

**Securityin
Networks**

IanJohnson

Room3Q77

Telephone:extension 83167

Email:lan.Johnson@uwe.ac.uk

**[http://www.cems.uwe.ac.uk/
~irjohnso](http://www.cems.uwe.ac.uk/~irjohnso)**

Assumptions:

You have a decent understanding of TCP/IP networking.

You can program! (We'll probably C and possibly some C++ and/or Java) and are comfortable with unix derived o.s.'s

This module is running for the first time and is therefore best viewed as a work in progress! Much may change from what I envisage now.

By the end of the module, that you've read Stallings!

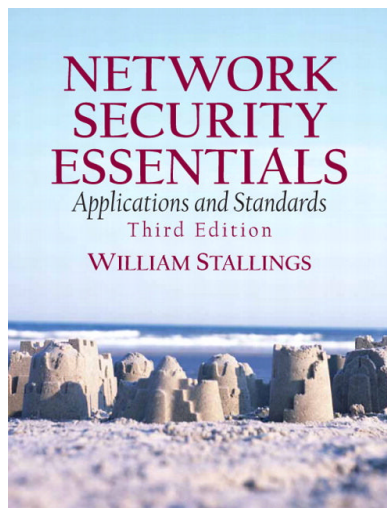
LabWo rk:

Ideally, I'd like to organise some discussion sessions, as well as some "Sys admin" style exercises and some programming. In particular I'd like to ensure you understand how to develop and deploy SSL.

We'll need to build some linux caddies to use to the secure network.

Books:

Recommended to Buy:



Stallings, William;
"Network Security Essentials",
3rd ed. Pearson/Prentice-Hall, 2007

Othergood sources:

Anderson is more general, but a must-read future classic, available online as PDF. Everyone has their own favourites, this list is not exhaustive!

Anderson, Ross; "Security Engineering", Wiley, 2001 (online)

www.its.bth.se/staff/hjo/ (Slides to go with Stallings Book)

www.wilyhacker.com (Cheswick & Bellovin) (1st ed. online)

Farmer, D & Venema, W, "Forensic Discovery" 2005. (online)

Singh, Simon; "The Code Book", 1999.

RSA Inc.; "RSA Security FAQ 4.1" (www.rsasecurity.com)

Ferguson, Niels & Schneier, Bruce; "Practical Cryptography", 2003

Mitnick, K.D. & Simon, W.L.; "The Art of Deception", 2002

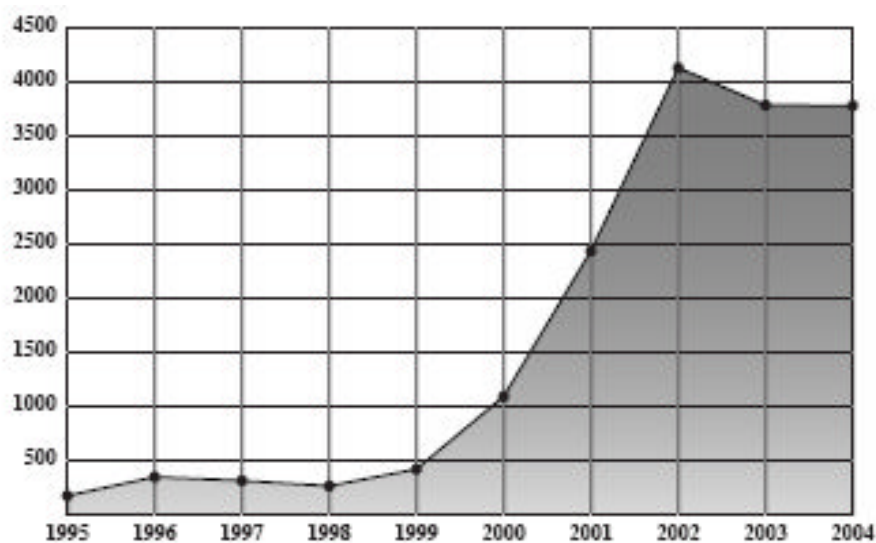
Hatch, B. & Lee, J., "Hacking Linux Exposed", 2nd Ed. , 2003.

I've quoted and used diagrams & figures from a few above. My key sources are Stallings 2nd & 3rd Editions.

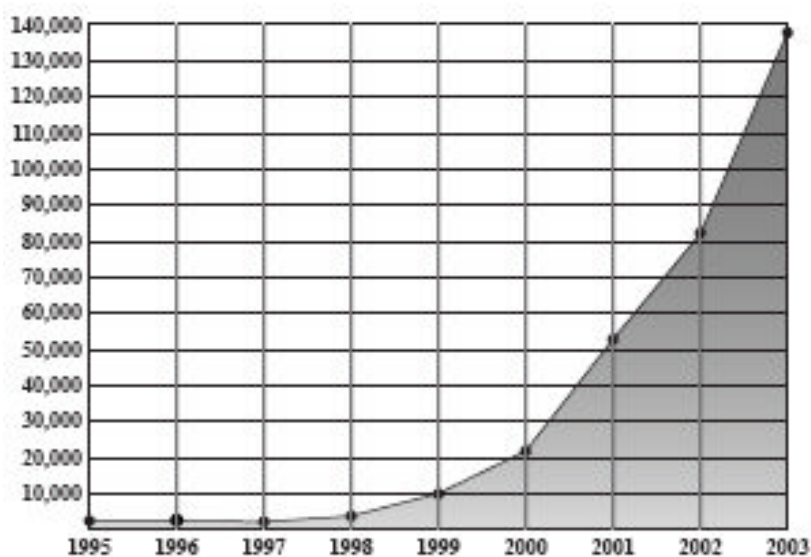
I Know I'm Paranoid, But am I paranoid enough? ☺

Why should we care about Network Security?

Lets look at some of Stallings (ch1) graphs:



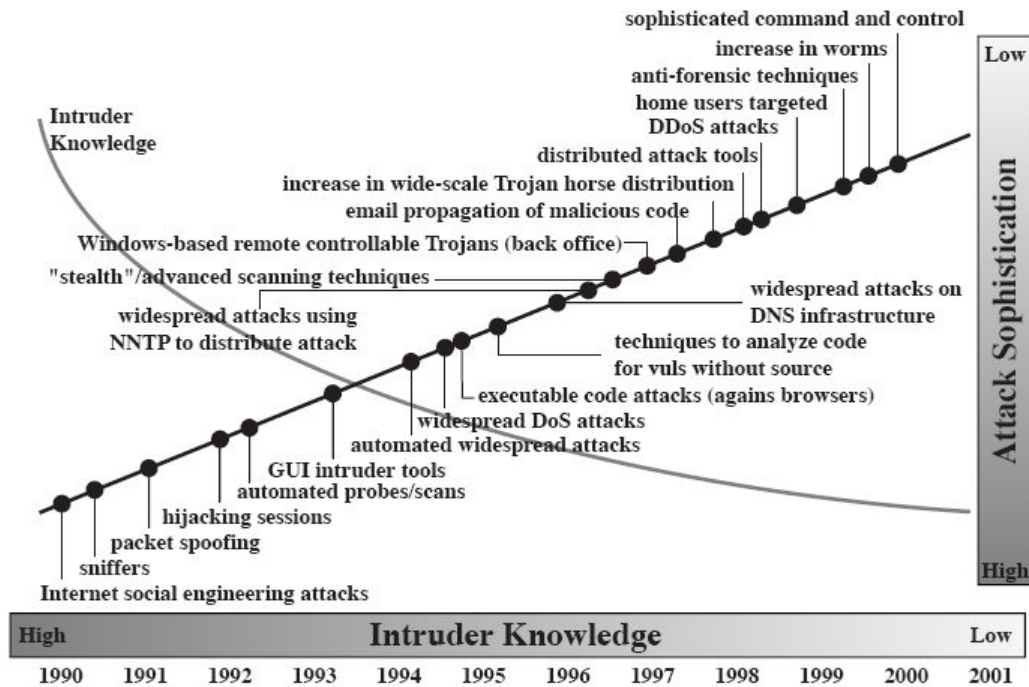
(a) Vulnerabilities reported



(b) Incidents reported

Note: Vulnerabilities plateau in 2003, Incidents continue to grow!

Why?



Source: CERT

What is Security?

“Security is keeping unauthorized entities from doing things you don’t want them to do.”

Bellovin

Security is a process, not an event!

Objective

What do we want to achieve in the abstract?

Policy

Organisation's guidelines to achieve Objective.

Mechanisms

Tools/Technical to implement policy

Assurance

Are we sure that:

- a) Our tools work
- b) They help us achieve our policy?

Response

What to do when things go wrong (should form part of the policy)

Education

Getting people to understand the need for, and use our mechanisms.

In short?

Who are you trying to protect **what** from and **why**?

Different enemies have different abilities & motivations:

- • Teenage hackers can't crack a modern cryptosystem
- Serious enemies can exploit the "three Bs": burglary, bribery, and blackmail
- You can't design a security system unless you know who the enemy is:
 - Local drug addict looking to fund a fix
 - Organised crime
 - Nation-state

SWOTanalysis

Beloved of business schools, has practical value here.

