Encryption and Security Tutorial

Peter Gutmann

University of Auckland

http://www.cs.auckland.ac.nz/~pgut001

Security Requirements

Confidentiality

- Protection from disclosure to unauthorised persons

Integrity

- Maintaining data consistency

Authentication

- Assurance of identity of person or originator of data

Non-repudiation

- Originator of communications can’t deny it later

Security Requirements (ctd)

Availability

- Legitimate users have access when they need it

Access control

- Unauthorised users are kept out

These are often combined

- User authentication used for access control purposes
- Non-repudiation combined with authentication

Security Threats

Information disclosure/information leakage

Integrity violation

Masquerading

Denial of service

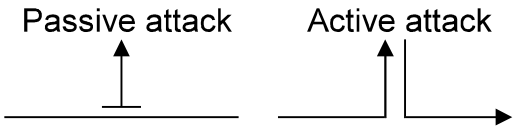
Illegitimate use

Generic threat: Backdoors, trojan horses, insider attacks

Most Internet security problems are access control or authentication ones

- Denial of service is also popular, but mostly an annoyance

Attack Types



Passive attack can only observe communications or data

Active attack can actively modify communications or data

- Often difficult to perform, but very powerful
  - Mail forgery/modification
  - TCP/IP spoofing/session hijacking

Attack Types (ctd)

Strong, effectively unbreakable crypto is universally available (despite US government efforts in the 1990s)

- Don't attack the crypto, attack the infrastructure within which it's used
- " " " " implementation
- " " " " users
  - See my Internet threat convergence tutorial for more on the latter

Security Services

From the OSI definition:

- Access control: Protects against unauthorised use
- Authentication: Provides assurance of someone's identity
  - Often confused with authorisation
- Confidentiality: Protects against disclosure to unauthorised identities
- Integrity: Protects from unauthorised data alteration
- Non-repudiation: Protects against the originator of communications later denying it

Security Mechanisms

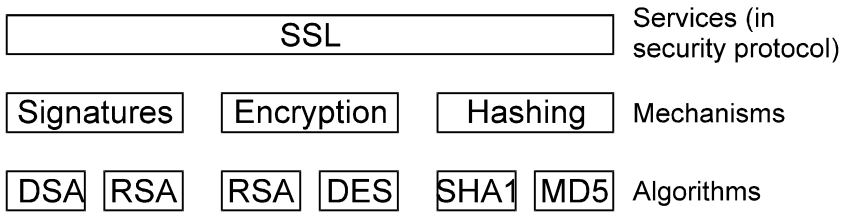
Three basic building blocks are used:

- Encryption is used to provide confidentiality, can provide authentication and integrity protection
- Digital signatures are used to provide authentication, integrity protection, and non-repudiation
- Checksums/hash algorithms are used to provide integrity protection, can provide authentication

One or more security mechanisms are combined to provide a security service

Services, Mechanisms, Algorithms

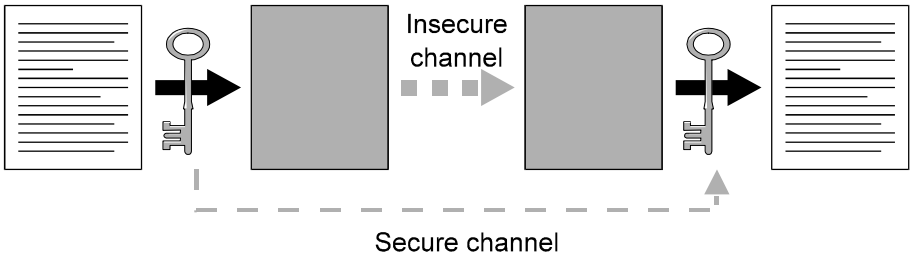
A typical security protocol provides one or more services



- Services are built from mechanisms
- Mechanisms are implemented using algorithms

Conventional Encryption

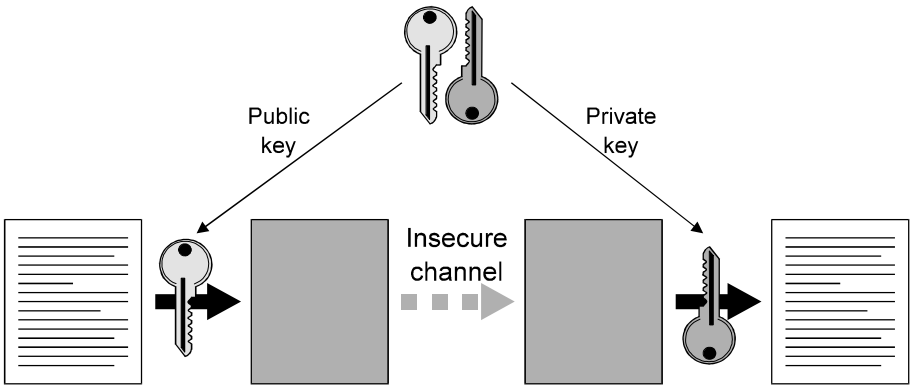
Uses a shared key



Problem of communicating a large message in secret is reduced to communicating a small key in secret

