DES
- IDEA
- Blowfish

Nowadays, a sufficient size of the key for symmetric encryption models is 100 bits. In the future, when hardware and software related on computing are developing this key size is getting too small.

2.5 Public Key Cryptography (asymmetric)

An encryption scheme introduced by Diffie and Hellman in 1976. In public key models encryption and decryption use a different key, but used keys have some mathematical dependencies so that a data which is encrypted using a certain

key can be decrypted only by using a pair of the encryption key. Key consists of a public and a private part. A private key is personal key and it is not distributed whereas a public key is for distributing.

Asymmetric cryptosystems are used to do authentication, non-repudiation and integrity of data. It is also used in symmetric cryptosystems to distribute a secret key (see “Secure channel” in Secure key cryptography –chapter). Asymmetric models are computationally more time consuming than corresponding symmetric ones. This is one reason why symmetric cryptosystems are used more often than asymmetric ones to encryption.

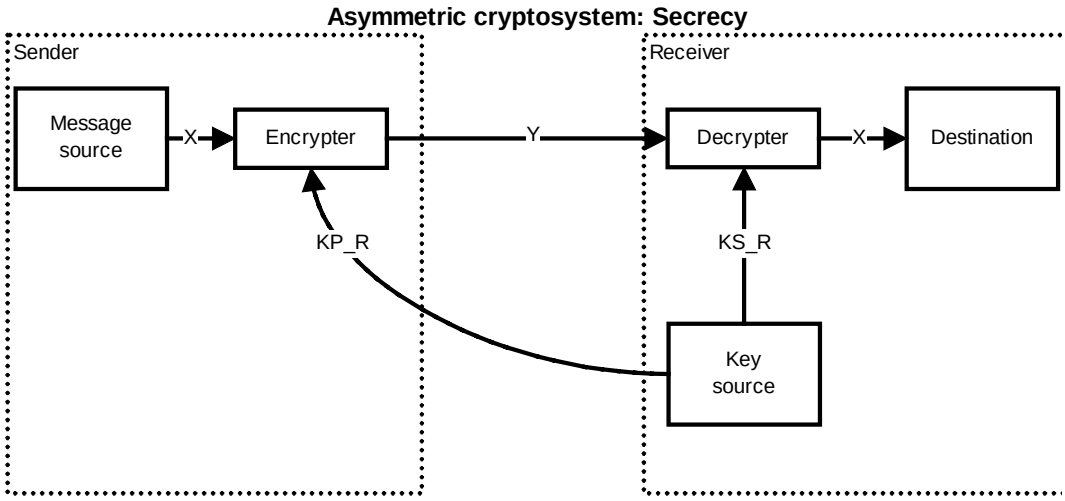

Figure 9. This figure represents a functionality of asymmetric model in secrecy.[7]

In figure above, X is plaintext, Y is ciphertext, KP_R is the public key of receiver used by encryption and KS_R is the private key of receiver used by decryption. A person how is receiving some encrypted data must share his/her public key to persons whose are sending some encrypted data to him/her. While distributing a public key, it should not necessary be so secrecy than in the case of distributing secure key of symmetric cryptosystem.

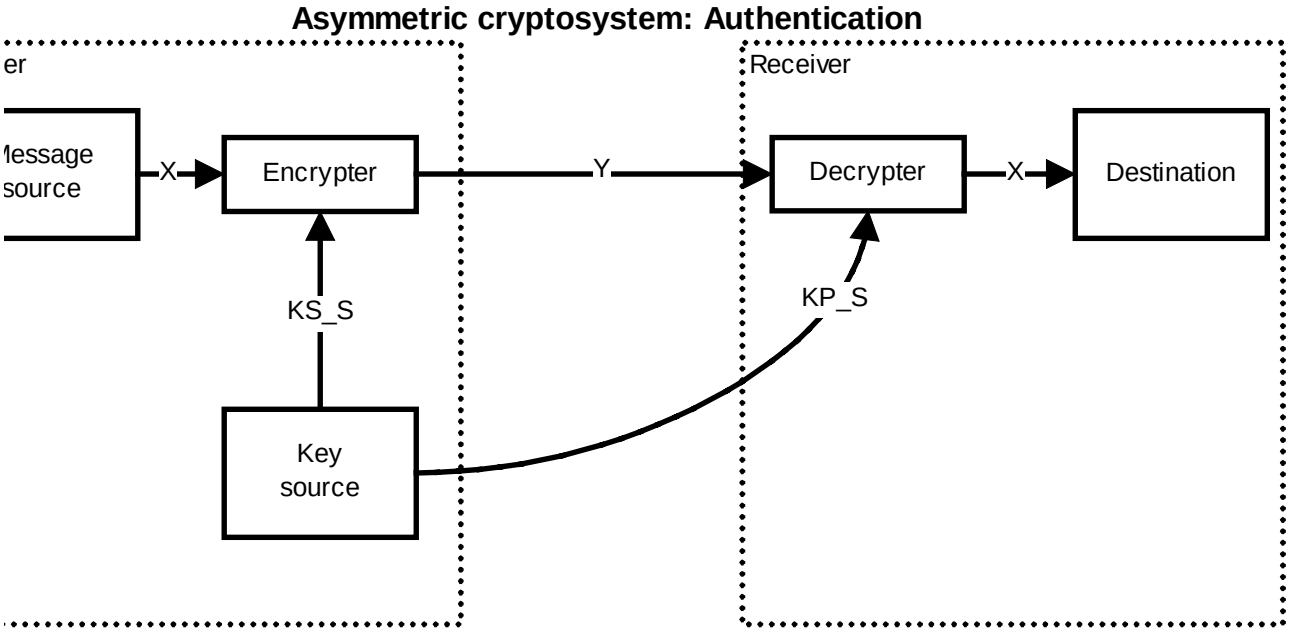

Figure 10. This figure represents a functionality of asymmetric model in authentication.[7]

In figure above, X is plaintext, Y is ciphertext, KP_S is the public key of sender used by decryption and KS_S is the private key of sender used by encryption. People how is sending some encrypted data must share his/her public key to persons whose are sending some encrypted data to him/her. It is impossible to change ciphertext Y without having an access the sender's private key KS_S and therefore this scheme is providing authentication for source and for data integrity. The hole ciphertext Y is a digital signature. It is noticeable that this scheme does not protect against outsiders (confidentiality), because everyone who have access the ciphertext Y and the public key of sender KP_S could decrypt ciphertext Y.

It is possible to provide both the authentication and confidentiality using asymmetric cryptosystem.