Start with objective.

Brainstorm current strengths and weaknesses with respect to that objective. (internal)

Brainstorm opportunities & threats (external).

Analyse and determine a plan of action.

Securityrequirements engineering

Design systems from the ground up with our security needs uppermost.
Fail-safe design.

“Building systems to remain dependable in the face of malice, error, or mischance”

Anderson

Read Chapter 1 of Anderson!

Penetrationtesting

“White Hat” hacking! One way of assuring the integrity of chosen security mechanisms.

SecurityGoals

Confidentiality (privacy)

Authentication (who created or sent the data)

Integrity (has not been altered)

Non-repudiation (the order is final)

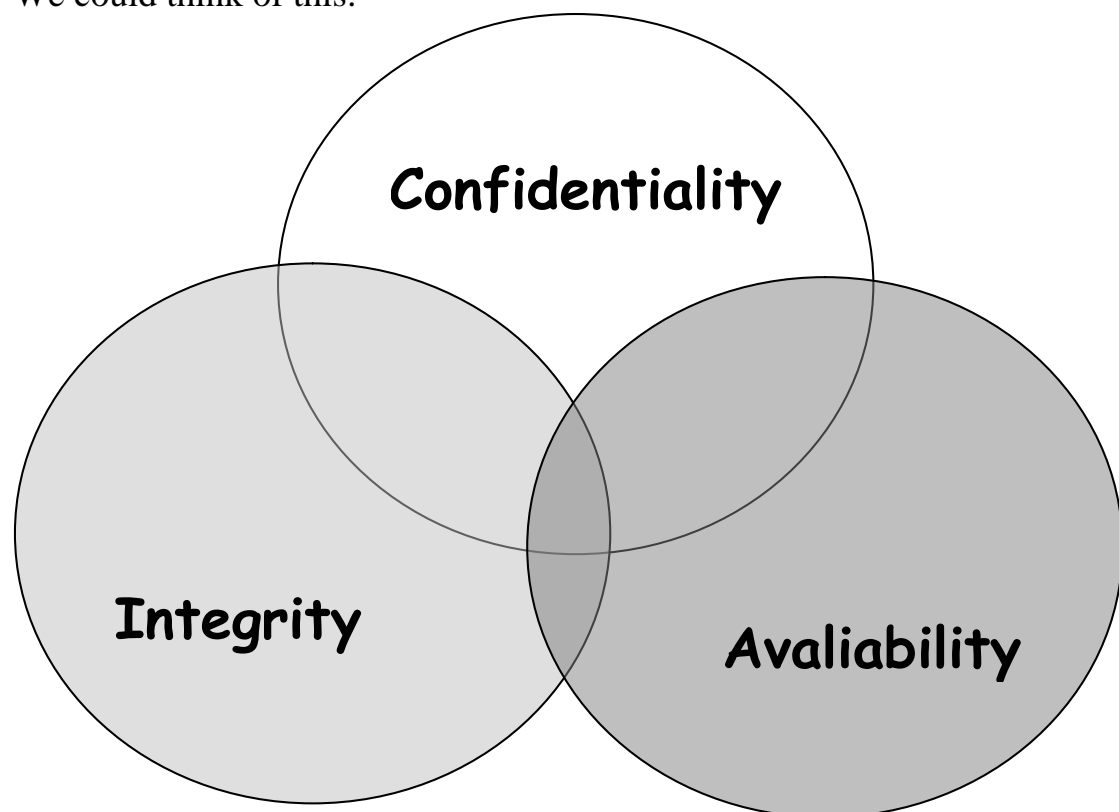
Access control (prevent misuse of resources)

Availability (permanence, non-erasure)

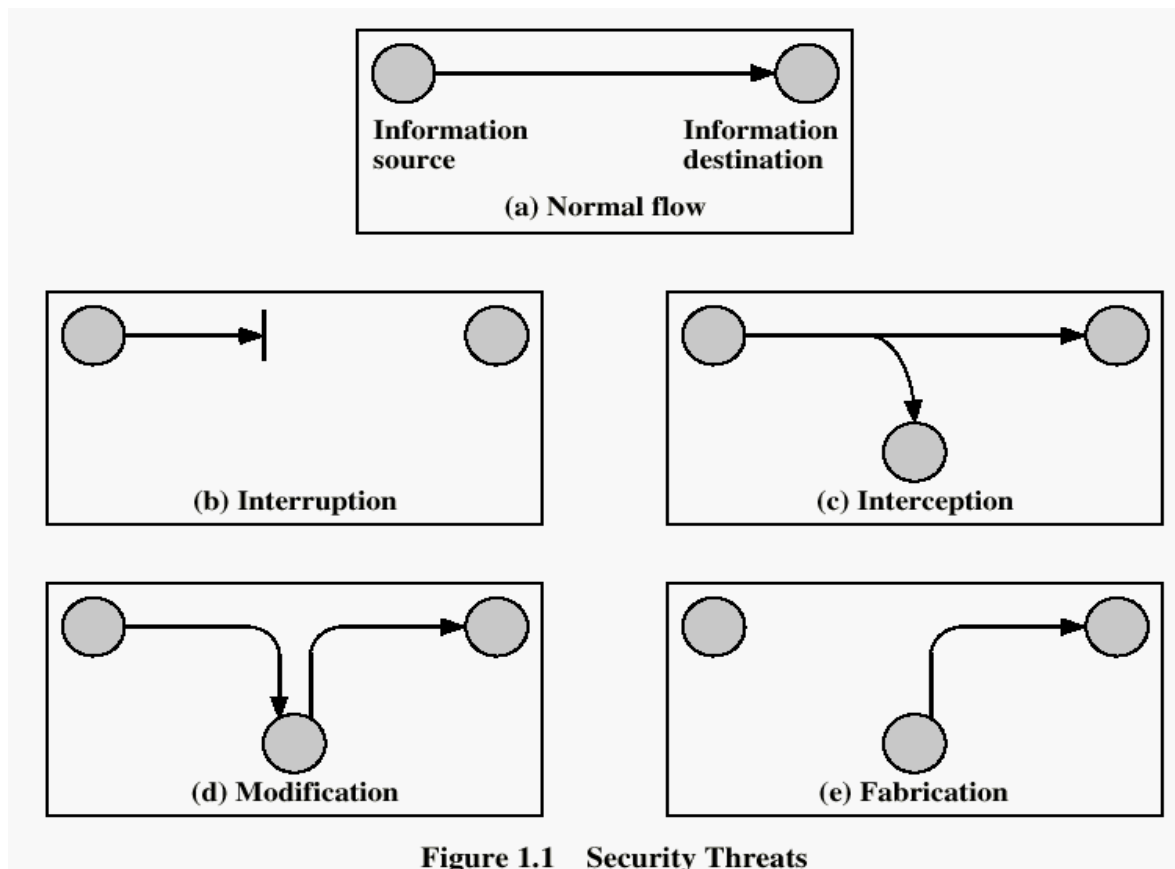
Denial of Service Attacks

Virus that deletes files

We could think of this:



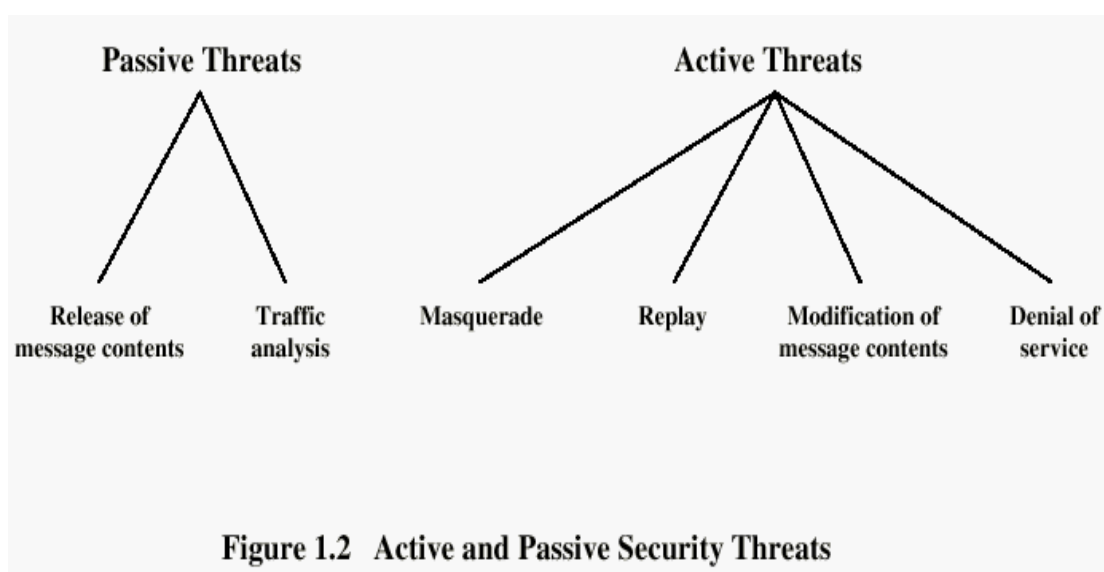
Security Threats :



Security Attacks

- Interruption: This is an attack on availability
- Interception: This is an attack on confidentiality
- Modification: This is an attack on integrity
- Fabrication: This is an attack on authenticity

We should also distinguish :



So how do we defend against (some) of these?

