

Security in Networks

Ian Johnson

Room 3Q77

Telephone: extension 83167

Email: Ian.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!

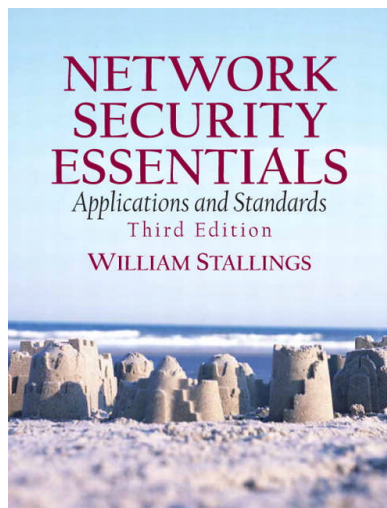
Lab Work:

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

Other good 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

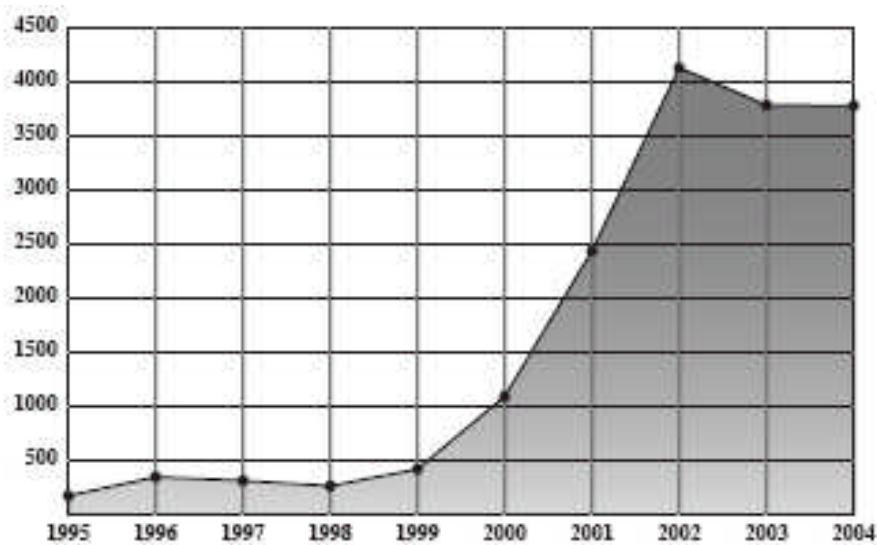
Hatch, B. & Lee, J., “Hacking Linux Exposed”, 2nd Ed. , 2003.

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

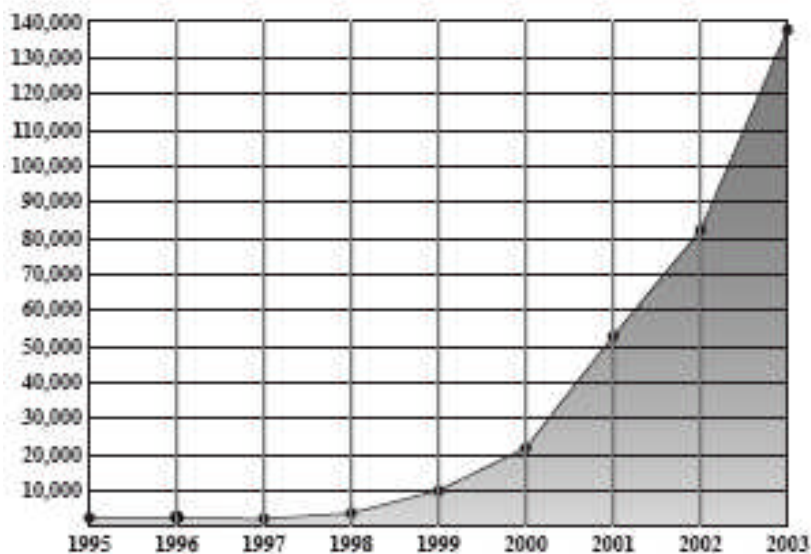
I Know I'm Paranoid, But am I paranoid enough? 😊

Why should we care about Network Security?