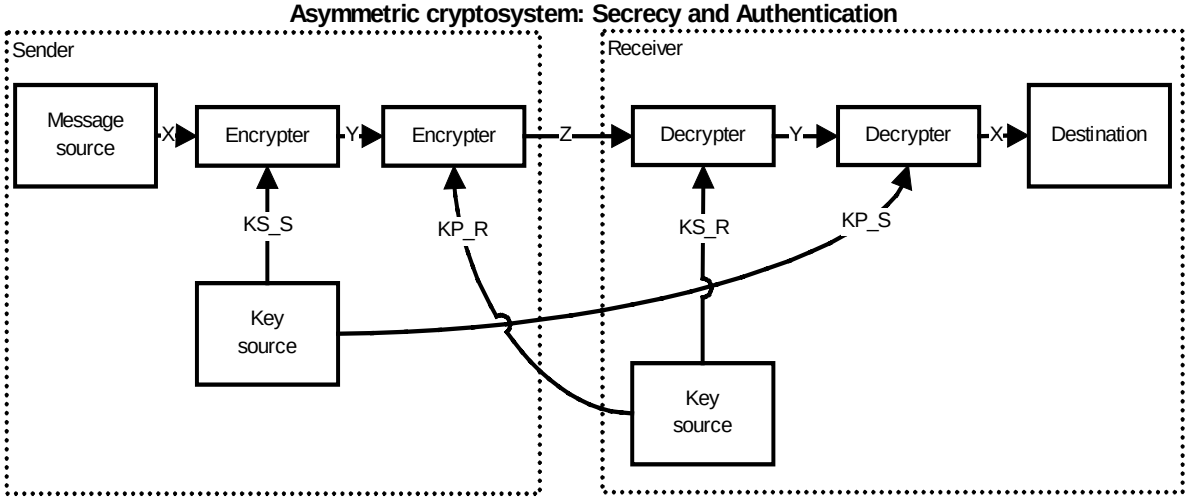

Figure 11. This figure represents a functionality of asymmetric model in secrecy and authentication.[7]

In figure above, X is plaintext, Y is ciphertext based on plaintext X encrypted using the private key of sender (KS_S) and Z is ciphertext based on ciphertext Y encrypted using the public key of receiver (KP_R). The public key of sender KP_S is used by decryption of ciphertext Y and the private key of receiver KS_R is used by decryption of ciphertext Z. A digital signature is achieved by encrypting plaintext X using sender's private key KS_S and confidentiality is achieved by encrypting ciphertext Y using receiver's public key KP_R. The disadvantages of this scheme is that asymmetric algorithm is used four times for each communication instead of two.

Examples of asymmetric cryptosystems:

- RSA
- DSS
- LUC
- Diffie-Hellman

Nowadays, a sufficient size of the key for asymmetric encryption models is 1000 bits. In the future, when hardware and software related on computing are developing this key size is getting too small and when a used keys expired, it is recommended that the used key sizes are checked whether they are large enough or not.

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
LUC	Yes	Yes	Yes
DSS	No	Yes	No
Diffie-Hellman	No	No	Yes

Table 2. This table represents features of some asymmetric algorithms.[7]

2.6 Encryption Algorithms/Standards

2.6.1 RSA (Rivest, Shamir, and Adleman)

RSA[8] is a public-key cryptosystem, which can be used both encryption and authentication. Ron Rivest, Adi Shamir, and Leonard Adleman invented it in 1977[9].

"RSA is part of the Society for World-wide Interbank Financial Telecommunications (SWIFT) standard, the French financial industry's ETEBAC 5 standard, the ANSI X9.31 rDSA standard and the X9.44 draft standard for the U.S. banking industry. The Australian key management standard, AS2805.6.5.3, also specifies RSA." [10]

According to RFC, it was intended to use to construct a digital signatures and digital envelopes a following ways:

digital signatures

a message-digest algorithm (e.g., MD5) is first used to reduce a message digest of the data to be signed and after that an octet string is encrypted using RSA together with signer private key. There are now the encrypted message digest and the original content, which are presented together by a certain syntax, which is represented PKCS#7[11], to achieve a digital signature. This is compatible with PEM (Privacy Enhanced Mail).

digital envelopes

a content-encryption algorithm (e.g., DES) with a content-encryption key is first used to encrypt the data to be enveloped and the used key is encrypted using the RSA with recipient's publickey (content). The encrypted content and the encrypted key are presented together by a certain syntax, which is

represented PKCS#7[11], to achieve a digital envelope. This is compatible with PEM.

However, the role of RSA is to encrypt a different part of data. Encryption process of RSA is described more specific in its RFC. In that algorithm are two large prime numbers *p* and *q*, and their product *n=pq*, which is also called as modulus.

The speed of RSA is depending on this modulus *n* in a following way:

- public key operations take $O(k^2)$ steps
- private-key operations take $O(k^3)$ steps
- key generation takes $O(k^4)$ steps

where k is the size of the modulus (number of bits).[10]

The key size in RSA is equal to the size of modulus *n* and user can normally decide number of bits used in key. Secrecy of RSA depends on the number of bits used in key; greater the number of bits is, more secure the encryption is. But the greater the key size is, the slower are the RSA operations. There is also other point, which affects the secrecy of RSA; it is more secure that those two large prime numbers *p* and *q* are so called "strong" primes instead of "weak" ones. RSA Laboratories[10] currently recommends a following key sizes:

- key sizes of 768 bits for personal use
- 1024 bits for corporate use
- 2048 bits for extremely valuable keys (e.g., the root-key pair used by a certifying authority)

The secrecy of the RSA based on the assumption that factoring is difficult. An attack, which includes an easy method for factoring large prime numbers, would break the RSA.

Encryption and authentication can done without sharing keys, because everyone can send encrypted data or verify signed data. But only persons whose have correct private key can decrypt or sign data.

If we compare the speed of RSA and other symmetric block ciphers based on software implementation, then DES is about 100 as fast as RSA. In hardware

based implementations, DES is from 1 000 to 10 000 as fast as RSA depending on the implementation.[10]

RSA is part of Sun, Novell, Microsoft and Apple operating systems. It is also used in secure telephones, in Ethernet networkcards and on smart cards. RSA is licensed by about 350 companies around the world (see http://www.rsa.com/html/licensees.html) and it is used in different internet protocols concerning security (e.g., S/MIME, S/WAN). [10]

2.6.2 DES (Data Encryption Standard)

DES (U.S. Governments' Data Encryption Standard) is defined FIPS 46. FIPS are Federal Information Processing Standards published by NTIS (National Technical Information Service). DES defines DEA (Data Encryption Algorithm), which was developed based on a LUCIFER product cipher (developed by IBM) and it is also defined in the ANSI standard X9.32. The NSA and the National Bureau of Standards (NBS, now the National Institute of Standards and Technology, NIST) had a considerable role while developing final stages of DES. NIST has recertified DES 1993 FIPS 46-1, but NIST has indicated that they are not recertified DES any more, because AES[12] in under development and it will replace DES.[10]

DEA is a symmetric product cipher, a 16-round Feistel cipher, and its' block size is a 64 bits and key size is a 56 bits (8 parity bits are removed). It was originally designed for hardware implementations, but there are also software-based implementations. Encrypting and decrypting are done using same key, which involves key distribution. DEA can be used also for single-user encryption. Because of "short" key, to improve security of DES, keys should change frequently.

There are different methods how a block cipher, like DES, can be used to encrypt messages, files and blocks of data. These methods are called also modes. There are four different modes, which are defined in FIPS 81[13]:

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- K-bit Cipher FeedBack (CFB)

- K-bit Output FeedBack (OFB)

ECB and CBC modes uses DES encrypt function to encrypting and decrypt function to decrypt. Feedback modes (CFB and OFB) use only encrypt function to both encrypting and decrypting data.