Table 1.4 Relationship Between Security Services and Mechanisms

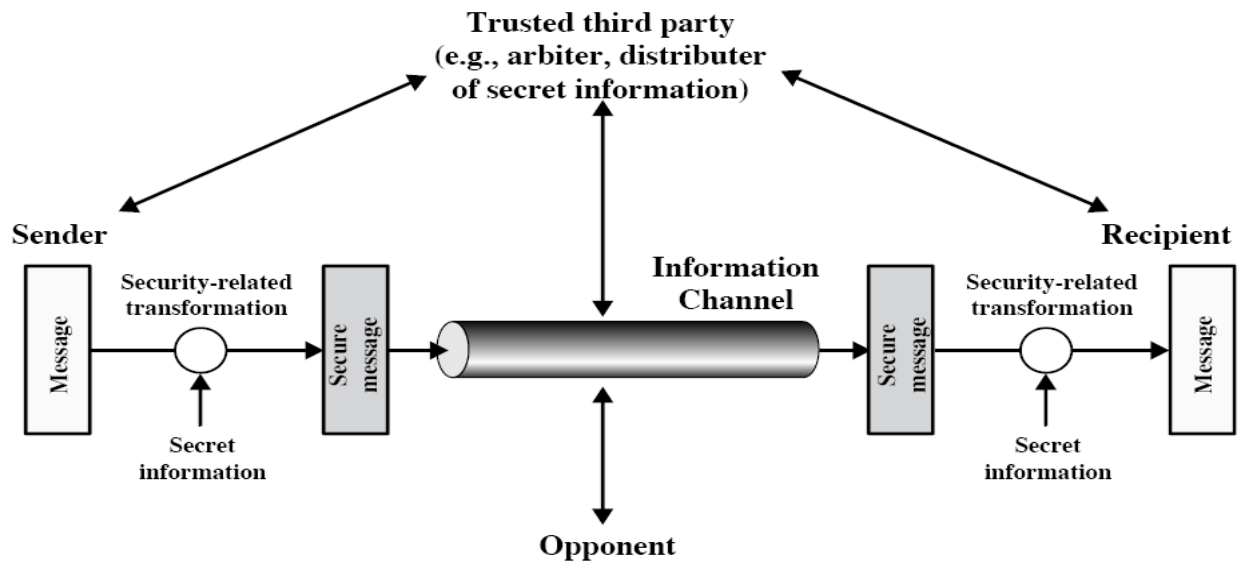
| Service | Mechanism | | | | | | | |
|------------------------------|--------------|-------------------|----------------|----------------|-------------------------|-----------------|-----------------|--------------|
| | Encipherment | Digital signature | Access control | Data integrity | Authentication exchange | Traffic padding | Routing control | Notarization |
| Peer entity authentication | Y | Y | | | Y | | | |
| Data origin authentication | Y | Y | | | | | | |
| Access control | | | Y | | | | | |
| Confidentiality | Y | | | | | | Y | |
| Traffic flow confidentiality | Y | | | | | Y | Y | |
| Data integrity | Y | Y | | Y | | | | |
| Non-repudiation | | Y | | Y | | | | Y |
| Availability | | | | Y | Y | | | |

ISO/OSI X.800 Security Mechanisms

| SPECIFIC SECURITY MECHANISMS | PERVASIVE SECURITY MECHANISMS |
|---|---|
| <p>May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.</p> | <p>Mechanisms that are not specific to any particular OSI security service or protocol layer.</p> |
| <p>Encipherment The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.</p> | <p>Trusted Functionality That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).</p> |
| <p>Digital Signature Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).</p> | <p>Security Label The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.</p> |
| <p>Access Control A variety of mechanisms that enforce access rights to resources.</p> | <p>Event Detection Detection of security-relevant events.</p> |
| <p>Data Integrity A variety of mechanisms used to assure the integrity of a data unit or stream of data units.</p> | <p>Security Audit Trail Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.</p> |
| <p>Authentication Exchange A mechanism intended to ensure the identity of an entity by means of information exchange.</p> | <p>Security Recovery Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.</p> |
| <p>Traffic Padding The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.</p> | |
| <p>Routing Control Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.</p> | |
| <p>Notarization The use of a trusted third party to assure certain properties of a data exchange.</p> | |

A model for network security

From Stallings:



But do not forget other issues:

Host security

Social Engineering

Physical Security

Reliability & Fault tolerance

....amongst others!

Cryptography101

“Cryptography is nothing more than a mathematical framework for discussing the implications of various paranoid delusions”

Don Alvarez

Traditionally, Crypto has been a battle between cryptanalysis and cryptographers. Currently, the cryptographers have the upper hand, and are likely to remain in this position for some significant time.

Cryptanalysts attack the algorithm (as per Stallings table below). Others take different routes!

Table 2.1 Types of Attacks on Encrypted Messages

| Type of Attack | Known to Cryptanalyst |
|-------------------|---|
| Ciphertext only | <ul style="list-style-type: none">•Encryption algorithm•Ciphertext to be decoded |
| Known plaintext | <ul style="list-style-type: none">•Encryption algorithm•Ciphertext to be decoded•One or more plaintext-ciphertext pairs formed with the secret key |
| Chosen plaintext | <ul style="list-style-type: none">•Encryption algorithm•Ciphertext to be decoded•Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key |
| Chosen ciphertext | <ul style="list-style-type: none">•Encryption algorithm•Ciphertext to be decoded•Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key |
| Chosen text | <ul style="list-style-type: none">•Encryption algorithm•Ciphertext to be decoded•Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key•Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key |

We don't need to be able to design or analyse algorithms!

Cryptosystems generally fail because of incorrect USE, not design.

We simply use well understood & tested algorithms generally as part of standard protocols.

Kerckhoffs Principle (1883):

The security of a cryptosystem should depend on the key alone, not the algorithm.

Kerckhoffin detail:

1. The system must be practically, if not mathematically, indecipherable;
2. It must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience;
3. Its key must be communicable and retainable without the help of written notes, and changeable or modifiable at the will of the correspondents;
4. It must be applicable to telegraphic correspondence;
5. It must be portable, and its usage and function must not require the concurrence of several people;
6. Finally, it is necessary, given the circumstances that command its application, that the system be easy to use, requiring neither mental strain nor the knowledge of a long series of rules to observe.

Reference: Auguste Kerckhoffs, *La cryptographie militaire*, Journal des sciences militaires, vol. IX, pp. 5-83, Jan. 1883, pp. 161-191, Feb. 1883.

The big problem!

Communicating a shared secret!

This is why publickey cryptosystems generated an explosion!

Previously, you needed a seriously strong infrastructure for key distribution.

Feistel Cipher



From Wikipedia:

The basic operation is as follows:

Split the plaintext block into two equal pieces, (L_0, R_0)

For each round $i = 1, 2, \dots, n$, compute:

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

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_{i-1})$$

where f is the round function and K_i is the sub-key.

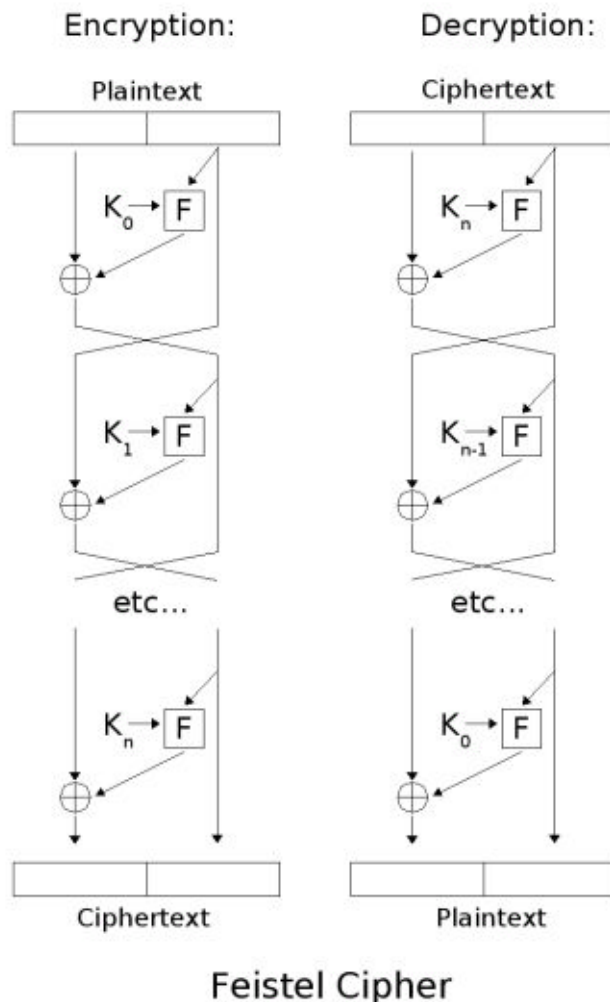
Then the ciphertext is (L_n, R_n) .

