Security in Computing & Information Technology

Lecture 3
Security Mechanisms
Elementary Cryptography

Lecture Schedule

Foundations

1. Introduction
2. Vulnerabilities, Threats, Attacks

Basic mechanisms

3. Security mechanisms, Elementary cryptography
4. Authentication
5. Access control

Major computing security areas

6. Operating systems
7. Databases
8. Networks
9. Web
10. Mobile computing

Applications

11. Privacy
12. Internet banking

Lecture Topics

- Security mechanisms
- Encryption basics
- Secure digest functions
- Digital signatures

The Security Process

- Security is not a static feature
  New threats emerge regularly
  - Technology changes
  - New vulnerabilities are discovered in old systems
  - People change, forget practices, …
- Security life cycle (infinite loop)
  - Plan
  - Implement
  - Evaluate

Evolution of Technology

NASA Space Flight Control Centre

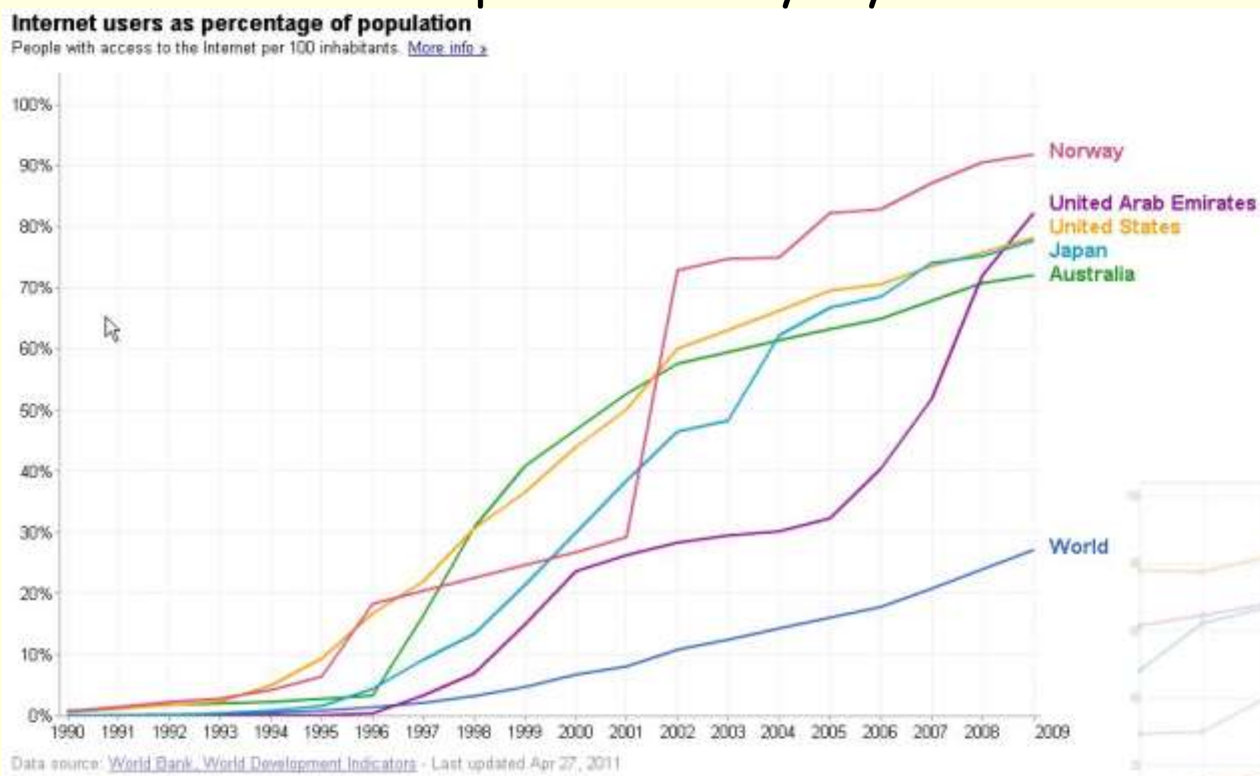
At the time of Moon landing

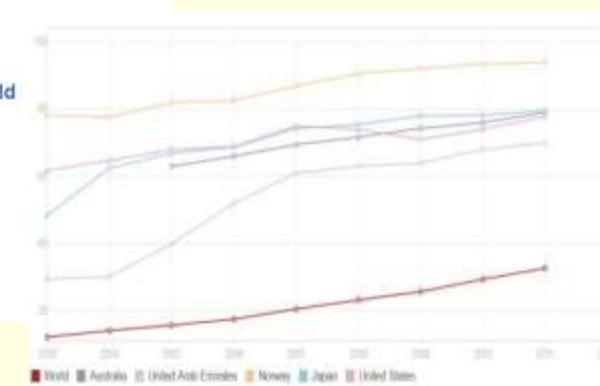
Today

Tomorrow
?