Lets look at some of Stallings (ch1) graphs:



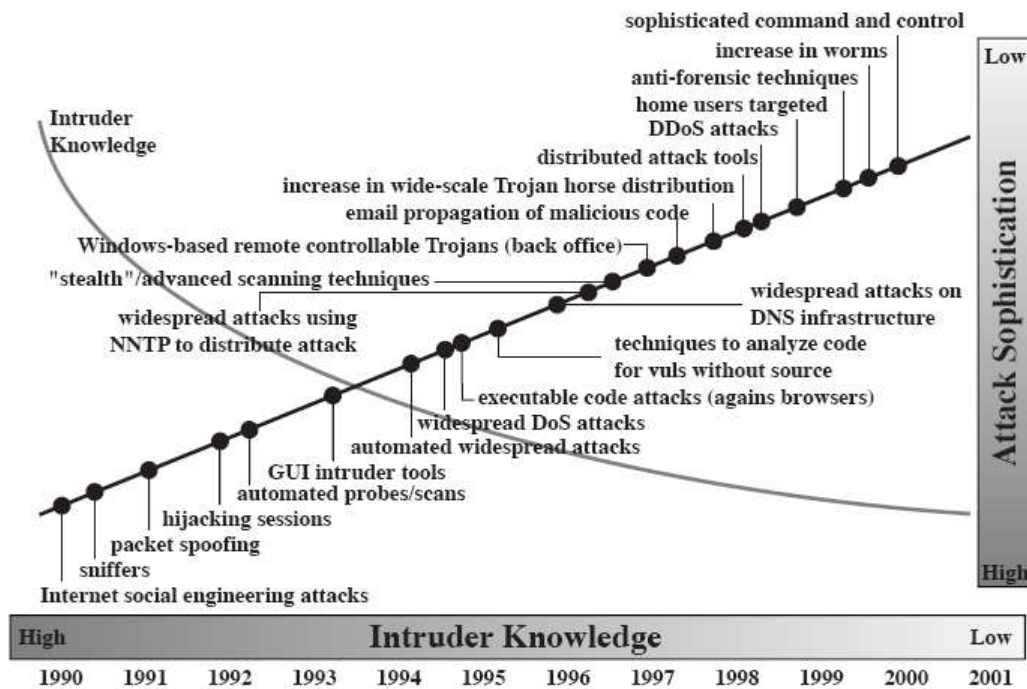
(a) Vulnerabilities: reported



(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

SWOT analysis

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.

Security requirements 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!

Penetration testing

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

Security Goals

Confidentiality (privacy)

Authentication (who created or sent the data)

Integrity (has not been altered)

Non-repudiation (the order is final)