Decryption is accomplished via

$$\begin{aligned} R_{i-1} &= L_i \\ L_{i-1} &= R_i \oplus f(L_i, K_i) \end{aligned}$$

One advantage of this model is that the round function f used does not have to be invertible, and can be very complex.

This diagram illustrates both encryption and decryption. Note the reversal of the subkey order for decryption; this is the only difference between encryption and decryption:



Key Parameters:

- **Block size:** larger block sizes mean greater security
- **Key Size:** larger key size means greater security
- **Number of rounds:** multiple rounds offer increasing security
- **Subkey generation algorithm:** greater complexity will lead to greater difficulty of cryptanalysis.

Traditional symmetric block ciphers (e.g. DES (56 bit key, 64 bit block) or IDEA) are effective and relatively computationally cheap.

From Stallings, "Network Security (old edition)"

| Key Size (bits) | Number of Alternative Keys | Time required at 106 Decryption/ μ s |
|-----------------|--------------------------------|--|
| 32 | $2^{32} = 4.3 \times 10^9$ | 2.15 milliseconds |
| 56 | $2^{56} = 7.2 \times 10^{16}$ | 10 hours |
| 128 | $2^{128} = 3.4 \times 10^{38}$ | 5.4×10^{18} years |
| 168 | $2^{168} = 3.7 \times 10^{50}$ | 5.9×10^{30} years |

But:

- Small key = weak (56 bit EFF ~3 days)
- We need to block our data
- We need to communicate the key!

Most modern cryptosystems (SSL, PGP/GPG) use an asymmetric cipher (e.g. RSA) to communicate or negotiate a symmetric key. This is often used for a single session, in which case it is usually referred to as a session key.

SymmetricStream Ciphers

Have the advantage of no need to block and pad

Fit nicely with the unix socket/file model

Hard to do well!

Discovered by Vernam in 1917.

The infamous one time pad (OTP)!

The basic idea is very simple:

Key seeds pseudo-random number generator

Output of pseudo-random generator XOR'ed with plain text.

Quality really depends on the RNG.

Real example RC4

Too weak to recommend for future use!

Note: WEP problems are due to key-generation component as implemented being predictable NOT weaknesses in the algorithm!

RandomNumbers

“Real” – generated from natural phenomena

Basis for a One Time Pad

“Pseudo” – generated by an algorithm

For a given seed the sequence will always be identical.

An example (inadequate for crypto!)

Park & Miller suggest that the simple multiplicative congruential generator:

$$I_{j+1} = a I_j \pmod{m}$$

can be as good as the linear congruential generators providing that the values of the multiplier (a) and modulus (m) are chosen extremely carefully. In this respect, Park & Miller recommend $a = 16807$ (7^5), and $m = 2147483647$ ($2^{31}-1$).

- Period will be approximately m .
- But usage errors again!

Look at the code on the next slide:

```

main()
{
    int i,j;

    srand(113);
    for (i=0;i< 10; i++)
    {
        for (j=0; j < 10; j++)
            printf("%2d ",rand()%2);
        printf("\n");
    }
}

```

Whats the Output look like?

```

0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1

```

Asymmetric(often a.k.a.Public key) algorithms

The basis (or key ☺) of most security today!

Without PK – No SSL, No SSH, Digital signatures, email encryption, etc. etc.

Computationally **VERY** expensive

Can always in theory be broken eventually.

Public key encryption relies on the difficulty of factoring very large numbers, and testing numbers for primeness. (Or less commonly equally hard problems such as locating points on an elliptical curve).

As a result, given sufficient time, any public key algorithm can be broken, but the time required depends on key length:

But all problems that interest public key cryptographers are either P or NP.

P = can be solved in polynomial time, e.g. product of 2 numbers of n bits can be solved in at most n^2 bit operations

All problems that are P are also NP, but (it is believed) not the reverse! A problem is NP if a solution can be checked in polynomial time. Factoring a number is NP.

Diffie & Hellman published the first description of Public Key in 1976.

It had been independently discovered in the late 1960's by Ellis, Cox & Williamson at GCHQ. Some reports were declassified in 1997.

All PK algorithms are based on (presumed) Trapdoor One Way Functions. One way functions are functions that are easy to compute one way and hard the other. Trapdoor one way functions have a trapdoor that (if you know it) makes the reverse direction easy too.

Most modern cryptosystems use factoring as the “hard” problem, although Elliptic-curve algorithms hold some promise, since currently no non-exponential time solution is known.

Size of problem is directly proportional to key length.

RSA Challenges have now demonstrated the 512 bit is weak.

Increasing the key length by 1 bit doubles the effort required to brute force the key. Doubling key length therefore increases effort required by something of the order of 2^{512} .

Doubling the key length makes decrypt operations around four times slower and encrypt operations around eight times slower.

From RSA Security FAQ (www.rsasecurity.com)

| | Block Cipher | RSA | Elliptic Curve | DSA |
|-----------------------------|--------------|--------------|----------------|--------------------------|
| Export Grade | 56 | 512 | 112 | 512 / 112 |
| Traditional recommendations | 80 112 | 1024 2048 | 160 224 | 1024 / 160 2048 / 224 |
| Lenstra/Verheul 2000 | 70 | 952 | 132 | 952 / 125 |
| 2010 | 78 | 1369 | 146 / 160 | 1369 / 138 |

Table 2. Minimal key lengths in bits for different grades.

Some key points:

Public key compromises are often very serious!

Digital signatures may be required to persist for a very long time!

What is secure today, may not be tomorrow, will not be in 100 years.

Bit length is not a good indicator of strength on its own. All things being equal, it may well be.

Paranoia is important!

Worked Example

- This is ~10 bit RSA. We know 512 bit is insecure.
- To brute force we need to test prime factors of N
- Security is therefore proportional to the difficulty of doing so!

Lets have a look at RSA:

Select 2 large (>1024 bit) prime numbers P & Q p = 17, q = 11

Compute product (N) 187

Select E where:

$$E > 1,$$

$$E < PQ$$

E and (P-1)(Q-1) have no common prime factors

We'll use E = 7

Ciphertext (C) = Message (M)^E (mod N)

We want to transmit ASCII X (88 decimal)

N & E are the public key

$$C = 88^7 \pmod{187}$$

$$C = 11$$

When this is received, it cannot be easily decrypted without knowledge of P & Q. Using these values we can compute the private key (D).

$$E * D = 1 \pmod{(p-1) * (q-1)}$$

$$7 * D = 1 \pmod{16 * 10}$$

$$D = 23$$

To decrypt we compute:

