Cybersecurity

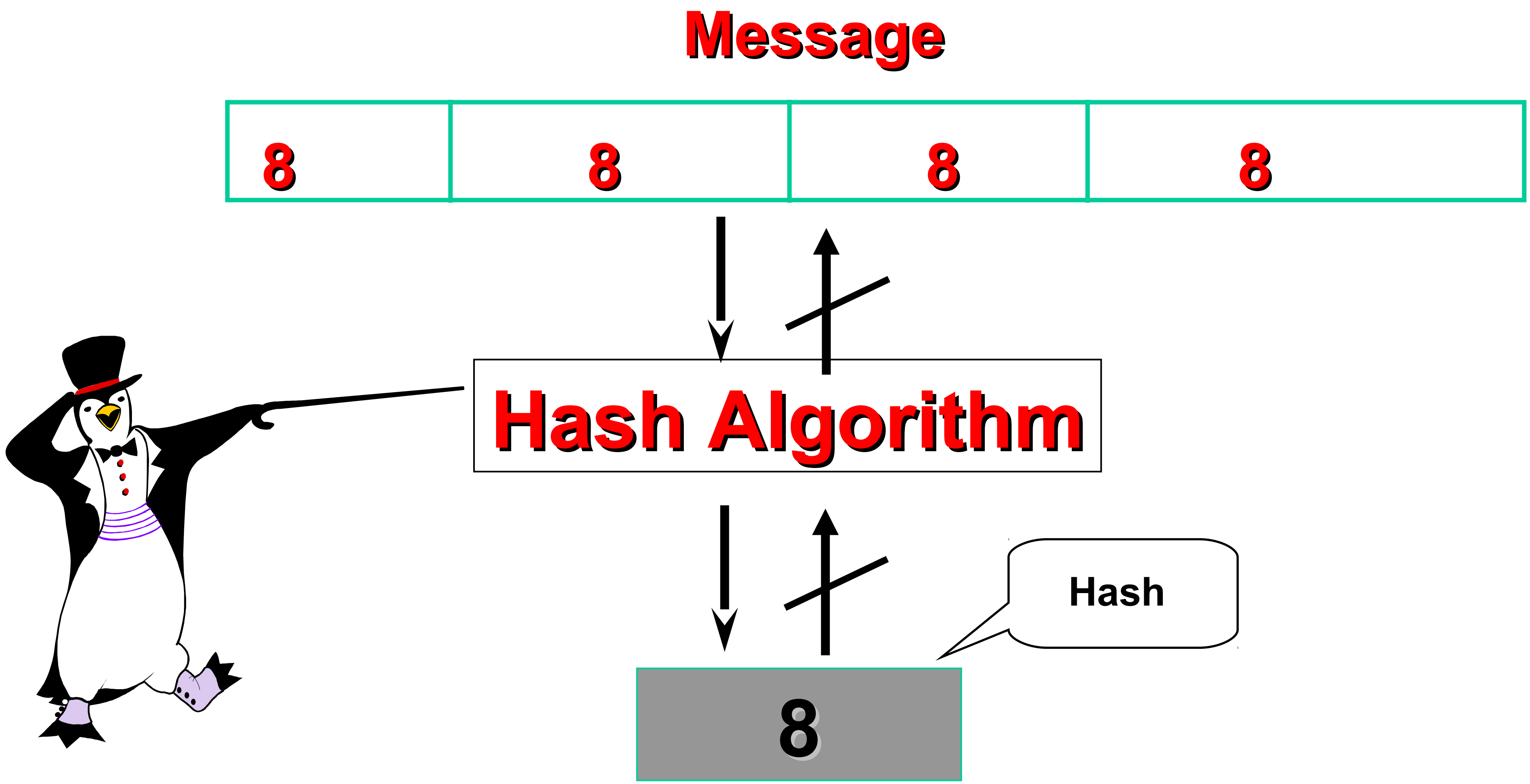
Kasun De Zoysa

Department of Communication and Media Technologies
University of Colombo School of Computing
University of Colombo
Sri Lanka

Hash Functions

- Arbitrary message to fixed size
- Usually assume that the hash function is public and not keyed
  - MAC which is keyed (will discuss soon)
- Hash used to detect changes to message
- Can use in various ways with message
  - most often to create a password, digital signature etc.

Hash Functions



Requirements for Hash Functions

- Can be applied to any sized message M
- Produces fixed-length output h
- Easy to compute h = H(M) for any message M
- Given h, it is infeasible to find x s.t. H(x) = h

  one-way property
- Given x, it is infeasible to find y s.t. H(y) = H(x)