Public-key Encryption

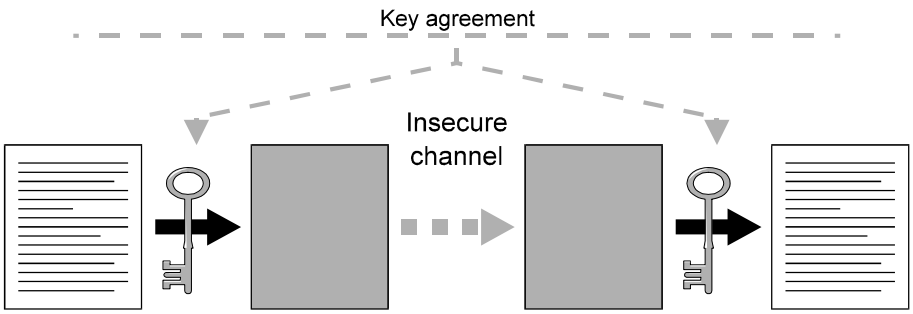
Uses matched public/private key pairs



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

Key Agreement

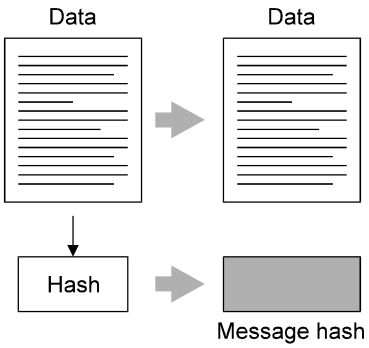
Allows two parties to agree on a shared key



Provides part of the required secure channel for exchanging a conventional encryption key

Hash Functions

Creates a unique “fingerprint” for a message

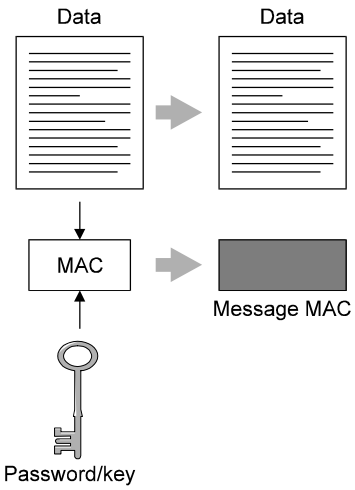


Anyone can alter the data and calculate a new hash value

- Hash has to be protected in some way

MAC’s

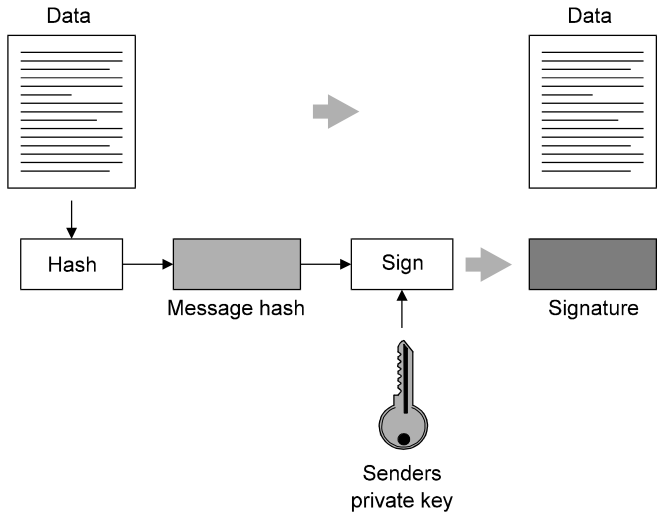
Message Authentication Code, adds a password/key to a hash