DES mode	**Purpose**
Electronic Code Book	Secure transmission of single value (e.g., key)
Cipher Block Chaining	General-purpose block transmission, authentication
K-bit Cipher FeedBack	General-purpose stream transmission, authentication
K-bit Output FeedBack	Stream transmission over noisy channel (e.g., satellite)

Table 3. This table represents different DES modes and their propriety for different tasks.[7]

There are also some other modes:

- Encrypt-Decrypt-Encrypt (EDE)[14]
- Error-Propagating Cipher Block Chaining (PCBC) (This mode is not defined in any standard, but it is used by Kerberos and some other applications)

FIPS 81 defines also that in the case of 7-bit ASCII data, the unused most significant bit is set to 1.

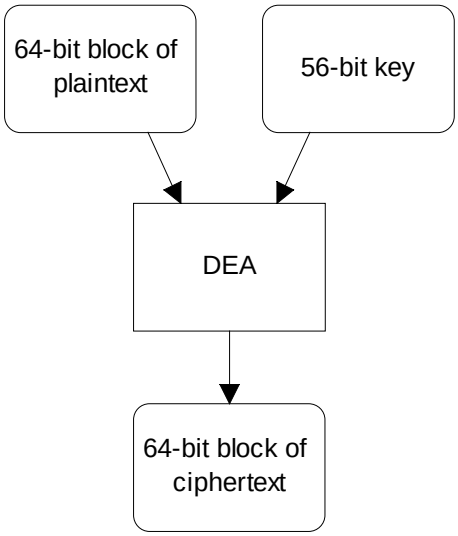

Figure 12. This figure represents an environment of DEA-algorithm (DES).

DES carries out the avalanche effect. This means that a small change of plaintext of key produce a significant changes in ciphertext. This small change may be only one bit. If these small changes do not produce a significant changes in ciphertext, an attacks could easier and faster to execute against such algorithm, because it may be possible to reduce search spaces of plaintext and key.

Because of the key size of DES, its secrecy is not at the same level as it was early 1990. Starting November 1998, DES is not allowed to use by U.S. government any more. 3DES is used instead of DES so far until AES will be ready for use.

2.6.3 3DES

3DES (Triple DES) is a product cipher, which data block size is 64-bit. There are several forms of 3DES. Some of these use two 56-bit keys, other uses three. 3DES uses multiple encryption with DES and multiple keys (three specially connected DES operations). One simple way to implement this is $C=E_{k3}(E_{k2}(E_{k1}(P)))$, where C is ciphertext, P is plaintext, E_X is encryption with key X.

There are proposed a several modes for 3DES[6]:

- Three DES encryptions with three different keys (EEE3)
- Three DES operations in the sequence encrypt-decrypt-encrypt with three different keys (EDE3)
- Same as the previous formats except that the first and third operations use the same key (EEE2 and EDE2)

The standard modes of DES can also be used in 3DES.

2.6.4 IDEA (International Data Encryption Algorithm)

IDEA was developed by Xuejia Lai and James Massey of the Swiss Federal Institute of Technology (original version[15])[7]. It is one possible candidate to replacing DES and it is included in PGP (Pretty Good Privacy).

IDEA is a symmetric block cipher, which uses 64-bit block size and 128-bit key size. Design goals of IDEA are based on general cryptographic strength (e.g., key size, block size) and easy of implementation. Structure of encryption process and decryption process are equals, but input is different and key is

processed different way. There are same modes available to use with IDEA as with DES (ECB, CBC, CFB, OFB).

2.6.5 AES (Advanced Encryption Standard)

On August 20, 1998, at the First AES Candidate Conference, NIST announced the FIFTEEN (15) OFFICIAL AES CANDIDATES:

Algorithm Name	Submitter Name(s)
CAST-256	Entrust Technologies, Inc. (represented by Carlisle Adams)
CRYPTON	Future Systems, Inc. (represented by Chae Hoon Lim)
DEAL	Richard Outerbridge, Lars Knudsen
DFC	CNRS – Centre National pour la Recherche Scientifique – Ecole Normale Superieure (represented by Serge Vaudenay)
E2	NTT – Nippon Telegraph and Telephone Corporation (represented by Masayuki Kanda)
FROG	TecApro Internacional S.A. (represented by Dianelos Georgoudis)
HPC	Rich Schroeppel
LOKI97	Lawrie Brown, Josef Pieprzyk, Jennifer Seberry
MAGENTA	Deutsche Telekom AG (represented by Dr. Klaus Huber)
MARS	IBM (represented by Nevenko Zunic)
RC6	RSA Laboratories (represented by Matthew Robshaw)
RIJNDAEL	Joan Daemen, Vincent Rijmen
SAFER+	Cylink Corporation (represented by Dr. Lily Chen)
SERPENT	Ross Anderson, Eli Biham, Lars Knudsen
TWOFISH	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson

Next AES Candidate Conference will be March 22-23, 1999, in Rome, Italy.[12]

2.6.6 Other Algorithms

- CAST (RFC 2144, The CAST-128 Encryption Algorithm)
- RC2 (RFC 2268, A Description of the RC2(r) Encryption Algorithm)

- RC5 (RFC 2040, The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS Algorithms)
- CMEA (Cellular Message Encryption Algorithm)
- LUC (Smith, P., LUC Public-Key Encryption: A Secure Alternative to RSA, Dr. Dobb's Journal, January 1993. Smith P., Lennon M., LUC: A New Public Key System, Proceedings, Ninth International Conference on Information Security, IFIP/Sec, 1993.)

2.7 One-way Hash Functions

A typical one-way hash functions input is variable-length message and it produces a fixed-length hash. Produced hash does not include any usable information about message with that hash. It is also impossible mathematically determine two messages with same hash. A one-way hash function is also called message digest function (e.g., MD2, MD4, MD5).

Here are some hash functions:

- SHA[16]
- MD2[17]
- MD4[18],[19]
- MD5[20]

2.7.1 SHA (Secure Hash Algorithm)

SHA was developed by NIST and it is defined in FIPS 180[16]. It is based on algorithm of MD4. SHA takes variable-length message as input and produces 160-bit hash for it. Input is processed in 512-bit blocks. Maximum message size is 2^{64} bits. SHA is about 25% slower than MD5 using the same hardware, but it is safer that MD5 in terms of size of hash.[7]

2.7.2 MD2 (Message-Digest Algorithm)

MD2 takes variable-length message as input and produces 128-bit message digest (fingerprint) for it. Input is processed in 512-bit blocks. It is intended to use for digital signature applications, where message is encrypted first to gain confidentiality and after that signing is done. License to use MD2 is granted for non-commercial PEM.[17]

2.7.3 MD4 (Message-Digest Algorithm)

Generally this algorithm is like MD2, but this is designed to be quite fast on 32-bit machines and it does not need large substitution tables.[18],[19]

2.7.4 MD5 (Message-Digest Algorithm)

The MD5 algorithm is an extension of the MD4 message-digest algorithm and they both have same main design goals. It is a bit slower that MD4. It was released because MD4 was adapted to use too fast and in the design of MD5 have been made some jumps backwards; its design is little more ” conservative”. MD5 is designed to be little slower but more secure that MD5.