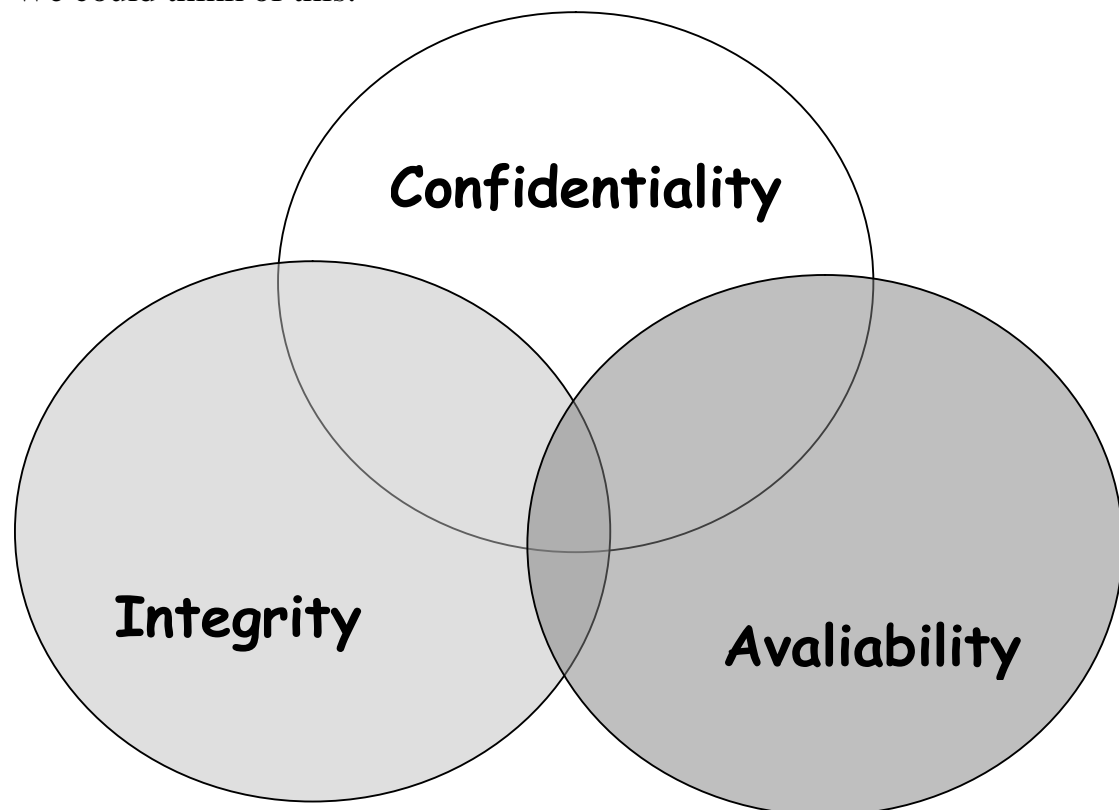
Access control (prevent misuse of resources)

Availability (permanence, non-erasure)