Only the password holder(s) can generate the MAC

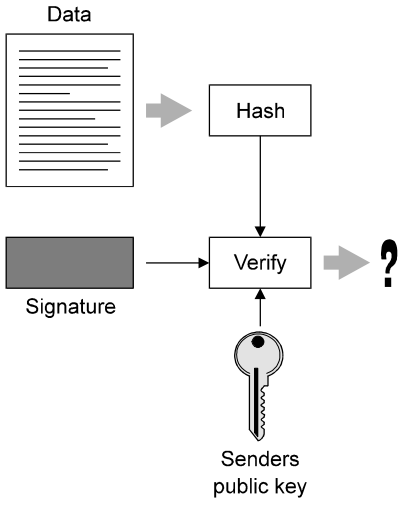
Digital Signatures

Combines a hash with a digital signature algorithm



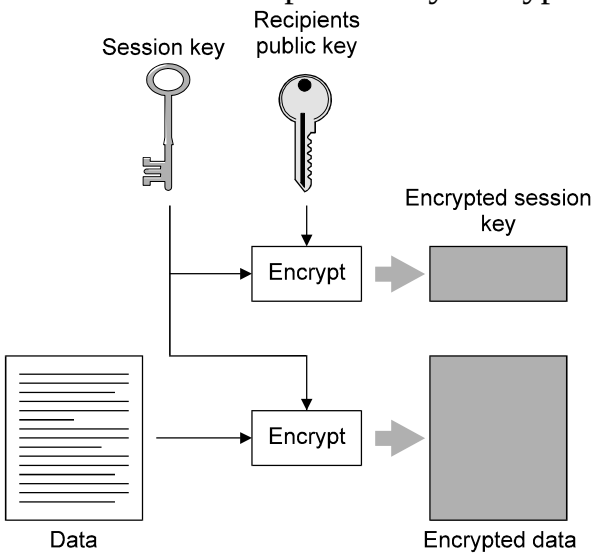
Digital Signatures (ctd)

Signature checking:

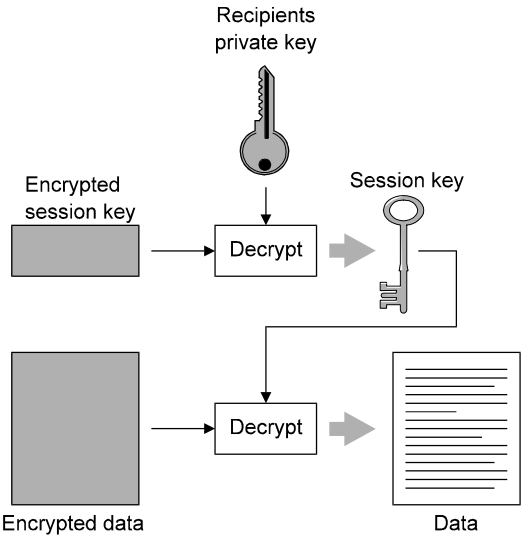


Message/Data Encryption

Combines conventional and public-key encryption



Message/data Encryption (ctd)



Public-key encryption provides a secure channel to exchange conventional encryption keys

Data Formats

One obviously-correct format for secured content

Information required to process payload
Payload
Result of processing payload

- Allows straightforward one-pass processing for encapsulation and decapsulation

Data Formats (ctd)

Signed data

Hash algo.for payload
Payload
Signature on payload

MACd data

Keying info for MAC
Payload
MAC on payload

Encrypted data

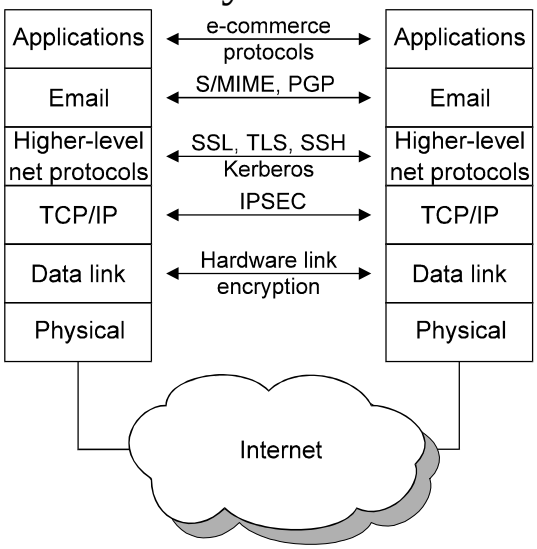
Keying info for encryption
Encrypted payload

Keying info = password derivation info, public-key-encrypted content-encryption key, …

This single obvious format is why PGP and S/MIME, SSL and SSH differ mostly in their bit-bagging formats

- Doesn't prevent standards groups from coming up with different (broken) versions

Security Protocol Layers

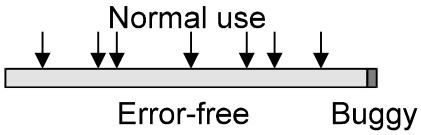


The further down you go, the more transparent it is
The further up you go, the easier it is to deploy

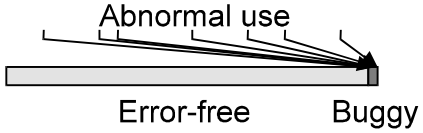
Why Security is Harder than it Looks

All software has bugs

Under normal usage conditions, a 99.99% bug-free program will rarely cause problems



A 99.99% security-bug-free program can be exploited by ensuring the 0.01% instance is always encountered



This converts the 0.01% failure to 100% failure

Why Security is Harder than it Looks (ctd)

Customers have come to expect buggy software

- Correctness is not a selling point
- Expensive and time-consuming software validation and verification is hard to justify

Solution: Confine security functionality into a small subset of functions, the trusted computing base (TCB)

- In theory the TCB is small and relatively easy to analyse
- In practice vendors end up stuffing everything into the TCB, making it a UTCB
- Consumers buy the product anyway (see above)