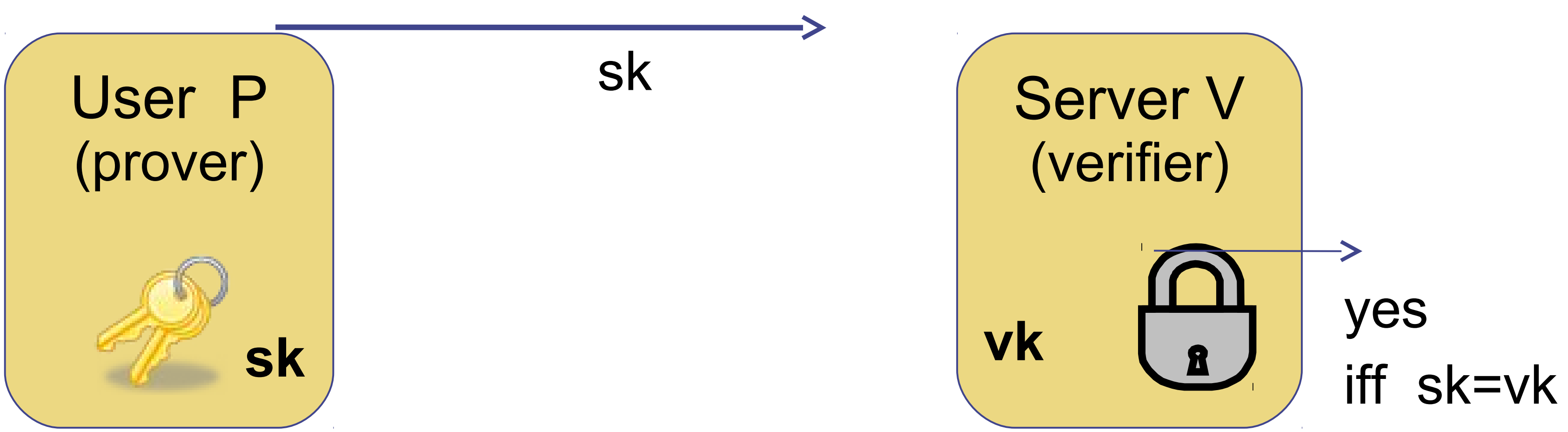
  weak collision resistance
- It is infeasible to find any x,y s.t. H(y) = H(x)

  strong collision resistance

Basic Password Protocol (incorrect version)

- **PWD**: finite set of passwords

- Algorithm G (KeyGen):
  - choose rand pw in PWD. output sk = vk = pw.



Basic Password Protocol (incorrect version)

- <u>Problem</u>: VK must be kept secret
  - Compromise of server exposes all passwords
  - Never store passwords in the clear!

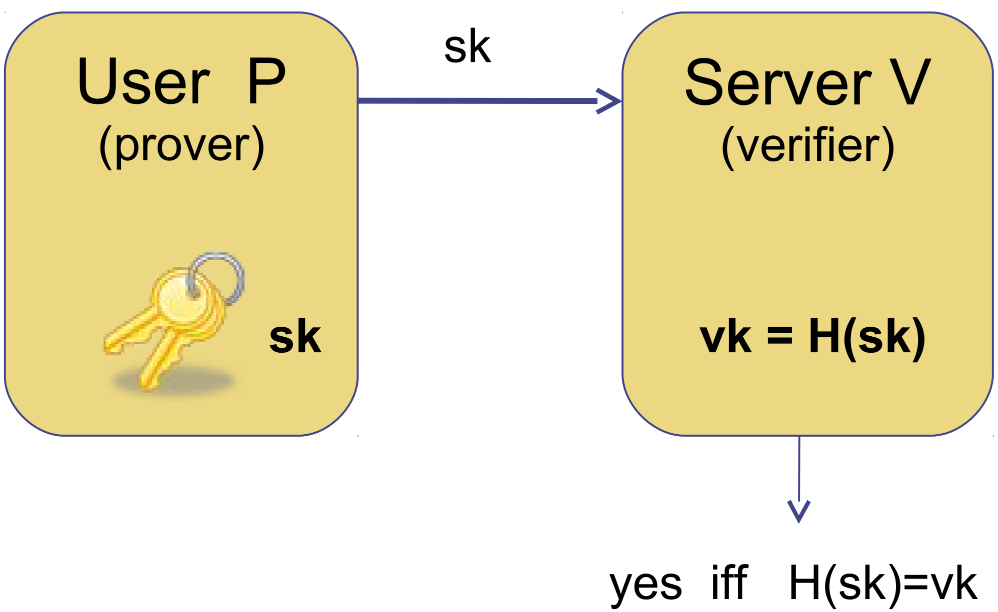
password file on server

Alice	pw_{alice}
Bob	pw_{bob}
…	…

Basic Password Protocol: version 1

H: one-way hash function from PWD to X

"Given H(x) it is difficult to find y such that H(y)=H(x)"



password file on server

Alice	$H(pw_A)$
Bob	$H(pw_B)$
…	…

Weak Passwords and Dictionary Attacks

- People often choose passwords from a small set:
  - The 6 most common passwords (sample of 32×10^6 pwds):
    123456, 12345, Password, iloveyou, princess, abc123
    ('123456' appeared 0.90% of the time)
  - 23% of users choose passwords in a dictionary of size 360,000,000

- **Online dictionary** attacks:
  - Defeated by doubling response time after every failure
  - Harder to block when attacker commands a bot-net

Preventing Dictionary Attacks

- Public salt:
  - When setting password, pick a random n-bit salt S
  - When verifying pw for A, test if **$H(pw, S_A) = h_A$**