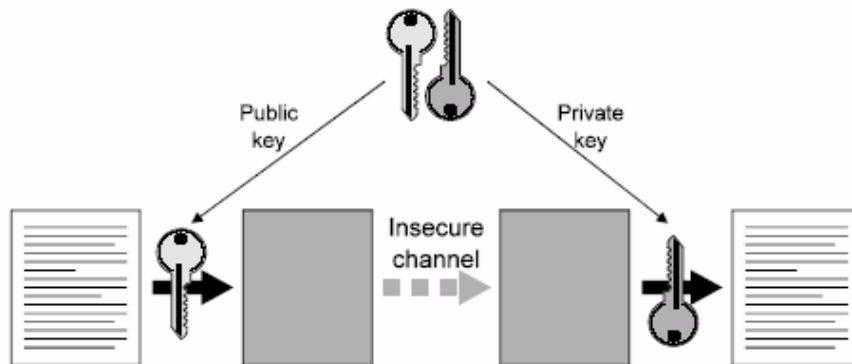
$$M = C^D \pmod{187}$$

$$M = 11^{23} \pmod{187}$$

$$M = 88$$

Public-key Encryption

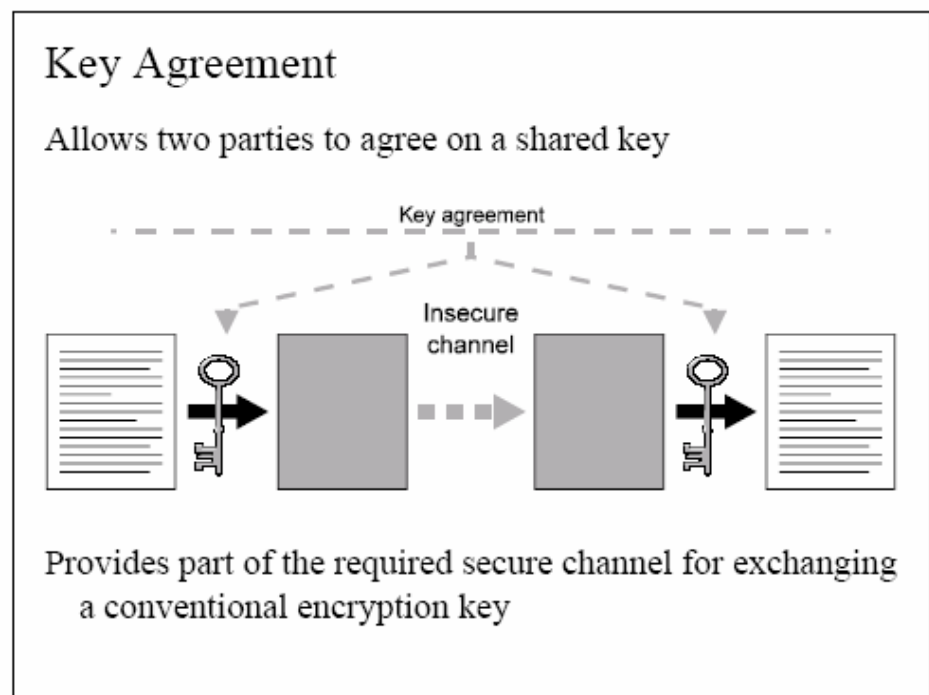
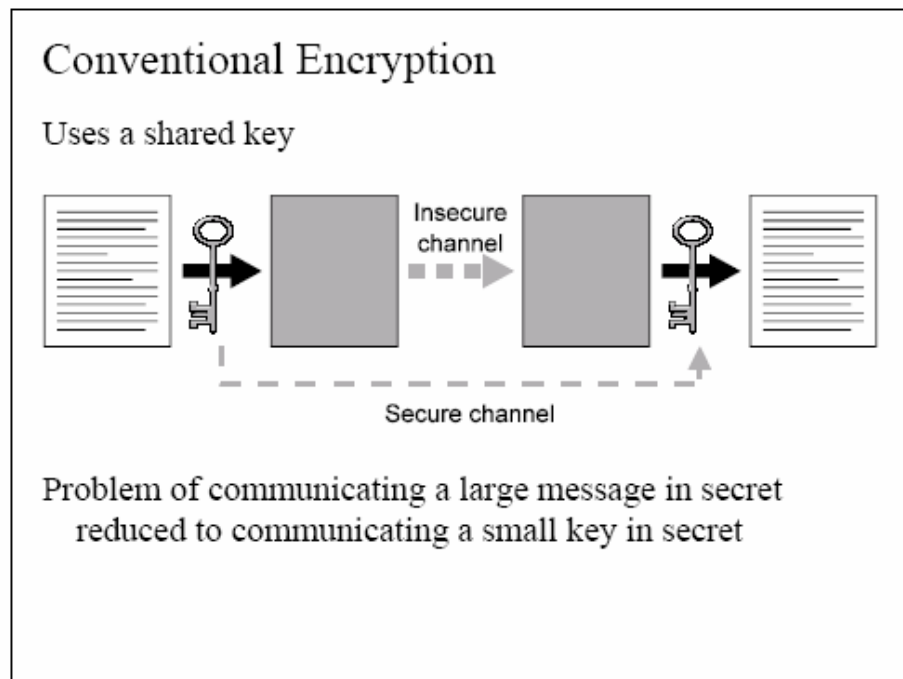
Uses matched public/private key pairs



Anyone can encrypt with the public key, only one person can decrypt with the private key

Symmetric algorithms

Key needs to be communicated securely



Message Digests

A hash function (MD5 is a well known example)

Generates a difficult to forge “fingerprint” for a file

1 in 18,446,744,073,709,551,616

Most (current) hash algorithms are block based and iterative.

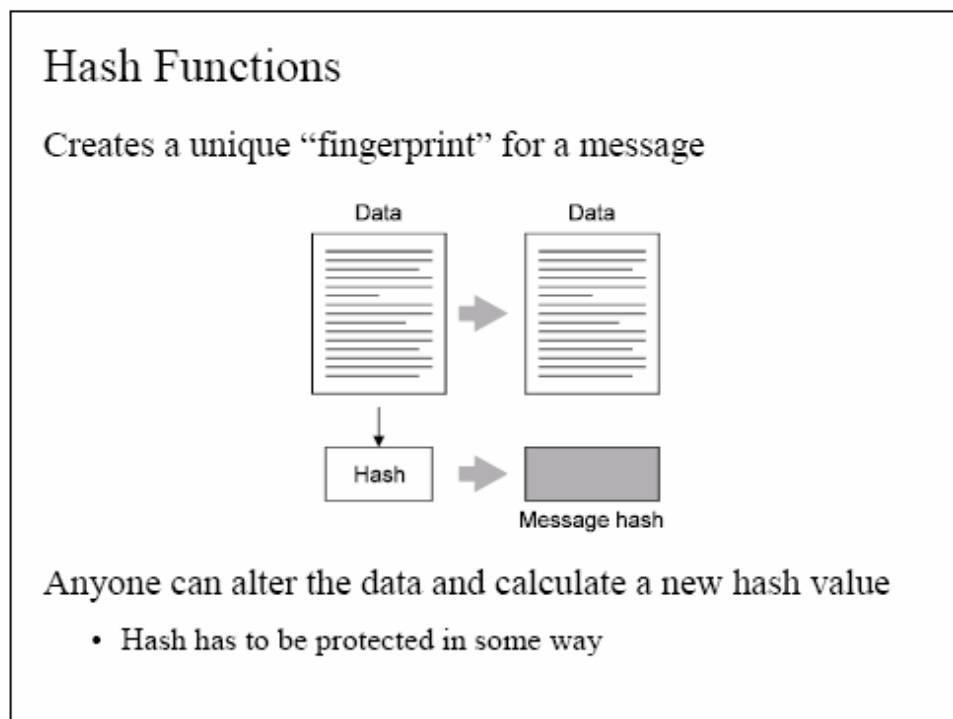
Typical block size is 512 bit.

Start with a fixed hash value H_0 .

Apply function to block k_i , to generate H_i

Repeat until last block (which is padded out to block size)

Result is hash value.



Currently no strong hash algorithm exists!

Note: Care needs to be taken when discussing hash attacks, all hash algorithms collide, we need usefully predictable collisions for an attack.

The Birthday Attack

The Birthday Paradox:

“How many people do you need in a room to be confident two share the same birthday (excluding year)?”

probability of 50% = 23
 >99% = 60

Hash algorithms must have a probability of collision.

Taylor series expansion of the exponential function approximates:

$$n(p) \approx \sqrt{2 \cdot H \cdot \ln \left(\frac{1}{1-p} \right)},$$

H = number of outputs
p = probability required.

Solving for the birthday paradox at 50%

$$n(0.5) = 1.1774 \times \sqrt{365} = 22.494$$

MD5 = $\sim 1.8 \times 10^{19}$ unique values, we'd expect 5.1×10^9 attempts to generate a collision.

The above assumes hashes are equally probable, a best case!

The goals

- Privacy
- Authentication

In reality, any system will always have more requirements

Summary

OTPs usually are not ☺

Expertise requires research on security engineering. Let others do the crypto.

Engineering secure systems is hard. In effect we are trying to prove the non-existence of an attack under all conditions. Proving security in these situations is logically impossible.

PublicKey Properties

If we know (E,N) the public key, we can encrypt so that decryption requires (D,N).

If we know (D,N) the private key, we can encrypt so that decryption requires (E,N), we can use this with a message digest as a digital signature. This allows us to authenticate.

Authentication in this case means matching a public and private key, to match to identity is harder and requires some sort of PKI infrastructure.

- What do we mean by identity?
- How does it relate to role?
- Authorisation, Authentication and Identity
- X.509 certificates common (problem with non-unique namespace)
- PGP Web of trust.
- Formal vs Informal, ad-hoc vs Hierarchical.

Other issues?

- Revocation (note: RIPA)
- Non-repudiation

Cracking Encryption

In 1998 under the direction of John Gilmore of the EFF, a team spent \$220,000 and built a machine that can go through the entire 56-bit DES key space in an average of 4.5 days. On July 17, 1998, they announced they had cracked a 56-bit key in 56 hours. The computer, called Deep Crack, uses 27 boards each containing 64 chips, and is capable of testing 90 billion keys a second.

More recently, in early 1999, Distributed.Net used the DES Cracker and a worldwide network of nearly 100,000 PCs to win the RSA DES Challenge III in a record breaking 22 hours and 15 minutes. The DES Cracker and PCs combined were testing 245 billion keys per second when the correct key was found. In addition, it has been shown that for a cost of one million dollars a dedicated hardware device can be built that can search all possible DES keys in about 3.5 hours. This just serves to illustrate that any organisation with moderate resources can break through DES with very little effort these days.

Where do we use encryption?

Encrypted Email (S/MIME, GPG, PGP, etc.)
key security?

Digital signatures (GPG, PGP)

We also use a digital signature to protect our tripwire fingerprint.

SSH – host/client verification, encrypted channel

SSL – encrypted channel, certificate based identification.

Standard use for secure HTTP

Can be used as a wrapper for any socket based service.

Supported by the openssl library.