The Current Landscape

- Increased Internet usage
  - It has become part of everyday infrastructure





- Insufficient risk/threat awareness

Security Problem Causes

- Software vulnerabilities
  - Problems (features) that allow unspecified behaviour
- Maintenance failure
  - Not updating software, not applying fixes, hardware maintenance issues
- Operational problems
  - Negligence (e.g. not protecting passwords), lack of expertise

Computer Accounts – Legal Aspects

- Provide authorised access to a computer
- Users have rights and responsibilities
- Should be used only for the purpose it was provided for

Security Mechanisms

- Implement security services
- Deal with
  - prevention of incidents
  - detection of incidents
  - recovery from incidents
- Are characterised by resilience
  - Ability to operate in a hostile environment
  - Fault tolerance

Security Mechanisms in Computing

- Protect data content from unauthorised
  - access and
  - modification
- Mostly rely on cryptographic methods
  - To hide content
  - To maintain integrity
- Examples

  Encryption, authentication, access control lists (ACLs)

Types of Security Mechanisms

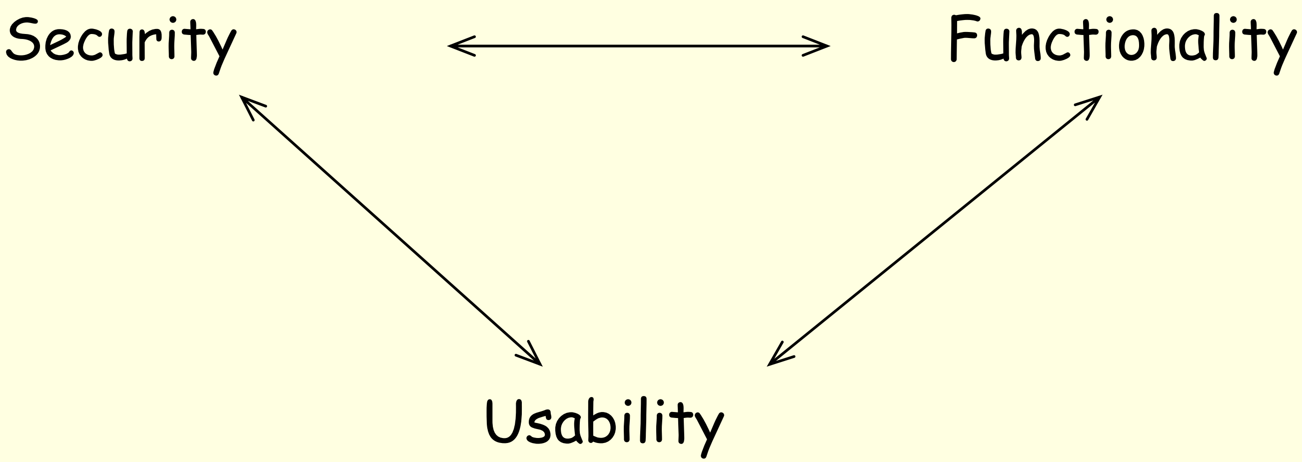
- Pervasive mechanisms
  - Protect against a number of threats
    - E.g. network firewalls
  - Protect individual computers or whole networks
    - E.g. virus checking programs (email filters)
  - More economical, less accurate
- Specific mechanisms
  - Protect against a specific threat
    - E.g. data integrity protection
  - Protect an individual data or a piece of hardware
    - E.g. controlling access to individual data items
  - More accurate, less economical

The Cost of Security

- Direct costs
  - Software, equipment, procedures
- Indirect costs
  - Reduced efficiency due to additional procedures
- Savings
  - Avoiding possible, expensive damage
  - Potentially: optimisation of procedures

Security Tradeoffs

- Security features may restrict functionality
  E.g. certain network connections are not allowed
- Security mechanisms may complicate user interaction
  E.g. additional procedures can be required for performing certain operations
- Proper balance between them is needed: Risk analysis