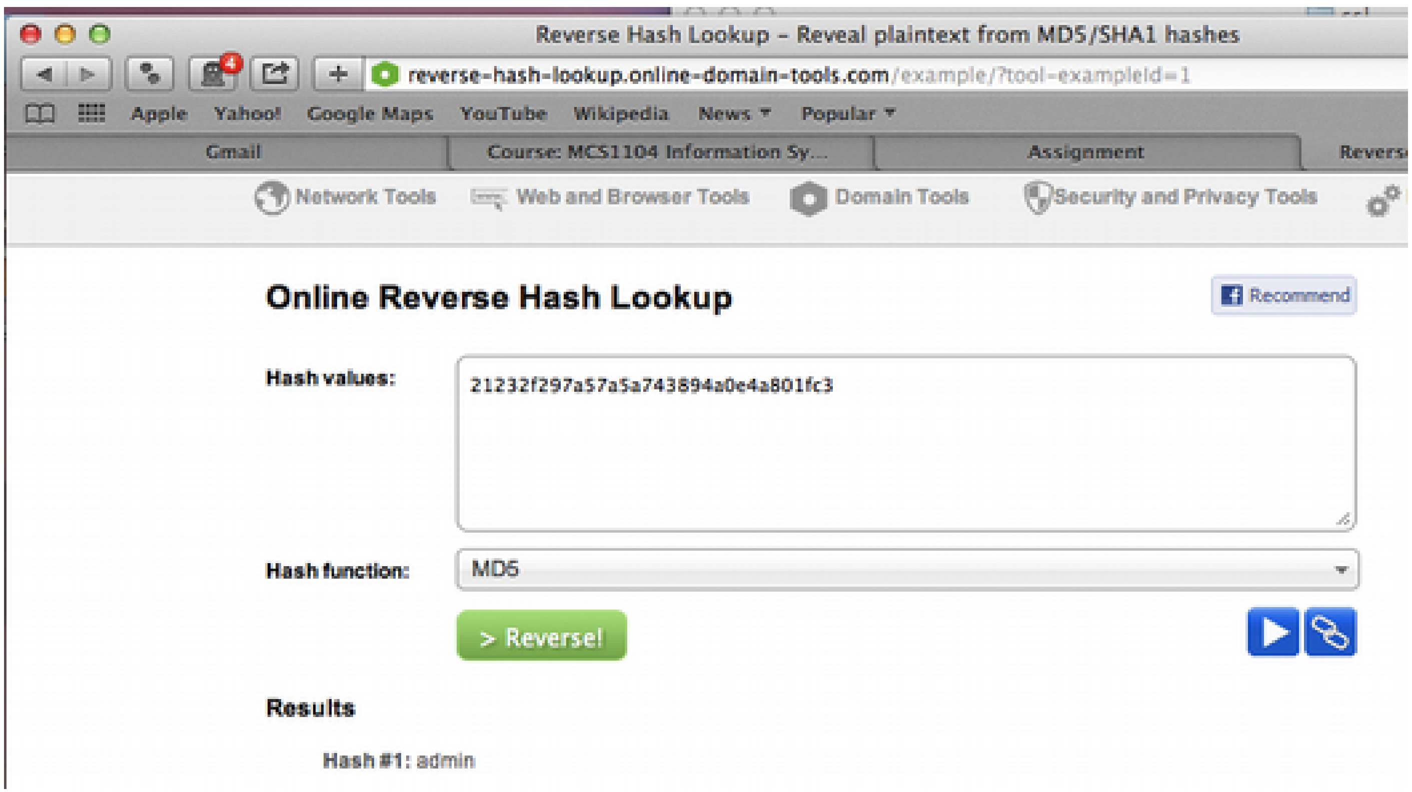
id	S	h
Alice	S_A	$H(pw_A , S_A)$
Bob	S_B	$H(pw_B , S_B)$
…	…	…

- Recommended salt length, n = 64 bits
  - Pre-hashing dictionary does not help

Authenticate the Evidence

- Prove that the evidence is indeed what the criminal left behind.
  - Contrary to what the defense attorney might want the jury to believe, readable text or pictures don’t magically appear at random.
  - Calculate a hash value for the data
    - MD5
    - SHA-1,SHA-256,SHA -512