2.8 Electronic Mail

2.8.1 PGP (Pretty Good Privacy)

PGP is a high security RSA public-key encryption application, which is released for several different platforms (e.g., MS-DOS, Unix, VAX/VMS, Win 3x, WinNT). It was written by Philip R. Zimmermann (Phil's Pretty Good(tm) Software). PGP was distributed freely ("guerrilla freeware"). PGP uses a public-key encryption algorithm claimed by US patent #4,405,829.
Because PGP based on asymmetric algorithm (RSA), it is easy to use. User does not need any secure channel for key distributing, but it is possible to gain privacy and authentication using normal common channels. PGP accepts files with any formats.

Digital signature (optional service)

After input file is defined, PGP takes that file and creates hash for it using MD5. Hash is encrypted using RSA and sender's private key. The resulting encrypted hash is digital signature for this file and using this signature receiver can be sure that sender is correct person and that file is not altered after sender was sent it (after hash was created).

Compression (optional service)

Input file and its digital signature are compressed using ZIP.

Encryption (optional service)

Input file and its digital signature are encrypted using IDEA, which is much faster than asymmetric algorithms. PGP generates randomly session key for IDEA and message is encrypted using this key. Because IDEA is symmetric algorithm, the used session key has to be sent securely to receiver. Session key

is encrypted with RSA and receiver's public key and attached to the encrypted message. Now demands of confidentiality are gained.

E-mail compatibility

If signing, compression or encryption is performed, message is in binary format. Because all E-mail programs do not accept binary files, PGP converts a raw binary file to a stream of printable characters (Radix 64 conversion), which is called ASCII armor.

Receiving PGP-message

Basically receiving is reverse sequence of sending operations. First ASCII armor is removed by converting message back to binary format. Then if message is encrypted, PGP decrypt session key (IDEA) with RSA and receiver's private key. After that a message is decrypted using IDEA and decrypted session key. If message was compressed, reversing compression algorithm decompresses it. Last phase is to compare digital signature, if it exists. Signature is decrypted with RSA and sender's public key. The resulting hash is compared the hash recalculated by PGP. If those hashes are equal, the message is genuine.

There are some limitations in existing E-mail system, which limit maximum size of message. PGP can automatically segment too large message. Segmentation is done after all other steps. The session key component and signature component appear only once in first segment. Receiving PGP just will strip off all extra mail headers and reassemble the message.

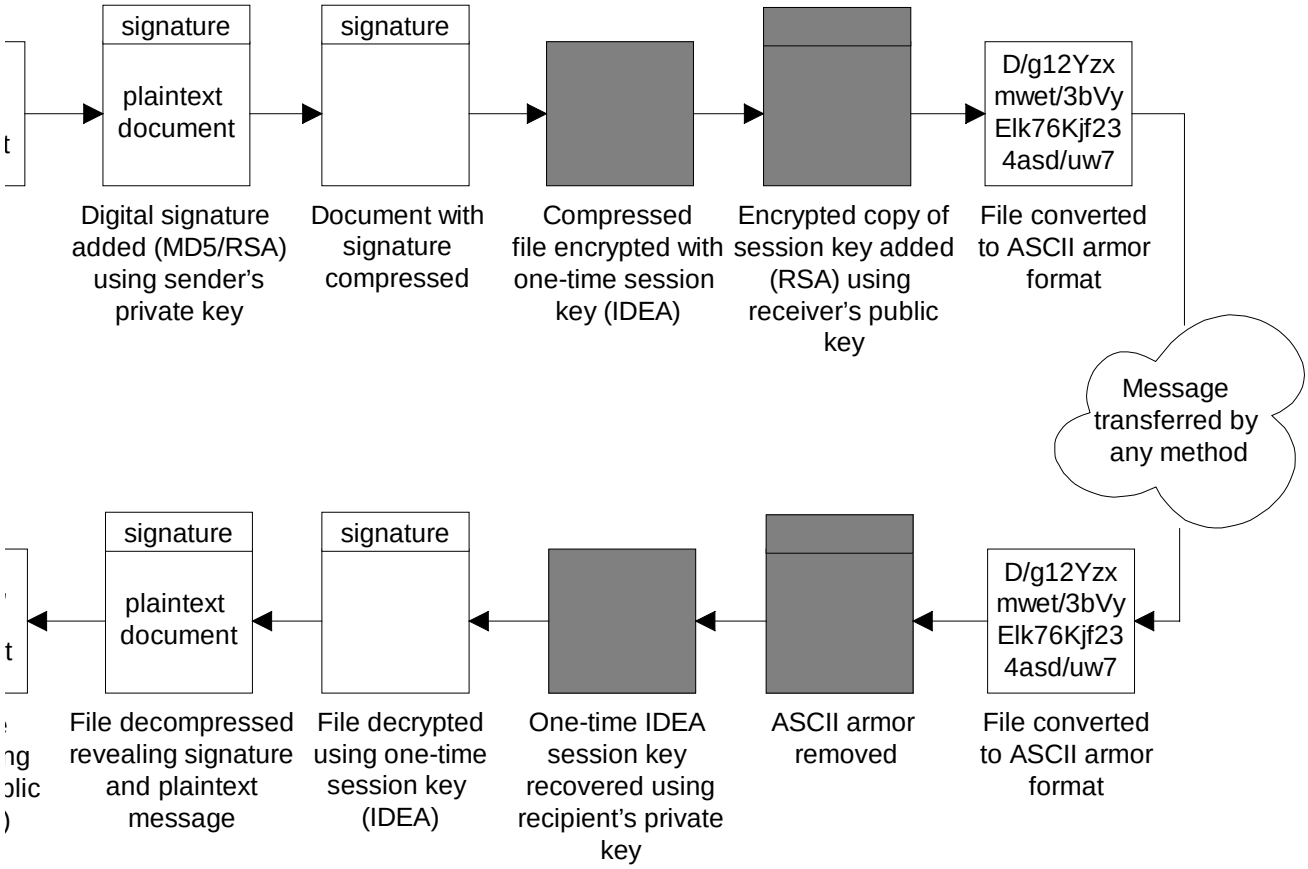

Figure 13. This figure represents the process of sending and receiving PGP-messages.[21]

Name	Encryption Algorithm	Use
Session key	IDEA	It is used to encrypt message before transmission. Session key is generated randomly and it is used only once.
Public key	RSA	It is used to encrypt used session keys. Sender and receiver must have a own copy of this key.
Private key	RSA	It is used to encrypt hash of message (message digest) and this form a digital signarure. Only a sender's private key is used.
Passphrase-based key	IDEA	It is used to encrypt sender's private keys for storage.

Table 4. This table represents all different keys used by PGP and their encryption algorithms.[7]

Further reading:

- RFC 1991 PGP Message Exchange Formats
- RFC 2015 MIME Security with Pretty Good Privacy (PGP)

2.8.2 PEM (Privacy Enhanced Mail)

PEM[22] is a draft internet standard. It defines a security services for electronic mail. Usually it is used with SMTP (Simple Mail Transfer Protocol) but can be used with other mail protocols/methods. There are four RFCs (Request for Comments) related to PEM (RFC 1421 – 1424). PEM is implemented in application layer and it is not dependant on lower layer protocols. It is compatible with a different mail transport protocols/methods, a different user interfaces, mailing lists and a different key distribution schemes.[7]

Function of PEM	**Used algorithms**
Message encryption	DES-CBC
Authentication and Digital signature	RSA with MD2 or MD5 (asymmetric encryption)
Authentication	DES-ECB or DES-EDE (3DES) with MD2 or MD5
Symmetric Key Management	DES-ECB or DES-EDE (3DES)
Asymmetric Key management	RSA, MD2
E-mail compability	Radix 64 conversion

Table 5. This table represents a different algorithms used by PEM.[7]

PEM supports both asymmetric encryption and symmetric encryption.