Risk Analysis

- Potential loss in case of an accident
  - Value of assets
    - Replacement value (equipment, software)
    - Potential damage (loss of data, privacy)
  - Probability of an accident
    - Identify potential accidents
    - Assess their frequency
  - Risk = asset value * accident probability
- Risk assessment

Vulnerability assessment → Risk assessment

Threat assessment → Risk assessment

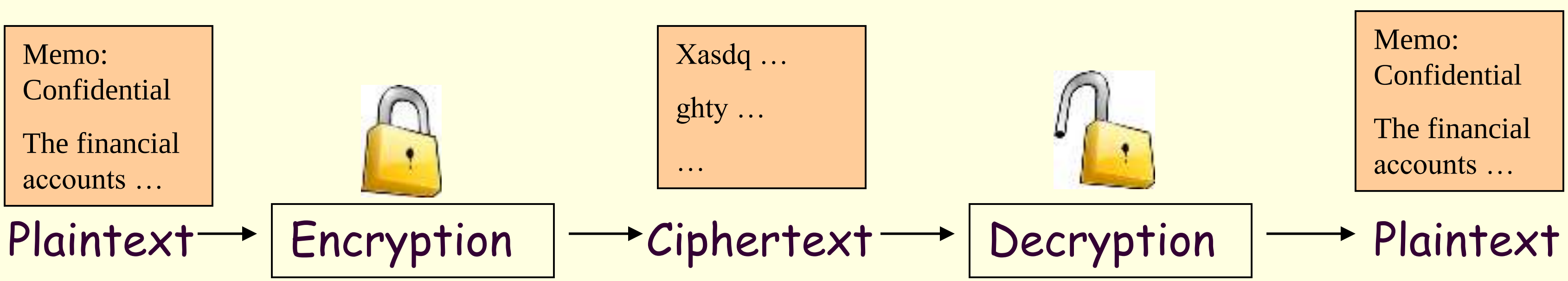
Encryption / Decryption

- Encryption (encoding, enciphering): processing a message so that its meaning becomes obscured
- Decryption (decoding, deciphering): the reverse of encryption

  Cyphertext = Encrypt (Plaintext)

  Plaintext: information is clearly understandable

  Ciphertext: information is hidden



- Historical terms
  - Cryptosystem: code, cipher (encoding / decoding, enciphering / deciphering)

Terminology

- Breakable encryption
  - The encryption / decryption algorithm can be determined without prior knowledge
  - Theoretically breakable: there is a method to break the encryption
  - Practically breakable: it can be done within reasonable time
- Cryptography

  (The art and science of) keeping messages secure
- Cryptanalysis

  (The art and science of) breaking ciphertext
  - Break a single message
  - Devise a method to break all messages
  - Find weaknesses in the algorithm or in its implementation

Encryption Methods and Keys

- Encryption method
  - The algorithm used to transform the plaintext (e.g. substitute each letter with another letter in the alphabet)
- Encryption key
  - Parameter that enables to translate the same plaintext with the same algorithm to different ciphertexts
  - Ciphertext becomes the function of (Plaintext + Key)
- Good encryption methods rely on the key for secrecy
  - No need to invent a new method for every application
  - Most commonly used encryption algorithms are published
  - Breaking the encryption requires finding the key

Secret and Public Key Encryption

- Encryption / decryption keys

  The ciphertext depends on the original plaintext, the algorithm, and a parameter called key

  Cyphertext = Encrypt (Plaintext, Key)

- Secret (symmetric) key encryption: E_Key can be easily calculated from D_Key
  - Simple, fast
- Public key (asymmetric) encryption: The keys cannot be calculated from each other in reasonable time
  - More secure, very slow

Cryptanalysis
Attacks against Encryption (1)

- Ciphertext-only attack
  - The cryptanalyst has access to encrypted messages only, the aim is to recover the plaintext, and possibly deduce the encryption / decryption key
- Known-plaintext attack
  - The cryptanalyst has access not only to the ciphertext, but also to the plaintext of those messages; the aim is to recover the key
- Chosen-plaintext attack
  - The cryptanalyst can even choose the plaintext of the messages
- Adaptive-chosen-plaintext attack
  - The cryptanalyst chooses the plaintext by using the results of previous encryptions (more efficient than simple chosen-plaintext attack)
- Chosen-ciphertext attack
  - The cryptanalyst can choose ciphered messages and has access to the decrypted messages
- Chosen-key attack
  - The key is given; used for evaluating an algorithm, not really an attack