Kerberos – encrypted network traffic

The Secure Sockets Layer (SSL)

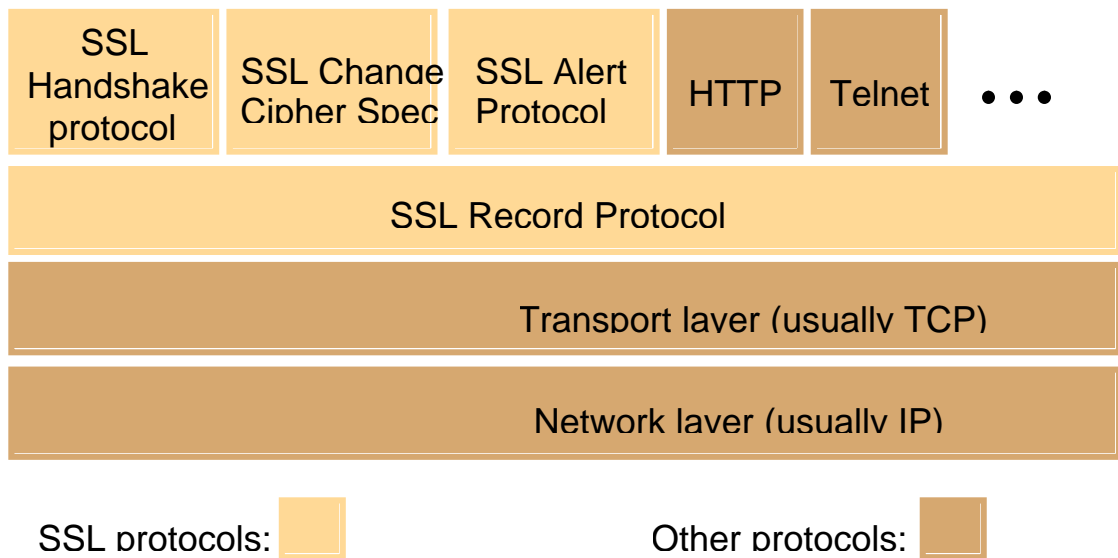
SSL is also known as TLS

Supported on most architectures

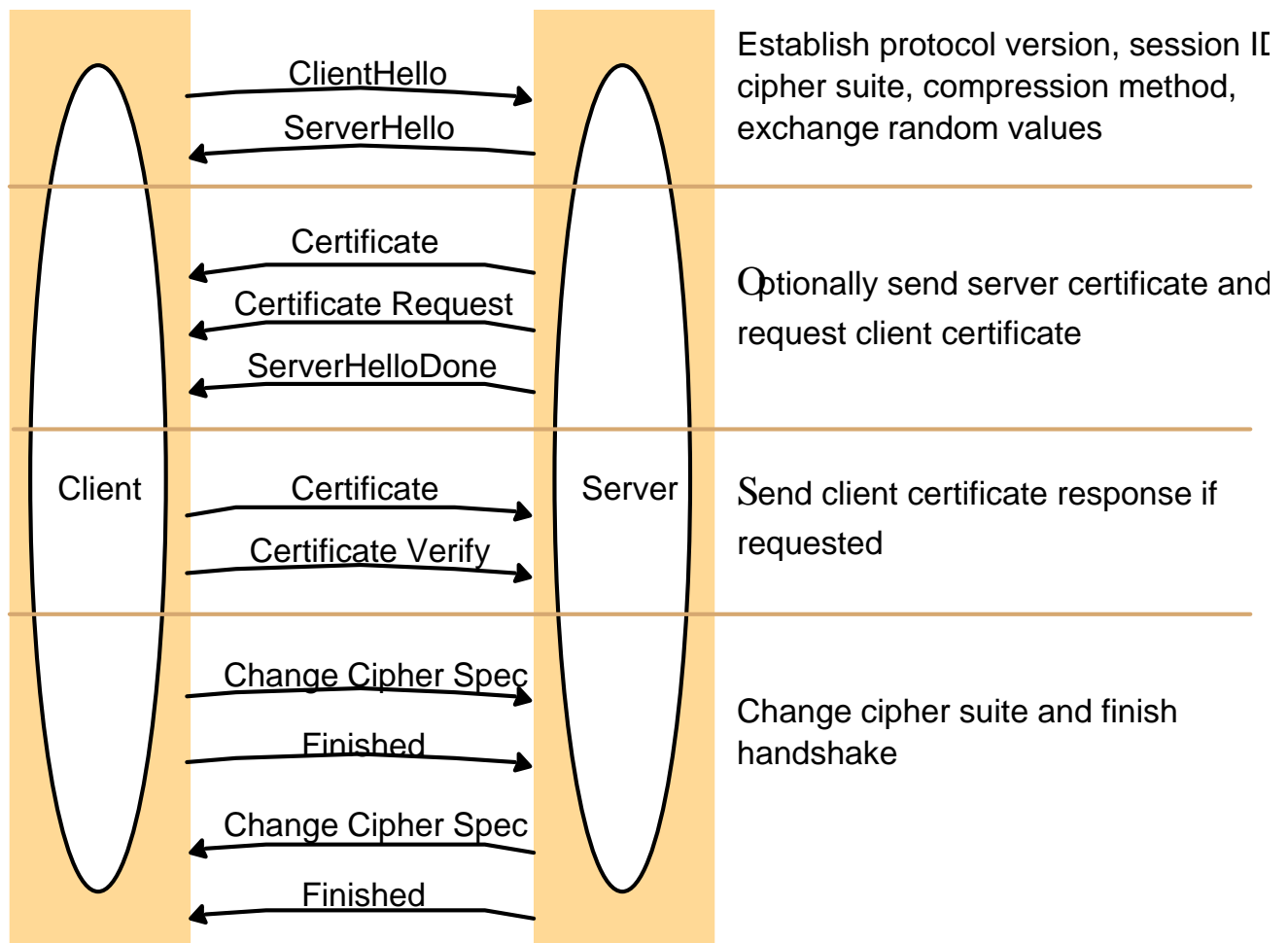
Supported by an open source library (openssl)