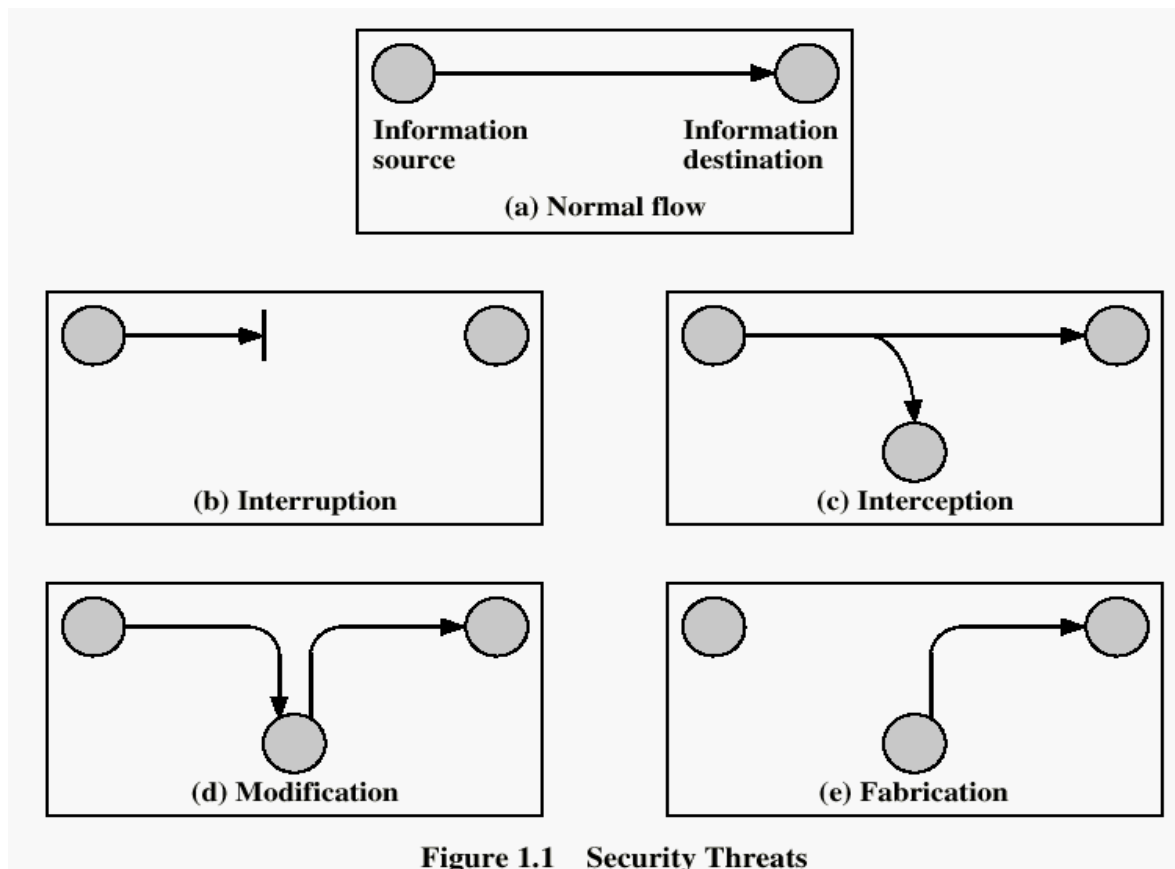
Denial of Service Attacks

Virus that deletes files

We could think of this:



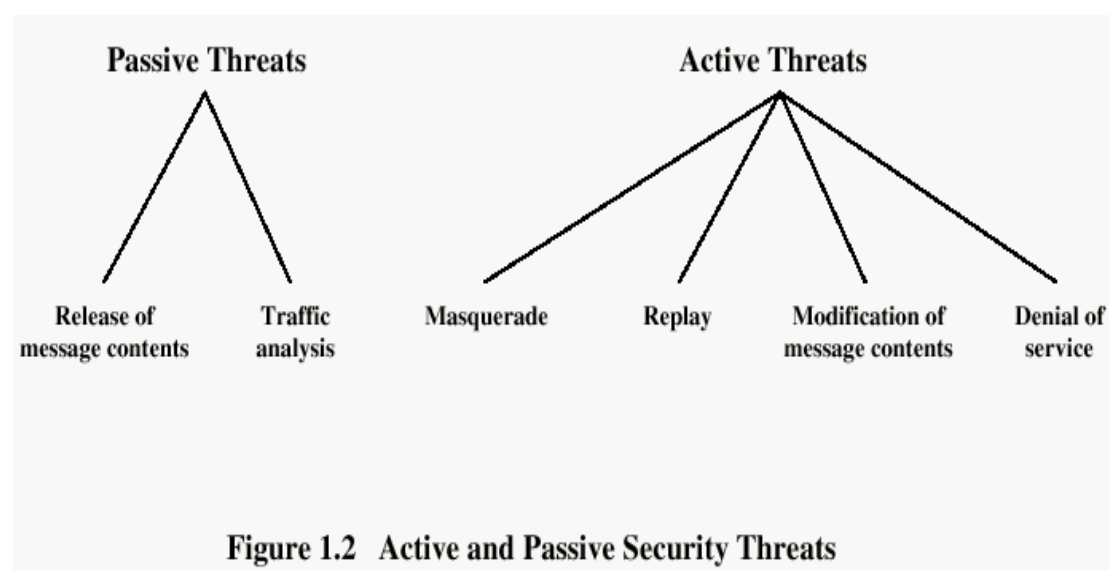
Security Threats:



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

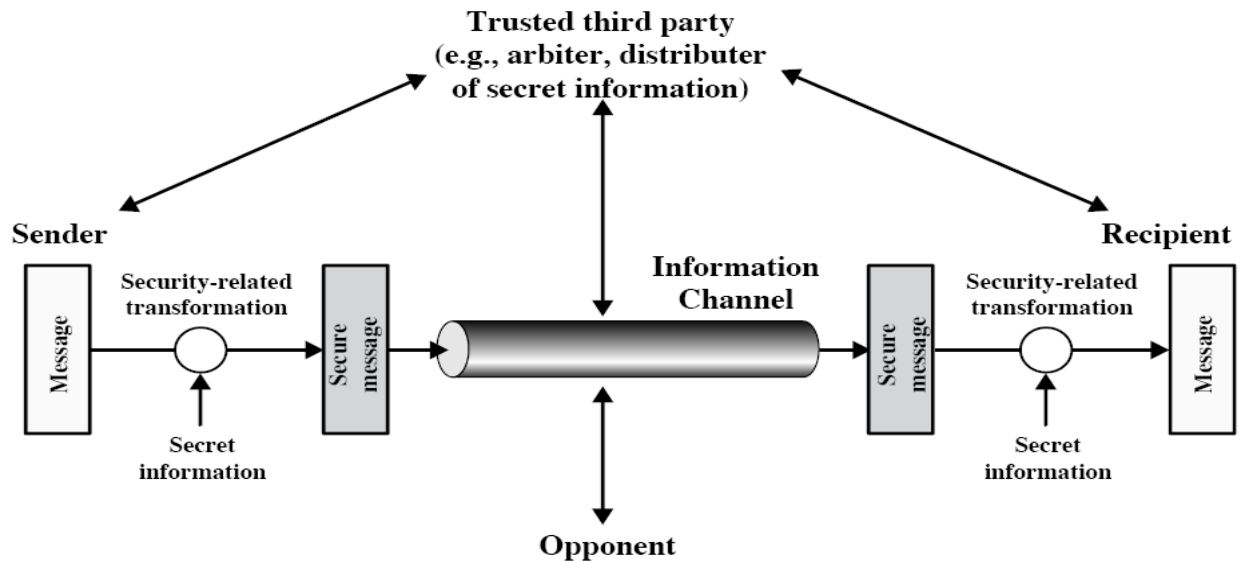
Service	Mechanism							
	Enciph-erment	Digital signature	Access control	Data integrity	Authenti-cation exchange	Traffic padding	Routing control	Notari-zation
Peer entity authentication	Y	Y			Y			
Data origin authentication	Y	Y						
Access control			Y					
Confidentiality	Y						Y	
Traffic flow confidentiality	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!

Cryptography 101

“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.

Kerckhoff in 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.

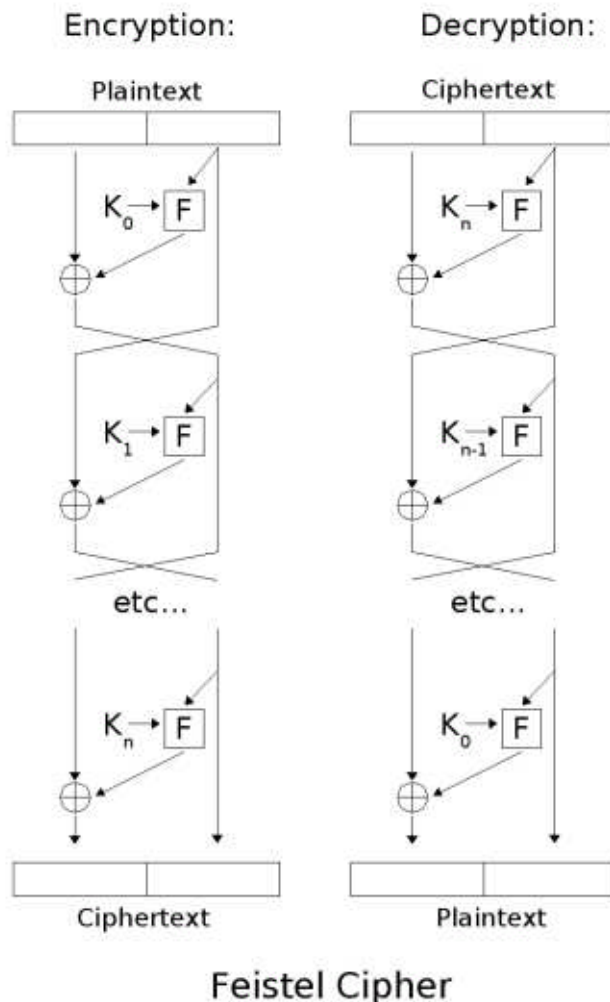
Then the ciphertext is (L_n, R_n) .

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.

Symmetric Stream 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!

Random Numbers

“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:

$$I_{j+1} = aI_j \pmod{m}$$

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

```