Cryptanalysis
Attacks against Encryption (2)

- Brute force attack: tries all possible solutions
  - (There may be a way easier than brute force to break the encryption)
- Classes
  - P (polynomial):
    - Problems for which the solution growth rate is a polynomial function
  - NP (nondeterministic polynomial):
    - The correctness of a guessed solution can be checked in polynomial time
  - EXP (exponential)
    - A deterministic solution exists in exponential time

Practical Security of Cryptosystems

- Security
  - Theoretically not breakable
    - Unconditionally secure
  - Practically not breakable
- Computationally secure or strong
  - Work factor
    - Computing time and power needed to recover the key
    - E.g. work factor = 2^{128} (2^{128} operations are needed)
  - Operations' complexity and computing time may change
- Confusion
  - Complexity of the relationship between plaintext and ciphertext
    - Breaking a few messages should not allow the breaking of all messages
- Diffusion
  - How the statistical properties of the encrypted text reflect the statistical properties of the plaintext

Stream & Block Ciphers

- Stream ciphers
  - The transformation depends only on the actual symbol, does not consider the previous or next symbol(s)
  - Fast
  - Low error propagation
  - Low diffusion (easy to break), susceptible to malicious insertions, modifications
- Block ciphers
  - Transforms a group of data (a block) at a time
  - Higher diffusion, immune to insertions
  - Slower encryption (Has to wait for whole blocks)
  - Error propagation problems
    One error affects a number of symbols

Classical Secret-Key Algorithms

Substitutions Ciphers

- Simple substitution cipher
  - One character of plaintext replaced with a corresponding character
- Homophonic cipher
  - A single character can map to one of several characters
- Polyalphabetic substitution cipher
  - Multiple simple substitution ciphers
  - The actual one used changes with the position of each character
- Polygram substitution cipher
  - Blocks of characters are encrypted in groups

Simple Substitution

Caesar Cipher

- Translate a letter to the letter *n* places to the right in the alphabet

  $c_i = E(p_i) = p_i + n$

  E.g. n = 3

a	b	c	d	e	f	g	…
↓	↓	↓	↓	↓	↓	↓	↓
d	e	f	g	h	i	j	…

"treaty impossible" → "wuhdwb lpsrvvleoh "

Other Substitution Methods

- Homophonic substitution
  - A single character can map into one of several characters in ciphertext e.g. 'A' $\rightarrow$ 3 or 5 or 7
- Polyalphabetic substitution
  - First key encrypts the first letter, second key the second letter etc. after using all keys, the keys are recycled

Transposition Ciphers

- (used in World War I)
- Characters of the plaintext remain the same, but their order is changed
- Arrange the text in columns (or in other patterns)

  E.g.

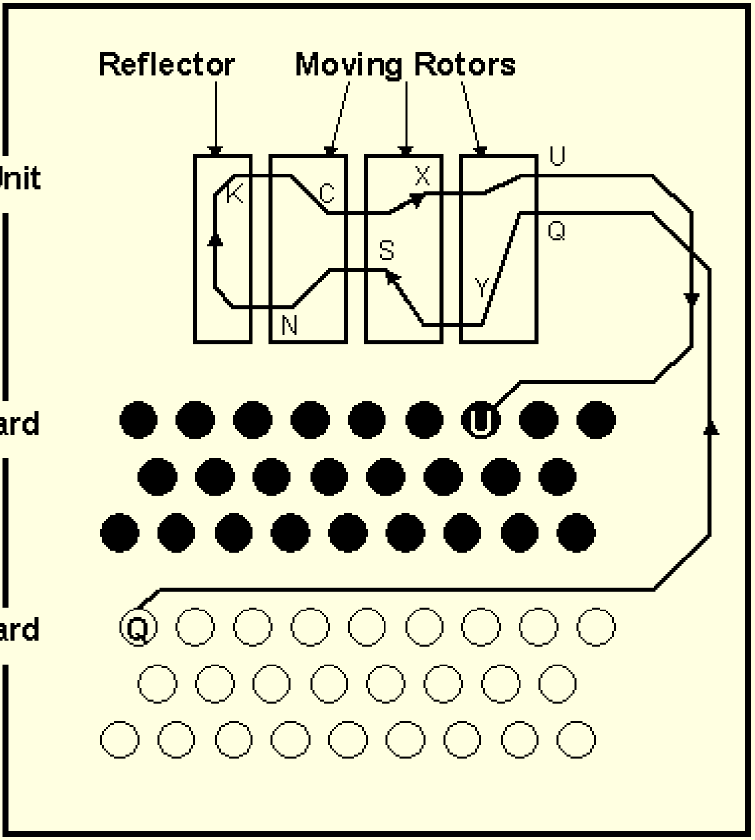
T	I	M	A	write direction ↓
H	S	E	G	
I		S	E	
S	A	S		