2.8.3 MOSS (MIME Object Security Services)

RFC 1848 MIME Object Security Services

2.8.4 S/MIME (Secure/Multipurpose Internet Mail Extension)

RFC 2311 S/MIME Version 2 Message Specification
RFC 2312 S/MIME Version 2 Certificate Handling

2.9 Authentication And Digital Signature

2.9.1 Kerberos

RFC 1510 The Kerberos Network Authentication Service (V5)
RFC 1964 The Kerberos Version 5 GSS-API Mechanism

2.9.2 DSS (Digital Signature Standard)

NIST published the Digital Signature Algorithm (DSA) in the Digital Signature Standard (DSS). It is a part of the U.S. government's Capstone project and NIST and NSA selected it.

DSS was designed only for digital signatures; it can not be used for data encrypting or key exchange. It uses SHA and its key size is fixed 512-bits. DSS uses a public-key technique. The scheme of digital signature is basically same as in RSA, but there is one important component more, which is a global key. This key is composed using set of parameters known a group of communication principals.

2.9.3 Other Algorithms

- S/KEY (RFC 1760 The S/KEY One-Time Password System; RFC 1938 A One-Time Password System; RFC 2289 A One-Time Password System; Haller, N., The S/Key(tm) One-Time Password System, Proceedings of the Symposium on Network & Distributed Systems Security, Internet Society, San Diego, February 1994.)
- DASS (Distributed Authentication Security Service) (RFC 1507 DASS Distributed Authentication Security Service)

2.10 Key Escrow/Recovery

In Key Escrow/recovery -system, there is a third party, which have possibility and means to decrypt encrypted transfer. There is one or more trusted party, which are storing secure keys or recovery keys. Recovery keys provide means to find out secure keys used to encryption or decryption of data. The proposal of these kinds of systems is made in United States. The main idea is that authorities can monitor secure transmission to discard criminal elements. The systems can be implemented in different levels. Applications with strong cryptosystems, which support key escrowing, do not have so critical export constraints by U.S. Government.

DRC (Data Recovery Component) consists of logigal components and hardware, which are needed to decrypt ciphertext using services of KEC (Key Escrow Component) ja DRF-fields (Data Recovery Field) of data.

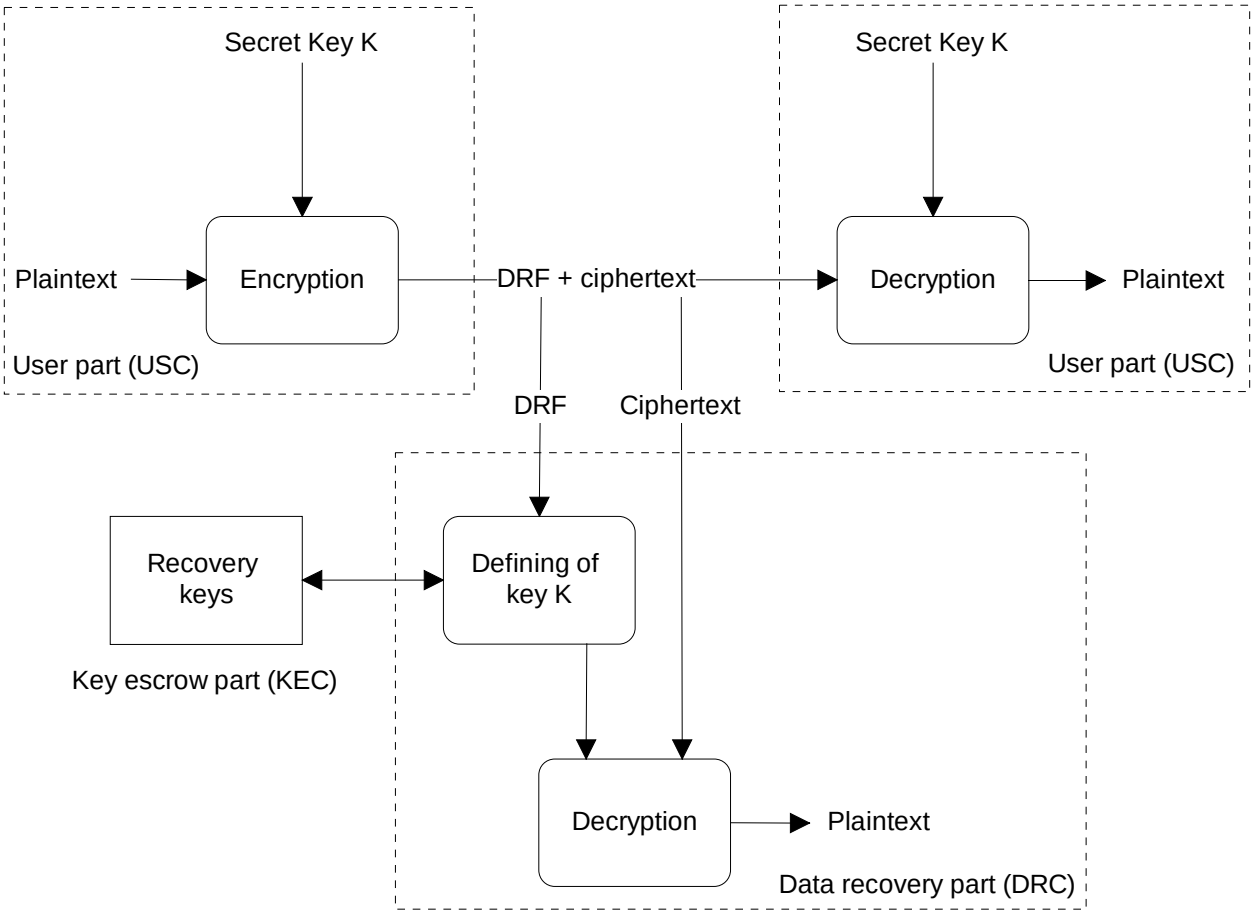

Figure 14. This figure represents main parts of Key Escrow-system.[2]

3. INTERNET LAYER PROTOCOLS

Internet layer protocols are transparent to application and to other levels in net. They can be used to build secured tunnels but they can't differentiate demands of separate processes. The security level depends completely on the strength of the implemented cryptographic algorithms, the strength of the key being used, and other security mechanisms in all of the participating systems.[23]

3.1 Internet Protocol Security Architecture

Internet Protocol Security Architecture (IPSEC) is developed by IETF's workgroup. IPSEC includes secure enhancement (IP Secure Protocol) for IP-protocol and Internet Key Management Protocol.[26]

3.1.1 IP Secure Protocol

IPSP is compatible for Ipv4 and for Ipv6. IPSP provides integrity and confidentiality for data flow and it identifies the source. All data that is sent to

specific address is encrypted with same encryption key. IPSP does not define the algorithm. [24]

IPSP is based to IP Authentication Header (AH) and to IP Encapsulating Security Payload (ESP). Authentication header provides provide integrity and authentication methods without confidentiality to IP datagrams. The IP Encapsulating Security Payload (ESP) provides methods for integrity, authentication, and confidentiality to IP datagrams. These methods do not provide protection against traffic analyses.