SHA1 Reverse Lookup



reverse-hash-lookup.online-domain-tools.com

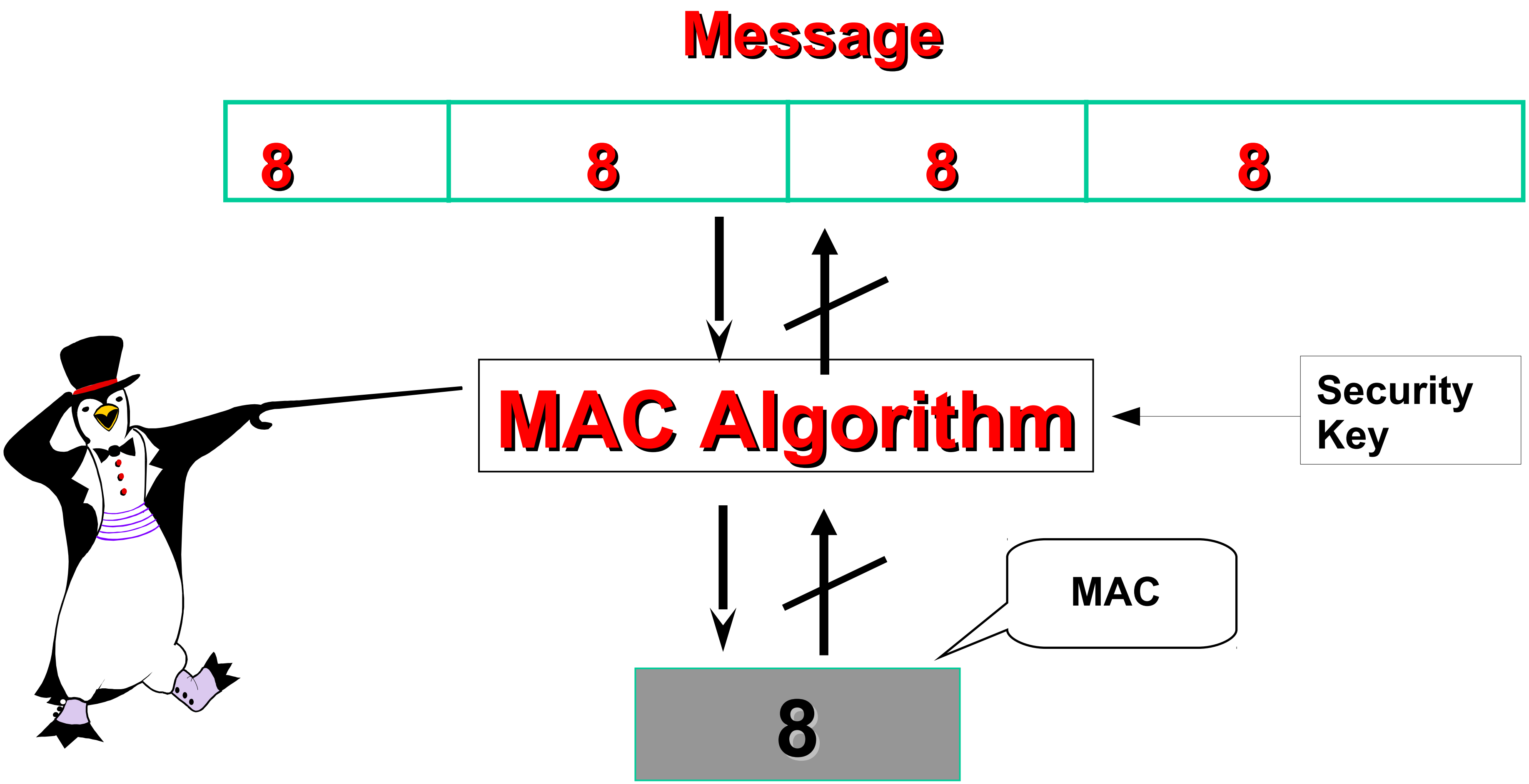
echo -n 'kasun'| md5

Strength of MD5

- MD5 hash is dependent on all message bits
- Rivest claims security is good as can be
- Known attacks are:
  - Berson (92) attacked any 1 round using differential cryptanalysis (but can't extend)
  - Boer & Bosselaers (93) found a pseudo collision (again unable to extend)
  - Dobbertin (96) created collisions on MD compression function (but initial constants prevent exploit)
  - Crypto 2004 attacks on SHA-0 and MD5
- Conclusion is that MD5 has been shown to be vulnerable
- MD5 Collision Demo:

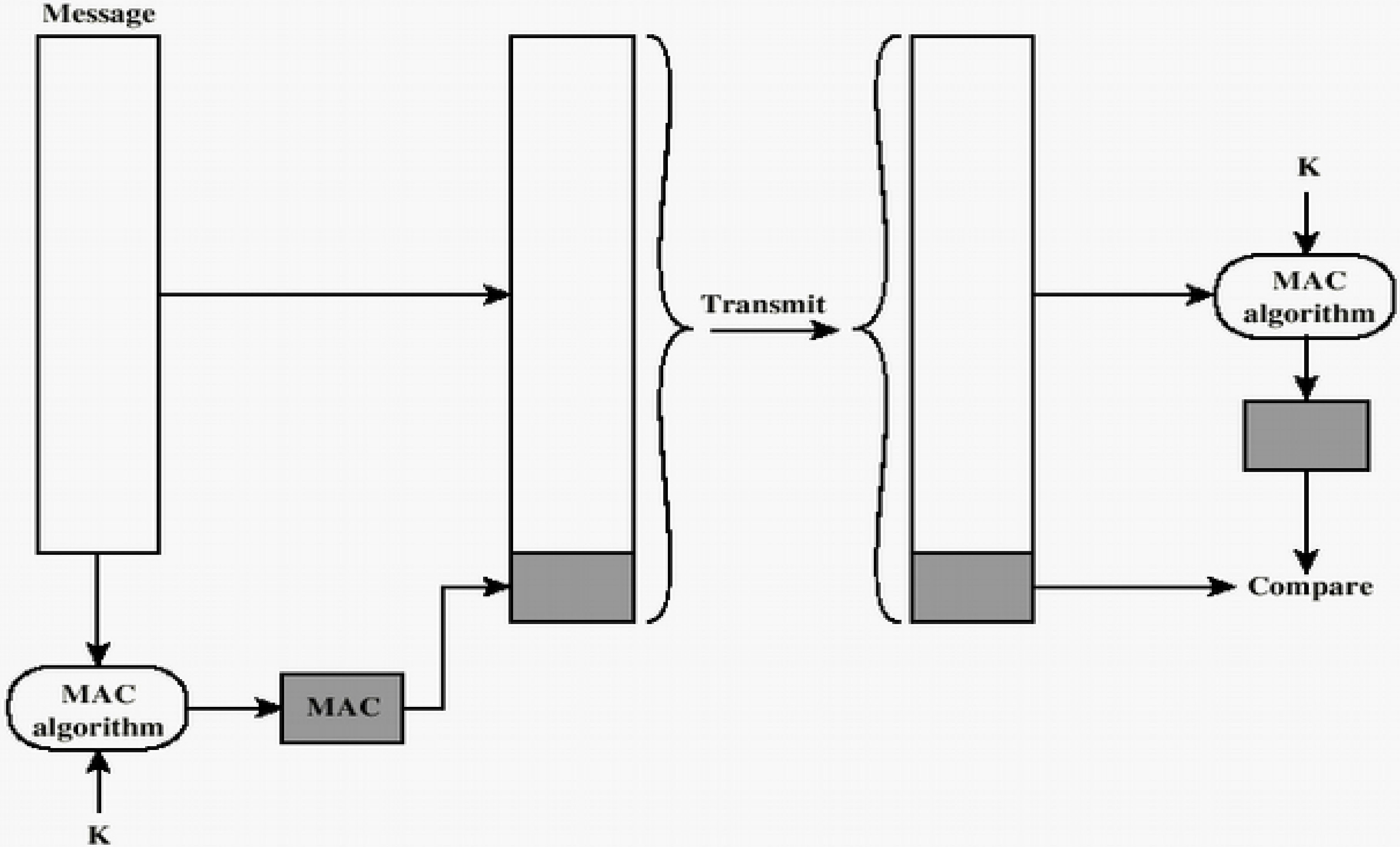
http://www.mscs.dal.ca/~selinger/md5collision/

Message Authentication Code (MAC)



Approaches to Message Authentication

- Authentication Using Conventional Encryption
  - Only the sender and receiver should share a key
- Message Authentication without Message Encryption
  - An authentication tag is generated and appended to each message
- Message Authentication Code
  - Calculate the MAC as a function of the message and the key. MAC = F(K, M)



Message Authentication Using a Message Authentication Code (MAC)

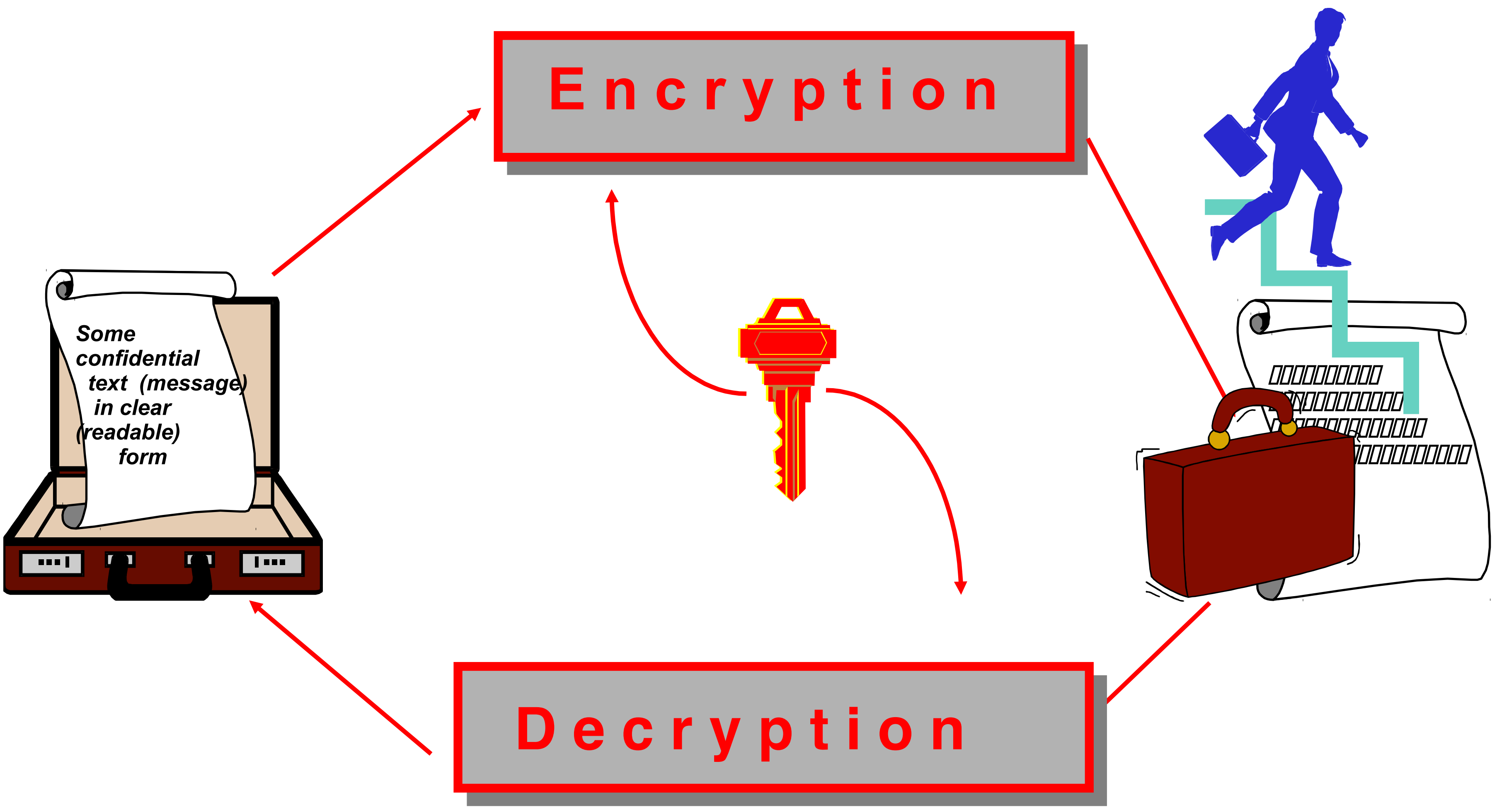
Keyed Hash Functions (HMAC)