Two part tutorial on the module web page

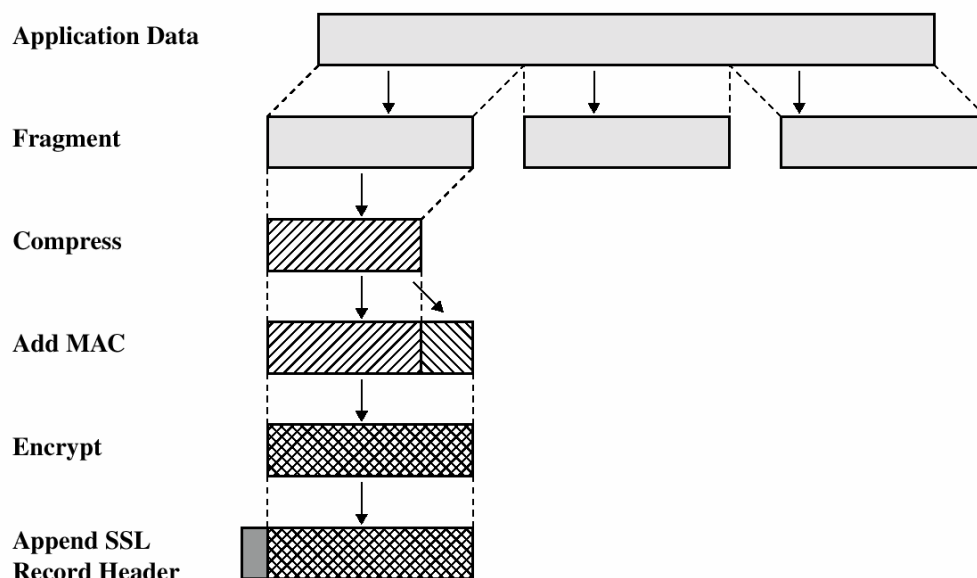
From CDK: 7.17, SSL protocol stack



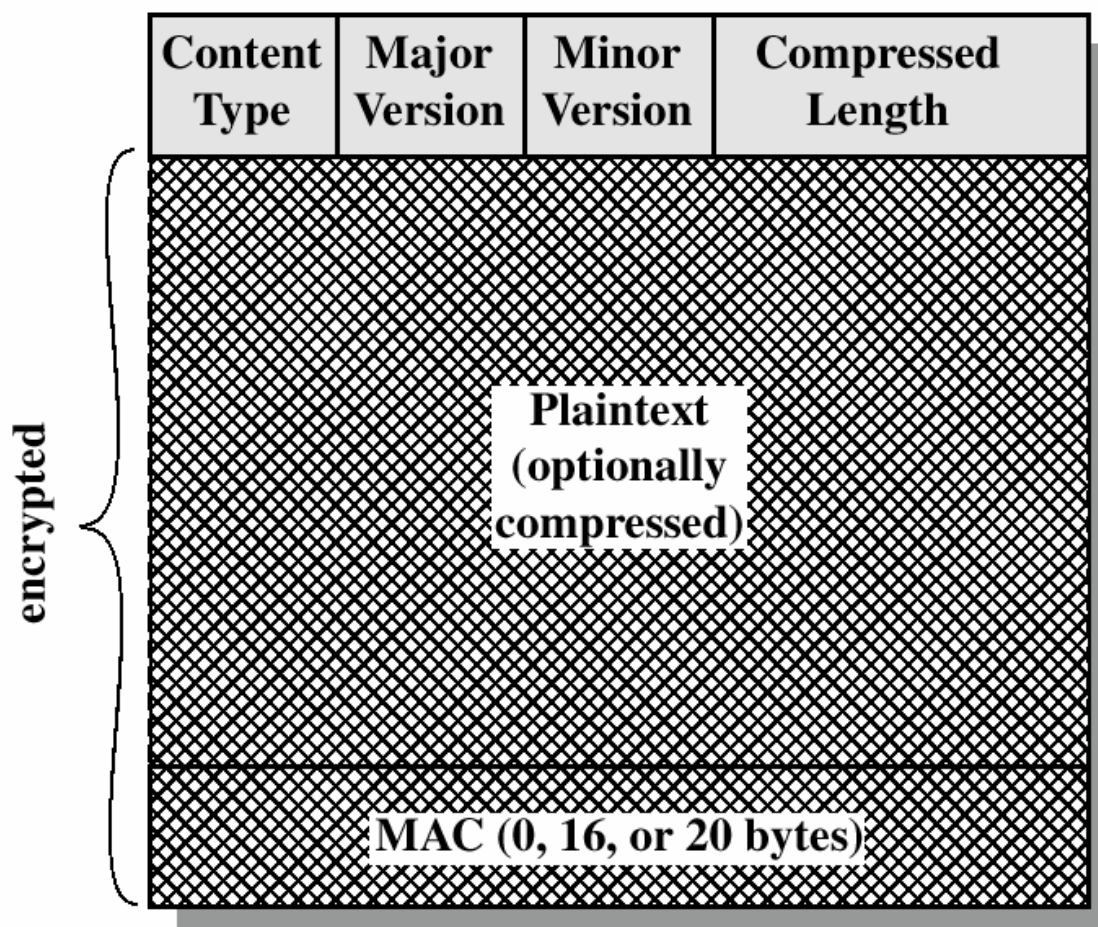
SSL Handshake (from CDK fig. 7.18)



SSLRecord Protocol



SSLRecord Format



SSL Handshake configuration options (CDK fig 7.19)

| <i>Component</i> | <i>Description</i> | <i>Example</i> |
|--------------------------|---|----------------------------------|
| Key exchange method | the method to be used for exchange of a session key | RSA with public-key certificates |
| Cipher for data transfer | the block or stream cipher to be used for data | IDEA |
| Message digest function | for creating message authentication codes | SHA |

X.509certificates

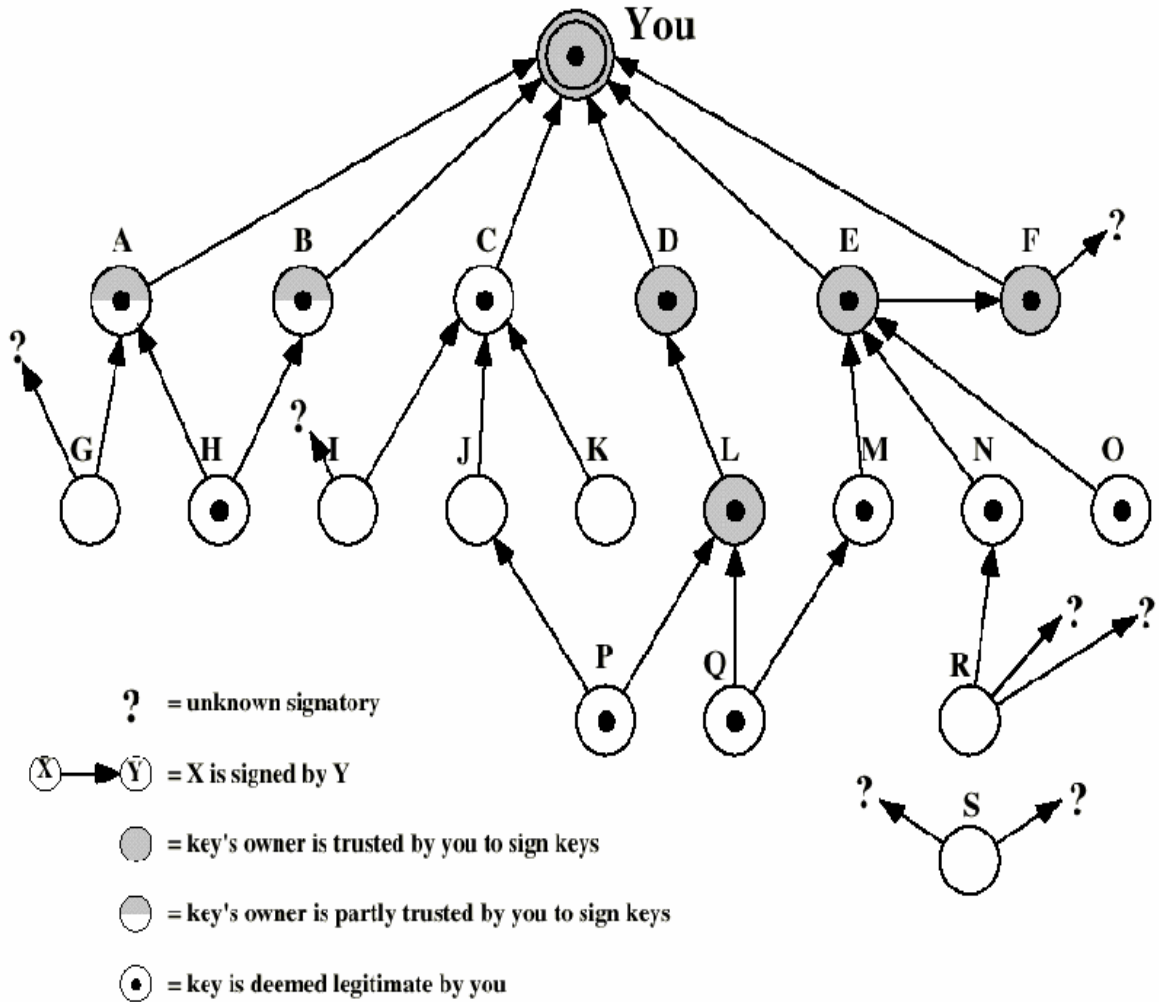
X.509 ITU standard adopted by IETF and widely used on the Internet

IETF version of X.509 is RFC 3280

X.509 is a way to describe certificates (meta- data or schema written in ASN.1)

| Field | Meaning |
|---------------------|--|
| Version | Which version of X.509 |
| Serial number | This number plus the CA's name uniquely identifies the certificate |
| Signature algorithm | The algorithm used to sign the certificate |
| Issuer | X.500 name of the CA |
| Validity period | The starting and ending times of the validity period |
| Subject name | The entity whose key is being certified |
| Public key | The subject's public key and the ID of the algorithm using it |
| Issuer ID | An optional ID uniquely identifying the certificate's issuer |
| Subject ID | An optional ID uniquely identifying the certificate's subject |
| Extensions | Many extensions have been defined |
| Signature | The certificate's signature (signed by the CA's private key) |

Compareback with Zimmerman's "Web of trust":



Both have their Roles!

Which is superior? depends!

An alternative to SSL – IPsec

Result of an argument over Internet security

Security mechanisms should be end to end hence in the applications
But users don't understand security & all applications will have to be changed

OK next best thing put it into the Transport protocol or just above it.
Well lets put it in the network because it will help the security naïve

The complete Ipsec is a complete a la carte framework
Multiple services, algorithms & granularities
RFCs 2401, 2402, 2406 (RFC 2410 the null algorithm)

Major services

Secrecy, integrity, protection from replay

Symmetric key cryptography for performance

Algorithm independent so new ones can be introduced in the future

Connection oriented – Security Association

2 Major Parts

2 new headers added to IP packets

Internet Security Association & Key Management Protocol (ISAKMP)

IPSecmodes

Transport Mode

The Ipsec header is inserted just after the IP header

Tunnel Mode

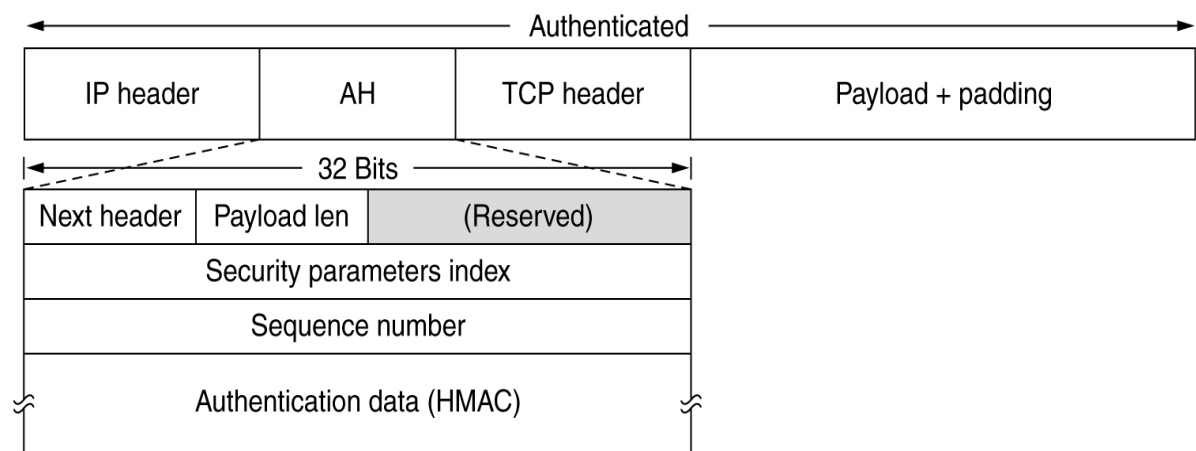
The entire IP packet header and all is encapsulated in the body of the new packet with a new IP header

This is ideal when the two end points are firewalls. So between secure networks the firewalls can encapsulate/decapsulate packets as they pass through the firewall. This means company machines don't have to worry about Ipsec.