Both AH and ESP are based on security association (SA) and can be used separately. SA is an agreement between two or more parties that defines how they are providing the security. It is a combination of "Security Parameters Index" (SPI) and a Destination Address.

Security parameters are typically:

- Authentication algorithm
- Encryption algorithm
- Lifetime for parameters
- Source address for SA
- Confidence level for data

Combination of a given Security Parameter Index (SPI) and Destination Address identify security Association.

3.1.1.1 Authentication Header

Authentication Header provides authentication for and data integrity for IP packets. It can be decrypted using different kinds of algorithms for encrypting and depending on an algorithm it can offer also non-repudiation. Authentication Header increases the IP protocol processing costs in participating systems and communications latency. The Authentication Header works properly without changing the entire Internet infrastructure, because the authentication data is carried in its own payload.[25]

AH is calculated using the IP datagram information which do not change in transit. The structure of AH is following.

Next Header 8-bit	Length 8-bit	Reserved 8-bit
Security Parameter Index (SPI) 32-bit		
Authentication data N* 32bits		

Table 6. This table represents Authentication Header syntax.

- Next Header field defines the type of the next payload header after AH.
- Length, the length of authentication data in 32-bit words
- SPI defines SA for receiver side
- Authentication data, encrypted (for example with MD5 or SHA-1)

Structure of IP datagrams with AH are described in tables below:

IPv6 Header	Hop-by-Hop/Routing	Auth Header	Others	Upper Protocol

Table 7. This table represents IPv6 with AH structure.

IPv4 Header	Auth Header	Upper Protocol

Table 8. This table represents IPv4 with AH structure.

3.1.1.2 Encapsulating Security Payload

The IP Encapsulating Security Payload (ESP) provides integrity, confidentiality to IP datagrams, also authentication may be provided. It encapsulates an entire IP datagram (tunnel mode) or only the payload (transport mode) data inside the ESP. It encrypts most of the ESP contents, and appends new cleartext IP header to the now encrypted Encapsulating Security Payload. This cleartext IP header is used to carry the protected data through the Internet. Encapsulation is needed to provide confidentiality for the entire original datagram.[26]

Security Parameter Index (SPI), 32-bits

Initialisation Vector (IV) n* 32 bits			
Payload data			
Payload data			
Payload data			
Payload data	Padding		
Padding		Pad length, 8-bits	Payload Type, 8-bits

Table 9. This table represents structure of ESP.

- SPI, on receiver's side
- IV, Random number
- Payload, encrypted as SA defines
- Padding, so that payload and padding fields are equal to 6 modulo 8
- Pad length, length of padding
- Payload type, protocol of payload data

Only SPI and IV exist in the clear, the rest of fields are encrypted. The transport mode encrypts and encapsulates only upper layer protocol data. Overhead is small because there is no additional encrypted IP headers.

In tunnel-mode the original IP datagram is placed in the encrypted portion of the Encapsulating Security Payload and that entire ESP frame is placed within a datagram having unencrypted IP headers. It can be used between two trusted gateways to create secure tunnel for data flow. It prevents the possible traffic analyst to analyse what happens behind the gateways.

ESP needs extra processing for sending and receiving host due the protocol, encrypting and decrypting.

3.1.2 Internet key management protocol

Security association needs shared keys that are known only authorised parties of SA. In Internet there can be large number of parties involved and efficient key management protocol is needed.

4. TRANSPORT LAYER SECURITY PROTOCOLS

Transport layer security protocol tries provide secure communications over unsecured communication channels. The most famous secure transport layer protocols are secure shell (SSH) and secure sockets layer (SSL).

4.1 SSH

Secure shell (SSH) utilises generic transport security protocol. SSH provides the authentication of both end of connection. Integrity and confidentiality of data is protected. Data can also be compressed.

SSH consists from three major components.

- Transport layer protocol (SSH-TRANS)
- User authentication protocol (SSH-USERAUTH)
- Connection protocol (SSH-CONN)

First version of SSH was developed by Tatu Ylönen. SSH is available for Windows (3.x, 95, 98, NT), Unix, OS/2 and MacOS. The commercial product can bought from Datafellows Ltd, which has sold its F-Secure SSH to more than 100 countries. Since it was founded in 1988, its annual net growth of net sales has been over 80%. Turnover has reached $3.3 million, $7.6 million and $14.1million in the fiscal years 1995, 1996 and 1997 respectively.

4.1.1 Protocol

SSH protocol starts with authentication. The client sends authentication request to server. The server sends back the public host key and server key. The host key binds to connection to wanted server. The server key chances periodically to avoid decrypting of recorded traffic. [27]

In the second phase of protocol the client checks the host against its own database. Clients database includes manually distributed and preconfigured public host keys. The client can accept the key of unknown host and add it to client’s databases. Its possible to configure client, so that it only accepts predefined keys for higher secure reasons, which is recommended. Otherwise SSH does not protect messages from modification, there can be attacker in the middle of connection.

The client responds to server. The response message includes encrypted session key. The client has generated a 256-bit random key and chosen one of encryption algorithm from servers supported algorithms. After receiving the session key server decrypts the key and sends acknowledgement to client. The connection is ready for use.

Its possible to server to ask client to authenticate itself with corresponding way or asking password. If the authentication is asked, it is sent naturally over the encrypted SSH channel.

SSH requires preconfigured public keys for clients and hosts. The public can be in any format to which both clients and hosts can understand. It allows using strong cryptographic technologies even in Europe despite United States laws. SSH does not require trusted third party for keys, which is good because it can be without changes to existing network. But in future it more advantegous to use public keys throug trusted thir party.

4.1.2 SSH Transport Layer Protocol

SSH transport layer protocol provides secure host authentication, confidentiality and integrity of data. If transportation media is TCP/IP the port number is defined to be 22. Protocol supports compression data. [28]

None	REQUIRED	no compression
Zlib	OPTIONAL	GNU ZLIB (LZ77)

Table 10. This table represents the compression methods in SSH.

Encryption method is negotiated during key exchange. The encryption method can be different to each direction. The following ciphers are currently defined:

3des-cbc	REQUIRED	three-key 3DES in CBC mode
Blowfish-cbc	RECOMMENDED	Blowfish in CBC mode
Arcfour	OPTIONAL	The ARCFOUR stream cipher
Idea-cbc	OPTIONAL	OPTIONAL

Cast128-cbc	OPTIONAL	CAST-128 in CBC mode
None	OPTIONAL	No encryption; NOT RECOMMENDED

Table 11. This table represents the ciphers in SSH.

Integrity of data is secured by using message authentication code (MAC). It is computed from a shared secret, packet sequence number, and the contents of the packet. The sequence number is not sent, but it's used in calculation for MAC so ensure that all messages are received. MAC is the last part of the SSH message and it is not encrypted.