  timahsegi sesas   read direction →

- Complexity
  - Time: proportional to the length of message
  - Space: length of the message (not good for long messages)
- Usage
  - Substitution is far more common

Rotor Machines

- Enigma (used in World War II)
  http://www.math.miami.edu/~harald/enigma/enigma.html
- Automate the process of encryption
  - Each rotor makes a substitution
  - After the substitution the last rotor rotates one step
- Combination of rotors makes it difficult to break (Period 26^n)



One-Time Pads

- Method
  - Pad of a large, non-repeating set of truly random keys
  - Each letter of the message is encrypted with a corresponding key from the pad
  - Used once, then destroyed (otherwise not secure)
- Perfect (theoretically not breakable) as long as keys are randomly selected
- Problems
  - Safe only with really random numbers (not with pseudo-random ones)
  - Length of key sequence is equal to the length of message (not feasible for a 1 Mbps channel)
- Usage
  - Ultrasecure low-bandwidth channels
  - One-time passwords are similar constructs

Secret-Key Encryption Example (1)

- Data Encryption Standard (DES 1977)
  - Was the most widely used algorithm until the late 1990s
  - Maps a 64 bit plain text into a 64 bit ciphertext using a 56 bit key
  - Has 16 key-dependent rounds, in which data is rotated and transposed
    - Split data in half, scramble right half, swap two halves
  - Successful attacks against it are possible
    - key is small enough for brute force attack
    - has been around for quite long, has been well analysed
  - Triple DES
    - DES is not considered to be secure anymore
    - Triple DES uses DES three times with three different keys
    - Most often as encrypt-decrypt-encrypt (EDE)

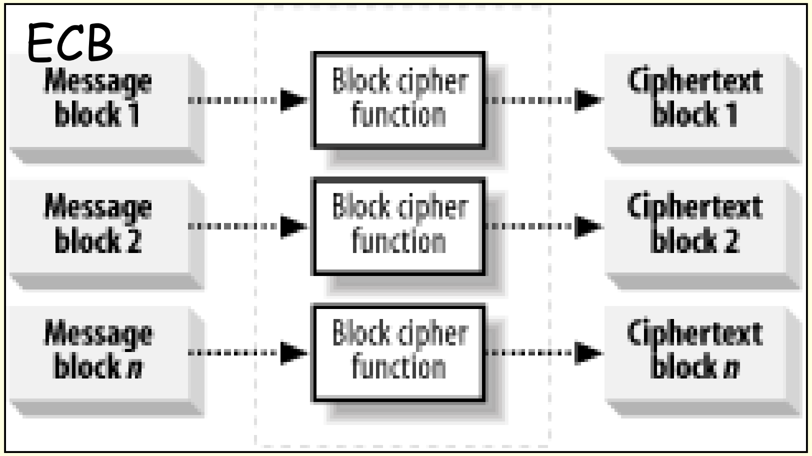
Secret-Key Encryption Example (2)

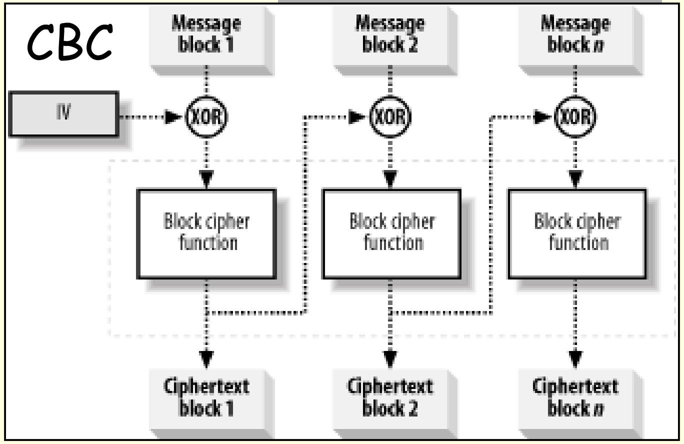
- Advanced Encryption Standard (AES)
  - Adopted as a standard in 2001
  - A version of the Rijndael block cipher
    - block size: 128 bits (4x4 array of bytes)
    - key sizes: 128, 192, 256 bits (10, 12, 14 rounds of calculations)
  - Each round has four steps
    - AddRoundKey each byte is combined (XOR-ed) with the subkey
    - SubBytes non-linear substitution of each byte by using a lookup table
    - ShiftRows cyclically shifts the bytes in each row by a certain offset
    - MixColumns combines the bytes in each column by using a linear transformation (in the last round this is replaced by another AddRoundKey)
  - A brute-force attack (computationally prohibitively expensive) was published in 2002