- Create a MAC using a hash function rather than a block cipher
  - because hash functions are generally faster
  - not limited by export controls unlike block ciphers
  - Hash includes a key along with the message
- Original proposal:

  KeyedHash = Hash(Key|Message)

  - some weaknesses were found with this
- Eventually led to development of HMAC

Symmetric key Cryptograms



The classic cryptography

- **Encryption algorithm and related key are kept secret.**
- **Breaking the system is hard due to large numbers of possible keys.**
- **For example: for a key 128 bits long**
- **there are**

keys to check $2^{128} \approx 10^{38}$ **using brute force.**

The fundamental difficulty is <u>key distribution</u> to parties who want to exchange messages.

Symmetric Key / Private Key Cryptosystem

- Uses a single Private Key shared between users

- Strengths
  - Speed/ Efficient Algorithms – much quicker than Asymmetric
  - Hard to break when using a large Key Size
  - Ideal for bulk encryption / decryption

- Weaknesses
  - Poor Key Distribution (must be done out of band – ie phone, mail, etc)
  - Poor Key Management / Scalability (each user needs a unique key)
  - Cannot provide authenticity or non-repudiation – only confidentiality

Requirements for Symmetric Key Cryptography

Two requirements for secure use of symmetric encryption:

- a strong encryption algorithm
- a secret key, *K*, known only to sender / receiver

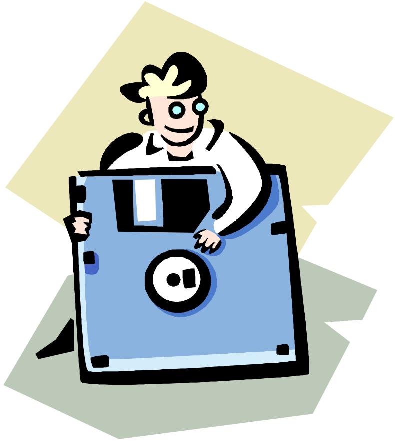
$Y = \mathrm{E}K(X)$

$X = \mathrm{D}K(Y)$

- Assume encryption algorithm is known
- Implies a secure channel to distribute key

Data Encryption Standard (DES)

- Most widely used block cipher in world
- Adopted in 1977 by NBS (now NIST) as FIPS PUB 46
- Encrypts 64-bit data using 56-bit key
- Has widespread use
- Has been the subject of considerable controversy over its security