The following MAC algorithms are currently defined:

Hmac-sha1	REQUIRED	HMAC-SHA1(length=20)
Hmac-sha-96	RECOMMENDED	First 96 bits of HMAC-SHA1 (length=12)
Hmac-md5	OPTIONAL	HMAC-MD5 (length=16)
Hmac-md5-96	OPTIONAL	First 96 bits of HMAC-MD5 (length=12)
None	OPTIONAL	No MAC; NOT RECOMMENDED

Table 12. This table represents the MAC algorithms in SSH.

4.1.3 SSH Authentication Protocol

SSH Authentication Protocol runs over SSH transport layer. It provides user authentication. The protocol should be executed over secure transport protocol as SSH-TRANS.[29]

First client tells to server its user name and asks service with service name the server responds to client with message that includes acceptable authentication methods to asked service. After these messages are changed using asked method and access is allowed or denied.

4.1.4 SSH Connection protocol

Connection protocol multiplexes the encrypted tunnel into several logical channels. It can be used in interactive login sessions, remote execution of commands, forwarded TCP/IP connections, and forwarded X11 connections. [30]

4.2 SSL

SSL (Secure Socket Layer) is the method proposed by Netscape Communications Corporation. It is used to encrypt transactions in higher-level protocols such as HTTP, NNTP and FTP. SSL is supported by several different browsers, including Netscape Navigator, and Microsoft Internet Explorer and many different servers, including ones from Netscape, Microsoft, IBM, Quarterdeck, OpenMarket and O'Reilly and Associates. SSL is nowadays the most supported security protocol in Internet. One big problem is that most of SSL programs have been developed in USA and Canada, that means that outside North America these products use weak keys for encrypting.[31]

The SSL protocol provides:

- server authentication
- encryption of data in transit
- optional client authentication

SSL works between TCP/IP and application. To enable use of SSL the server and client must know that they using it (SSL). It can be done by specific port numbers for every application (hhtps 443,sssmtt 465, snntp 563, sldap 636, and spop3 995); application itself negotiates it as part of protocol or use TCP option for negotiation. SSL includes SSL record protocol and SSL handshake protocol.

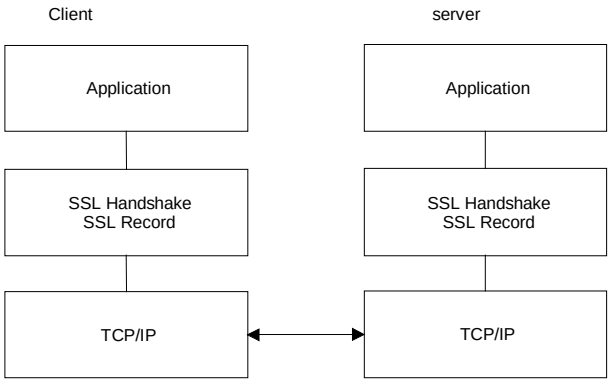

Figure 15. This figure represents that SSL works between TCP/IP and application.

4.3 SSL record protocol

SSL record protocol does the fragmentation, compression, authentication and encryption for data. It puts data to SSL records, which contain: the content type (higher level protocol), version of protocol (SSL), length, data payload and message authentication code. [31]

There can be several protocols above SSL record protocol as alert protocol, handshake protocol, and change cipher specification protocol.

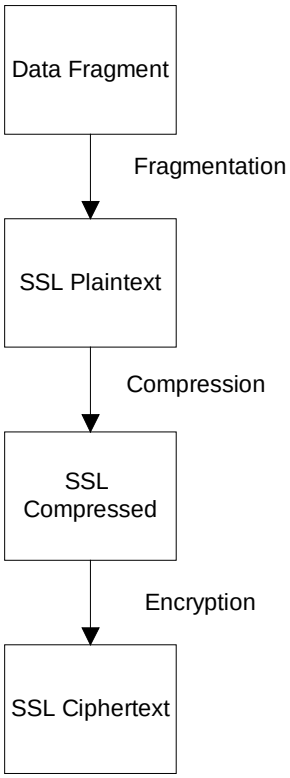

Figure 16. This figure represents SSL record protocol in shortly.

4.4 SSL Handshake protocol

The SSL Handshake Protocol produces cryptographic parameters of the session state. [31]

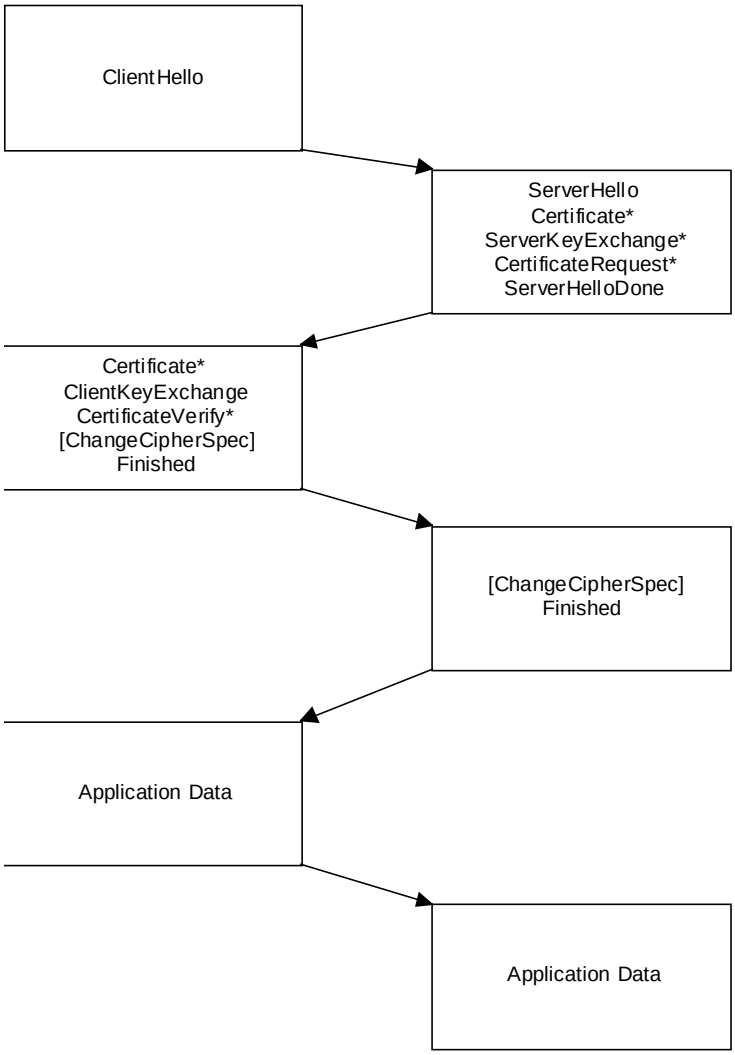

Figure 17. This figure represents SSL handshake protocol in shortly (* optional).

The client hello and server hello messages are used to establish security enhancement capabilities between client and server. The client hello and server hello messages establish the following attributes: Protocol Version, Session ID, Cipher Suite, and Compression Method.

First client sends hello message to server. Next step is that server send as many messages as needed to define its certificate (if it is authenticated) and client, a server key exchange and other parameters if

needed. At the end of this step server sends message to indicate that step is over. The server will then wait for a client response.