Cipher Modes

- Electronic code book (ECB)
  - A plaintext always encrypts to the same ciphertext
  - A "code book" can be built for each key (and any plaintext-ciphertext combination can be entered into this book)
  - Suitable e.g. for database encryption
- Improved method: Cipher block chaining (CBC)
  - The plaintext is XOR-ed with the previous ciphertext block and then encrypted
  - At decryption time the block is
    - (i) decrypted and
    - (ii) saved as ciphertext for feedback until the next block is decrypted
  - A random initialisation vector (IV) is used for the first block
  - Error propagation and extension
    - A small error in the ciphertext can lead to a large error in the decrypted plaintext – needs integrity protection

ECB and CBC





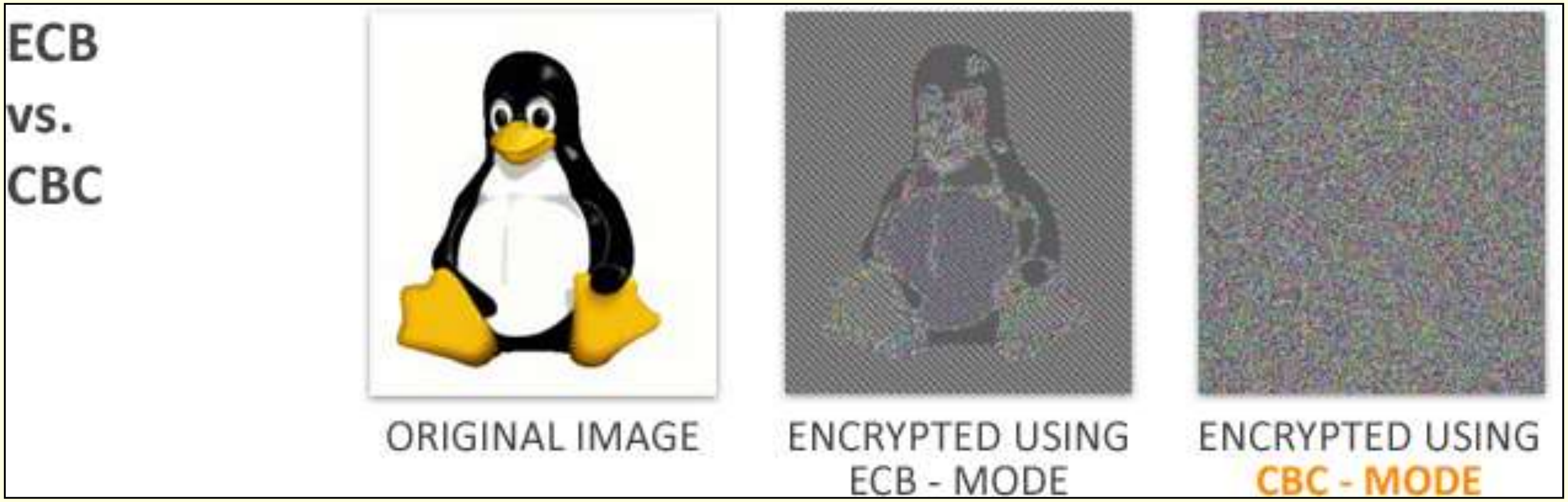


Image source: http://programming4.us/security/1604.aspx, http://blockmastersecurity.com

Problems of Symmetric Key Systems

- If key is revealed, security is broken
  (Keys in real systems are changed fairly frequently)
- Distribution of keys should be secure
  By hand (e.g. by a courier), in pieces on separate channels, etc.
- Simple methods can be vulnerable to attacks
- Number of keys increases with the square of the number of participants