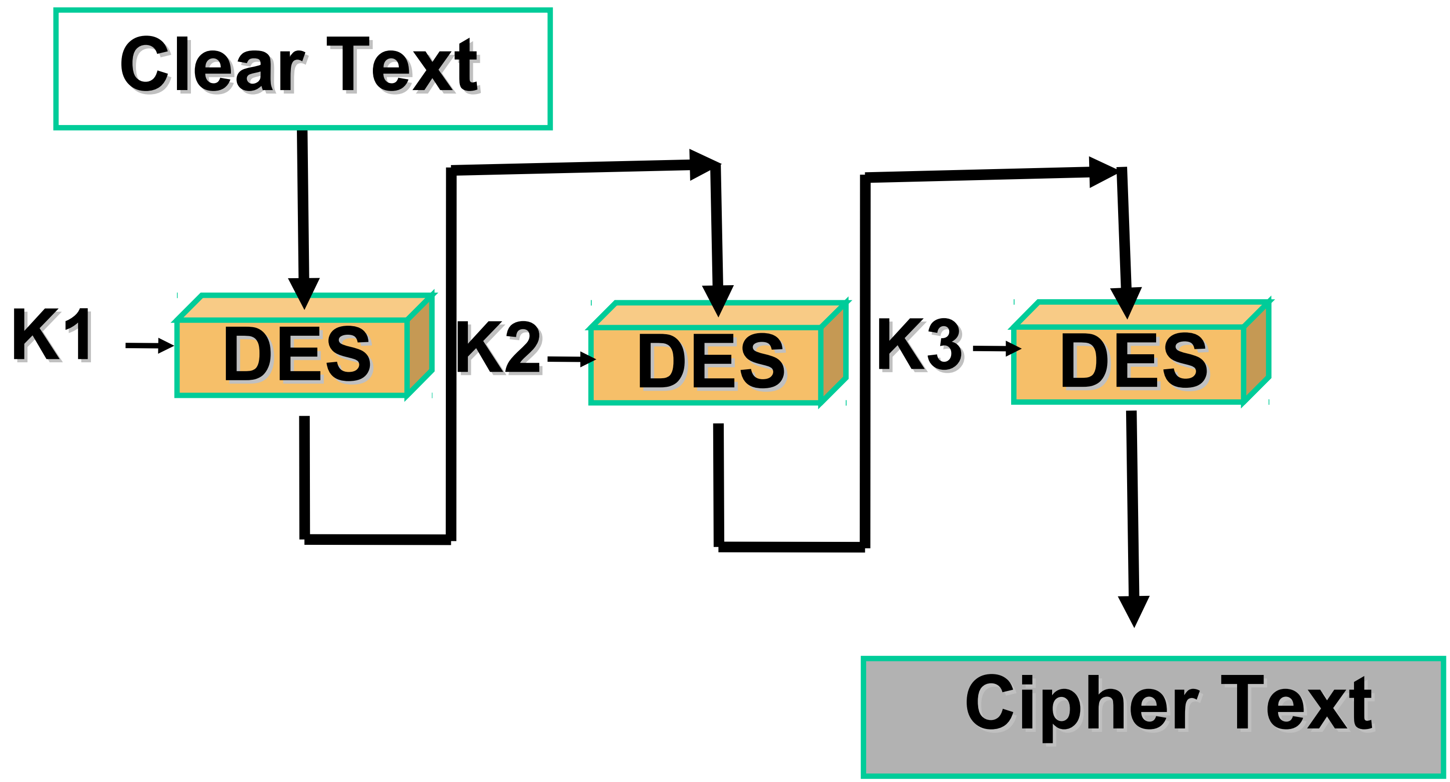


DES – Key Size

- 56-bit keys have 2^{56} = 7.2 x 1016 values
- Brute force search possible
- Recent advances have shown that this is possible
  - in 1997 on Internet in a few months
  - in 1998 on DES Cracker dedicated h/w (EFF) in a less than 3 days (cost: $250,000)
  - in 1999 on Internet in a few hours
    - in 2010 above on Internet in a few minutes

Now we have alternatives to DES

Triple DES



Triple-DES with Two-Keys

- Use 3 encryptions

  would seem to need 3 distinct keys

  But can use 2 keys with E-D-E sequence

C = EK1[DK2[EK1[P]]]

  Note: encrypt & decrypt equivalent in security

  if K1=K2 then can work with single DES

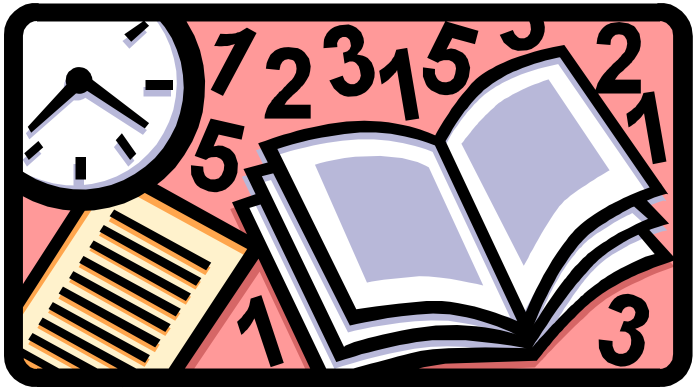
- Standardized in ANSI X9.17 & ISO8732
- No current known practical attacks

DES- AES

- Clearly, a replacement for DES was needed
  - have theoretical attacks that can break it
  - have demonstrated exhaustive key search attacks
- Can use Triple-DES – but slow with small blocks
- NIST issued a call for ciphers in 1997
- 15 candidates accepted in June 1998
- 5 were short listed in August 1999
- Rijndael was selected as the AES in October 2000
- Issued as FIPS PUB 197 standard in November 2001

Advance Encryption Standard (AES)

- In 2001, National Institute of Standards and Technology (NIST) issued AES known as FIPS 197
- AES is based on Rijndael proposed by Joan Daemen, Vincent Rijmen from Belgium



Advance Encryption Standard (AES)

- AES has block length 128
- Supported key lengths are 128, 192 and 256
- AES requires 10 rounds of processing
- Key is expanded into 10 individual keys
- Decryption algorithm uses the expanded keys in reverse order
- Decryption algorithm is not identical to the encryption algorithm

OpenSSL

 # encrypt file.txt to file.enc using 256-bit AES in CBC mode

```
>openssl enc -aes-256-cbc -in file.txt -out file.enc
```

 # decrypt binary file.enc

```
>openssl enc -d -aes-256-cbc -in file.enc
```

 # see the list under the 'Cipher commands' heading

```
>openssl -h
```

Symmetric Key Cryptography

- Traditional **secret/single key** cryptography uses **one** key
- Shared by both sender and receiver
- If this key is disclosed, communications are compromised
- Also is **symmetric**, parties are equal
- Hence receiver can forge a message and claim it was sent by sender

Why Public-Key Cryptography?