A number of TCP connections can be bundled encrypted & encapsulated in one packet. This can hide the number of communications.

IPSecauthentication header in Transportlevel (IPv4)

NOTE: No Encryption!

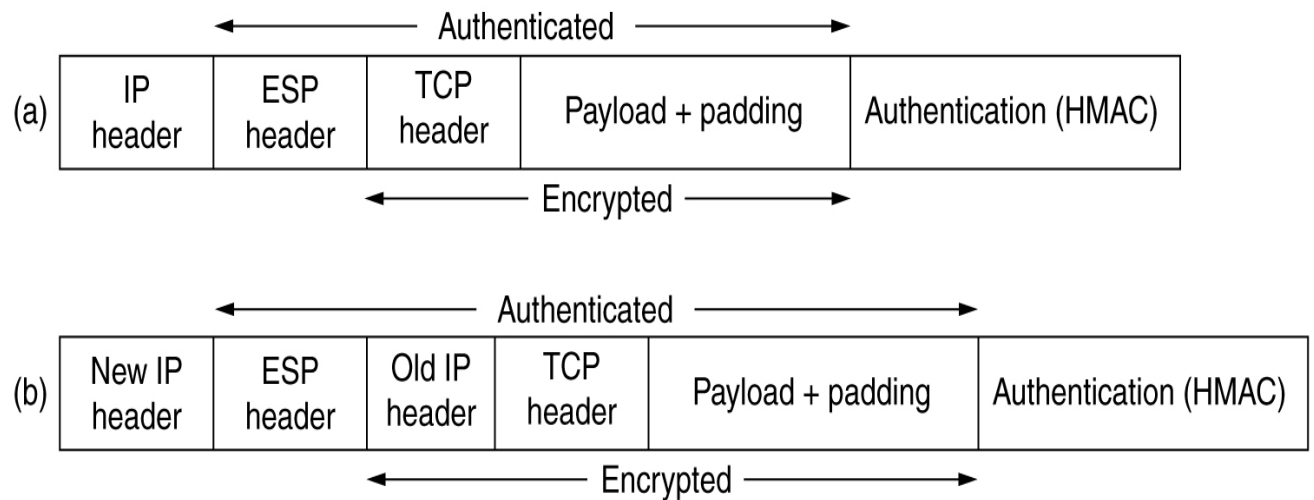


Encapsulating Security Payload (ESP)

transport mode

tunnel mode

Note: ESP will probably make AH obsolete



Needham-Schroeder Key Exchange

- Predates public key
- Symmetric key exchange with trusted third party

From wikipedia:

S is a server trusted by both parties

K_{AS} is a symmetric key known only to A and S

K_{BS} is a symmetric key known only to B and S

N_A and N_B are nonces

The protocol can be specified as follows in security protocol notation:

$$A \rightarrow S : A, B, N_A$$

Alice sends a message to the server identifying herself and Bob, telling the server she wants to communicate with Bob.

$$S \rightarrow A : \{N_A, K_{AB}, B, \{K_{AB}, A\}_{K_{BS}}\}_{K_{AS}}$$

The server generates K_{AB} and sends back to Alice a copy encrypted under K_{BS} for Alice to forward to Bob and also a copy for Alice. Since Alice may be requesting keys for several different people, the nonce assures Alice that the message is fresh and that the server is replying to that particular message and the inclusion of Bob's name tells Alice who she is to share this key with.

$$A \rightarrow B : \{K_{AB}, A\}_{K_{BS}}$$

Alice forwards the key to Bob who can decrypt it with the key he shares with the server, thus authenticating the data.

$$B \rightarrow A : \{N_B\}_{K_{AB}}$$

Bob sends Alice a nonce encrypted under K_{AB} to show that he has the key.

$$A \rightarrow B : \{N_B - 1\}_{K_{AB}}$$

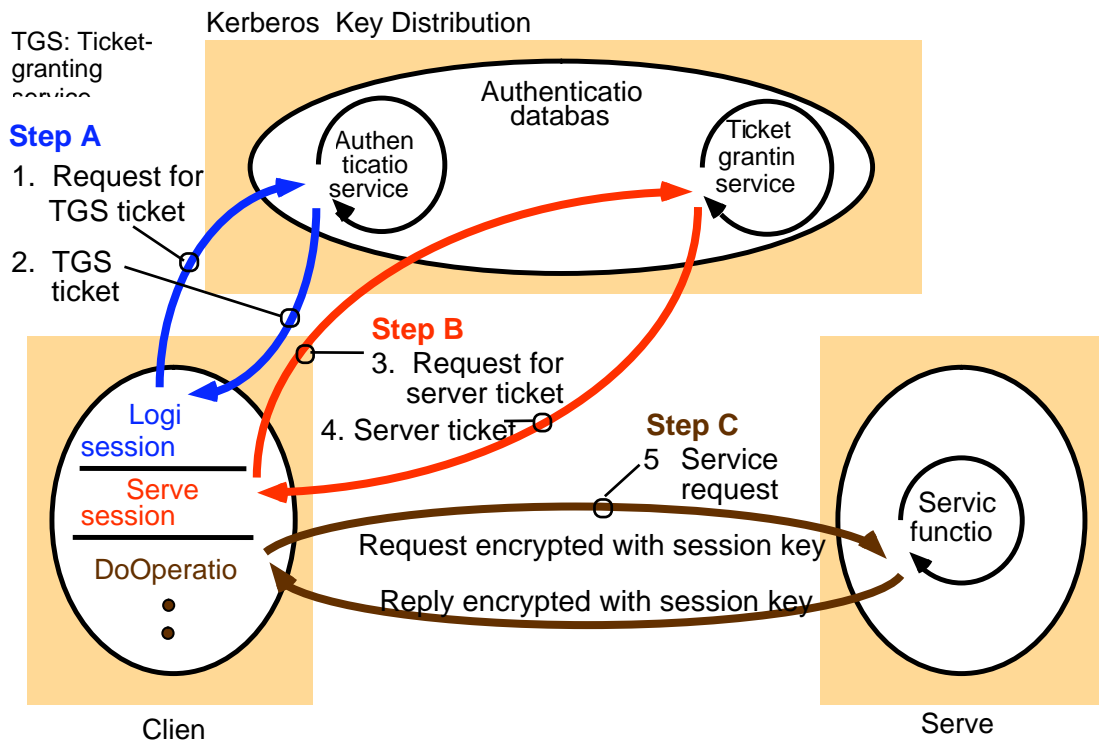
Alice performs a simple operation on the nonce, re-encrypts it and sends it back verifying that she is still alive and that she holds the key.

The protocol is vulnerable to a replay attack. If an attacker records one run of this protocol, then subsequently learns the value K_{AB} used, she can then replay the message $\{K_{AB}, A\}_{K_{ES}}$ to Bob, who will accept it, being unable to tell that the key is not fresh. This flaw is fixed in the Kerberos protocol by the inclusion of a timestamp.

Kerberos

- Secures communication with servers on a local network
 - Developed at MIT in the 1980s to provide security across a large campus network > 5000 users
 - based on Needham - Schroeder protocol
- Standardized ☺ and now included in many operating systems
 - Internet RFC 1510, OSF DCE
 - BSD UNIX, Linux, Windows 2000, NT, XP, etc.
 - Available from MIT
- Kerberos server creates a shared secret key for any required server and sends it (encrypted) to the user's computer
- User's password is the initial secret shared with Kerberos

From CDK:



Step A once per login session

Step B once per server session

Step C once per server transaction

Needham - Schroeder protocol

- 1.A - A, B, N_A
- 2.S - $\{N_A, B, K_{AB}, \{K_{AD}, A\}_{K_D}\}_{K_A}$
- 3.A - $\{K_{AD}, A\}_{K_D}$
- 4.B - $\{N_D\}_{K_{AD}}$
- 5.A - $\{N_D - 1\}_{K_{AD}}$

Problemapplication: NFS

- Kerberos protocol is too costly to apply on each NFS operation
- Kerberos is used in the mount service:
 - to authenticate the user's identity
 - User's UserID and GroupID are stored at the server with the client's IP address
- For each file request:
 - UserID and GroupID are sent encrypted in the shared session key
 - The UserID and GroupID must match those stored at the server
 - IP addresses must also match
- This approach has some problems
 - can't accommodate multiple users sharing the same client computer
 - all remote filestores must be mounted each time a user logs in