Public-Key Systems

- Public and private keys
  - Private encryption key: message can not be falsified
    Used for verifying authenticity: digital signature
  - Private decryption key: message can not be decoded
    Used for confidentiality/secrecy
- Some common methods and their use
  - RSA – encryption and digital signatures
  - El Gamal & DSS – digital signatures
  - Diffie-Hellman – establish a shared secret
- The principle first published in the mid 1970s
- Public-key algorithms are
  - much slower than symmetric-key systems
  - often used to encrypt a symmetric key (session key)
    - Faster data exchange by using a symmetric key
    - Symmetric key is secured by public-key encryption

Diffie-Hellman Key Agreement

- Protocol for establishing a shared secret via an insecure communication channel
- Solution
  - Two parties jointly calculate a shared secret via negotiation
- Features
  - Data exchanged during the negotiation is not sufficient to break the key
  - The established secret is never sent to the other side in any form (encrypted or otherwise)
- The algorithm is very frequently used in secure communication protocols

Secure Digest Functions Principle

- Fixed-length pattern characterising an arbitrary-length message
  $h = H(M)$
  - Given M, it is easy to compute h
  - Given h, it is hard to compute M

  H is a one-way function
  - Given M, it is hard to find another message M' such that $H(M) = H(M')$ – collision-secure
- AKA One Way Hash, Message Digest, Digital Fingerprint
- Usage: Digital signatures, protecting messages from alteration
- Types
  - Non-keyed: depends on the message alone
    AKA message integrity code (MIC), modification detection code (MDC)
  - Keyed: depends on the message and on a secret key
    AKA message authentication code (MAC)

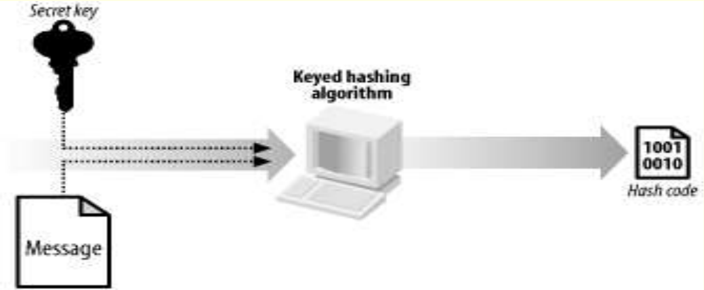


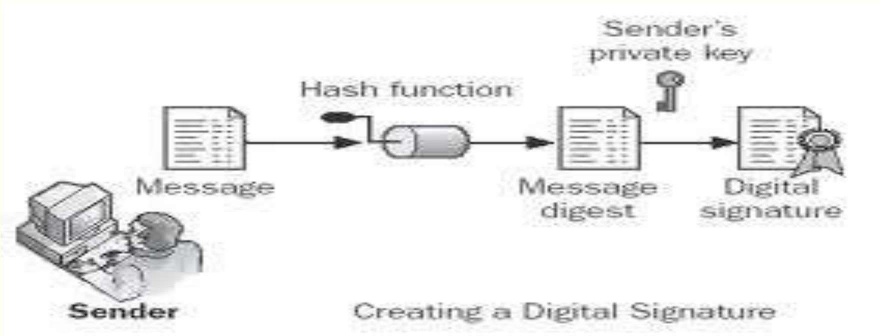
Image source: http://programming4.us/security/1550.aspx

Secure Digest Functions Practical Aspects

- Exploiting collisions: Birthday attack
  1. Alice prepares two versions *M* and *M'*, *M* is favourable for Bob, *M'* is not
  2. Alice makes several versions of *M* and *M'* that are visually indistinguishable from each other (e.g. by adding spaces at the end of lines) until she finds an *M* and an *M'* so that the calculated *h* is the same for the two
  3. Alice sends the favourable document *M* to Bob to sign it
  4. When Bob returns the signed document, Alice replaces *M* with *M'*
- Widely used hash functions
  - MD5: one of the most efficient methods, produces a 128-bit digest, makes only one pass over the data, vulnerable
  - SHA-0, SHA-1: produce a 160-bit digest – attacks have been found
  - SHA-2 (SHA-224, SHA-256, SHA-384, and SHA-512): still considered to be secure

Digital Signatures

- A recipient of a document can verify that the claimed originator is the real originator, and the message has not subsequently been altered.
- Calculated e.g. by encrypting a hash of the document with the signer's private key



- A digital signature is appended to document, i.e.