In next step client sends a change cipher spec message andthen immediately sends the finished message. The server will send its own change cipher spec message its finished message. Next step is that the client and server may begin to exchange application layer data.

5. APPENDIX 1

Kerberos based applications:
Kerberos FAQ, v1.9(http://www.nrl.navy.mil/CCS/people/kenh/kerberos-faq.html):
”A number of software vendors sell versions of Kerberos, or provide support for Kerberos:

* LatticeSoft Inc. has two Kerberos-based products: LattiXsite and LattiXcheck. They also provide technical support for a variety of Digital products (including DECathena).

  You can find out more information from <http://www.latticesoft.com>.

* Stonecast Inc. provides consulting, custom engineering, and custom products based on a wide variety of security technologies, but with special emphasis on Kerberos.

  You can find out more information from <http://www.stonecast.net>.

* CyberSafe sells and supports Kerberos 4 and Kerberos 5 with their TrustBroker product. In additional to normal Kerberos passwords, TrustBroker also supports the use of PKINIT authentication using public key certificates.

  You can find out more information from <http://www.cybersafe.com>.

* WRQ Inc. supports Kerberos on Win32 platforms with their Reflection Secure and Reflection Signature products. This includes a telnet client that does Kerberos 5 authentication and a graphical FTP client which supports Kerberos 5 (GSSAPI) authentication, data integrity, and privacy.

  You can find out more information from <http://www.wrq.com>.”

SSH Product prices from www.datafellows.com :”

SSH Prices F-Secure SSH Server

Server price: $495.
For each computer that the F-Secure SSH Server is installed a separate SSH Server license must be purchased.
Educational licenses are eligible for a %50 discount.

F-Secure SSH Tunnel&Terminal

Tunnel&Terminal price: starts at $99 per client.
Educational licenses are eligible for a %50 discount.

F-Secure FileCrypto

FileCrypto price: starting at $149 per client.
Educational licenses are eligible for a %50 discount.

F-Secure VPN+ Price List

VPN+ Client starts at $149.
VPN+ Server starts at $495.
VPN+ Gateway starts at $2495.
VPN+ Enterprise Gateway starts at $4990.

Educational licenses are eligible for a %50 discount. “

Should I use encryption with SSH , http://www.uni-karlsruhe.de/~ig25/ssh-faq/ “

“Today's CPUs are fast enough that performance losses (if any) only are noticable for local Ethernet speeds, or faster.

You might want to specify blowfish encryption instead of the default, IDEA, with -c blowfish, for faster operation.

Following are some measurements where the different encryption methods were applied between a P5/90 and a 486/100, both running Linux, for copying files with scp across a lightly loaded Ethernet.

The model chosen was t=a+x/b; a is the startup time in seconds, and b the sustainable transfer rate in kB/s. Also given are the 68.3%confidence intervals for the data, as determined by the Levenberg-Marquardt algorithm as implemented a pre-3.6 version of
gnuplot.

Encryption	a[s]	Da[s]	b[kB/s]	Db[kB/s]
None	2,37	0,37	386,1	5,8
RC4	1,96	0,27	318,2	2,9
TSS	2,33	0,37	298,5	3,5
DES	2,07	0,19	218,8	1,0
IDEA	2,25	0,45	169,6	1,3
3DES	1,92	0,11	118,2	0,2

Across a heavily loaded Ethernet, rc4 encryption together with compression may actually be faster than using rcp.

If you don't encrypt your sessions, you are vulnerable to all the attacks which are open on the "r" suite of utilities, and you might as well not use ssh.”
“

[1] W. Stallings. Network and Internet Security: principles and practice. Prentice-Hall, Inc., New Jersey, USA, 1995.

[2] Tiveke 2 -työryhmä: Tietoturva tietoverkoissa, 18.3.1998, http://www.telmo.fi/tiveke/, 11/98.

[3] Shannon C., Communication Theory of Secrecy System. Bell System Technical Journal 28(4), 656-715, 1949.

[4] Sorkin A., LUCIFER: a cryptographic algorithm, Cryptologia, 8(1), 22--35, 1984.

[5] Data Encryption Standard. National Bureau of Standards, FIPS PUB 46, Washington, DC, January 1977.

[6] RSA Laboratories FAQ 4.0, http://www.rsa.com/rsalabs/faq/, 11/98.

[7] W. Stallings. Network and Internet Security: principles and practice. Prentice-Hall, Inc., New Jersey, USA, 1995.

[8] Rfc 2313 PKCS #1: RSA Encryption Version 1.5, Network Working Group B. Kaliski, 1998.

[9] Rivest R., Shamir A., Adleman L., A Method for Obtaining Digital Signatures and Public-key Cryptosystems, CACM 21,2, 1978.

[10] RSA Laboratories FAQ 4.0, http://www.rsa.com/rsalabs/faq/, 11/98.

[11] RSA Laboratories. PKCS #7: Cryptographic Message Syntax, Version 1.5, November 1993.

[12] AES, http://www.nist.gov/aes, 11/98.

[13] Federal Information Processing Standards Publication (FIPS PUB) 81, DES Modes of Operation, 1980 December 2.

[14] Rfc 1423, Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers, Network Working Group D. Balenson, 1993.

[15] Lai X., Massey J., A proposal for a New Block Encryption Standard, Proceedings, Eurocrypt ’90, Springer-Verlag, 1990.

[16] National Bureau of Standards, FIPS PUB 180, Washington, DC, 1993.

[17] Rfc 1319, The MD2 Message-Digest Algorithm, Kaliski B., Network Working Group, April 1992.

[18] Rfc 1186, The MD4 Message-Digest Algorithm, Rivest R., Network Working Group, October 1990.

[19] Rfc 1320, The MD4 Message-Digest Algorithm, Rivest R., Network Working Group, April 1992.

[20] Rfc 1321, The MD5 Message-Digest Algorithm, Rivest R., Network Working Group, April 1992.

[21] Stallings W., Protect Your Privacy: A Guide for PGP Users, Prentice Hall, Inc., New Jersey, USA, 1995.

[22] Rfc 1424, Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services, Kaliski B., February 1993.

[23] Oppliger R., Internet and Intranet Security, Arhtec House, Inc., Norwood, Great Britain, 1998.

[24] Rfc 1827, IP Encapsulating Security Payload (ESP). R. Atkinson. August 1995.

[25] Rfc 1826, IP Authentication Header. R. Atkinson. August 1995.

[26] Rfc 1825, Security Architecture for the Internet Protocol. R. Atkinson. August 1995.

[27] Ylonen, T., Kivinen, T, and Saarinen, M., "SSH Protocol Architecture", Internet Draft, draft-secsh-architecture-00.txt.

[28] Ylonen, T., Kivinen, T, and Saarinen, M., "SSH Transport Layer Protocol", Internet Draft, draft-secsh-transport-02.txt.

[29] Ylonen, T., Kivinen, T, and Saarinen, M., "SSH Authentication Protocol", Internet Draft, draft-ietf-secsh-userauth-02.txt.

[30] Ylonen, T., Kivinen, T, and Saarinen, M., "SSH Connection Protocol", Internet Draft, draft-ietf-secsh-connect-02.txt

[31] Freier O, Karlton P, Kocher P,The SSL Protocol Version 3.0 draft-freier-ssl-version3-02.txt