$\langle M, S, \{ M \} K_S \rangle$ is sent.

Message

Sender

Digital signature

Image source: http://www.microsoft.com/mspress/books/sampchap/6429/0-7356-1877-3.gif

Checking Digital Signatures

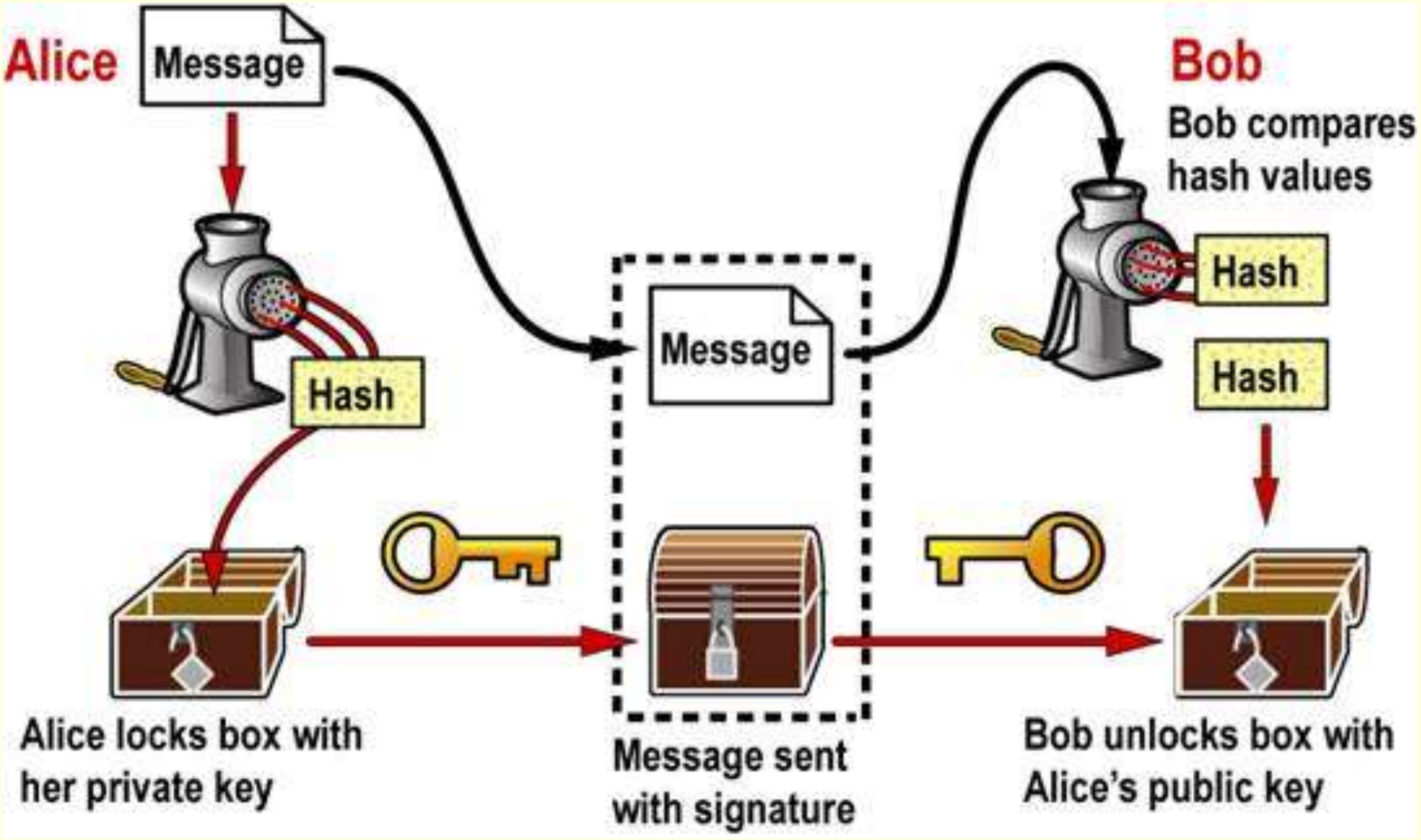


Image source: http://wiki.hill.com/wiki/index.php?title=Digital_signature

Summary

- Encryption is the mostly used way of hiding information content of data
- The main difference between secret-key and public-key encryption methods is in applicability and speed
- Data authenticity can also be proven via cryptographic methods