- Developed to address two issues:
  - **key distribution** – how to have secure communications in general without having to trust a KDC with your key
- **digital signatures** – how to verify a message comes intact from the claimed sender
- Whitfield Diffie and Martin Hellman in 1976 known earlier in classified community

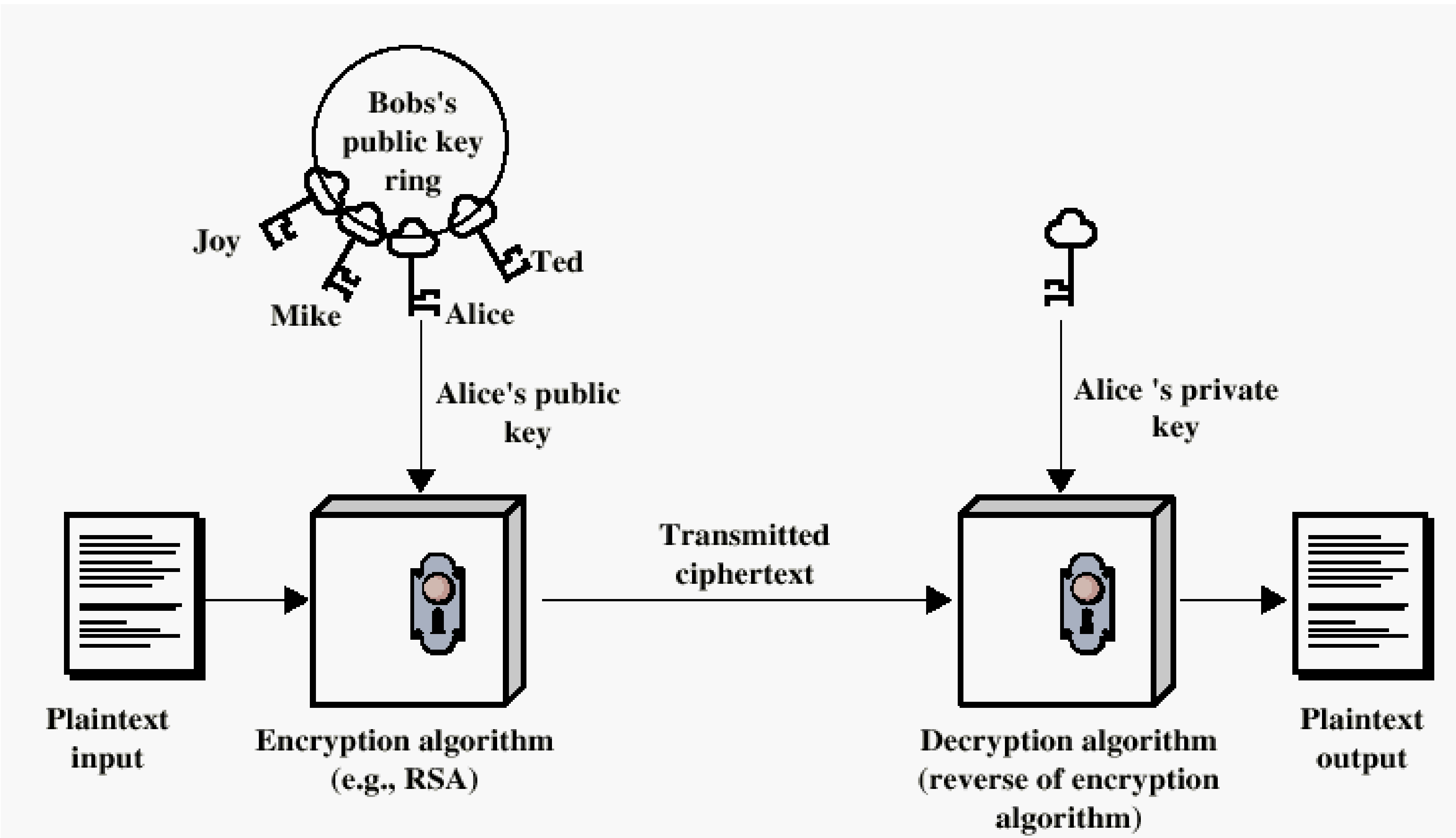
Public-Key Cryptography Principles

- The use of two keys has consequences in: key distribution, confidentiality and authentication.
- The scheme has six ingredients
  - Plaintext
  - Encryption algorithm
  - Public and private key
  - Ciphertext
  - Decryption algorithm

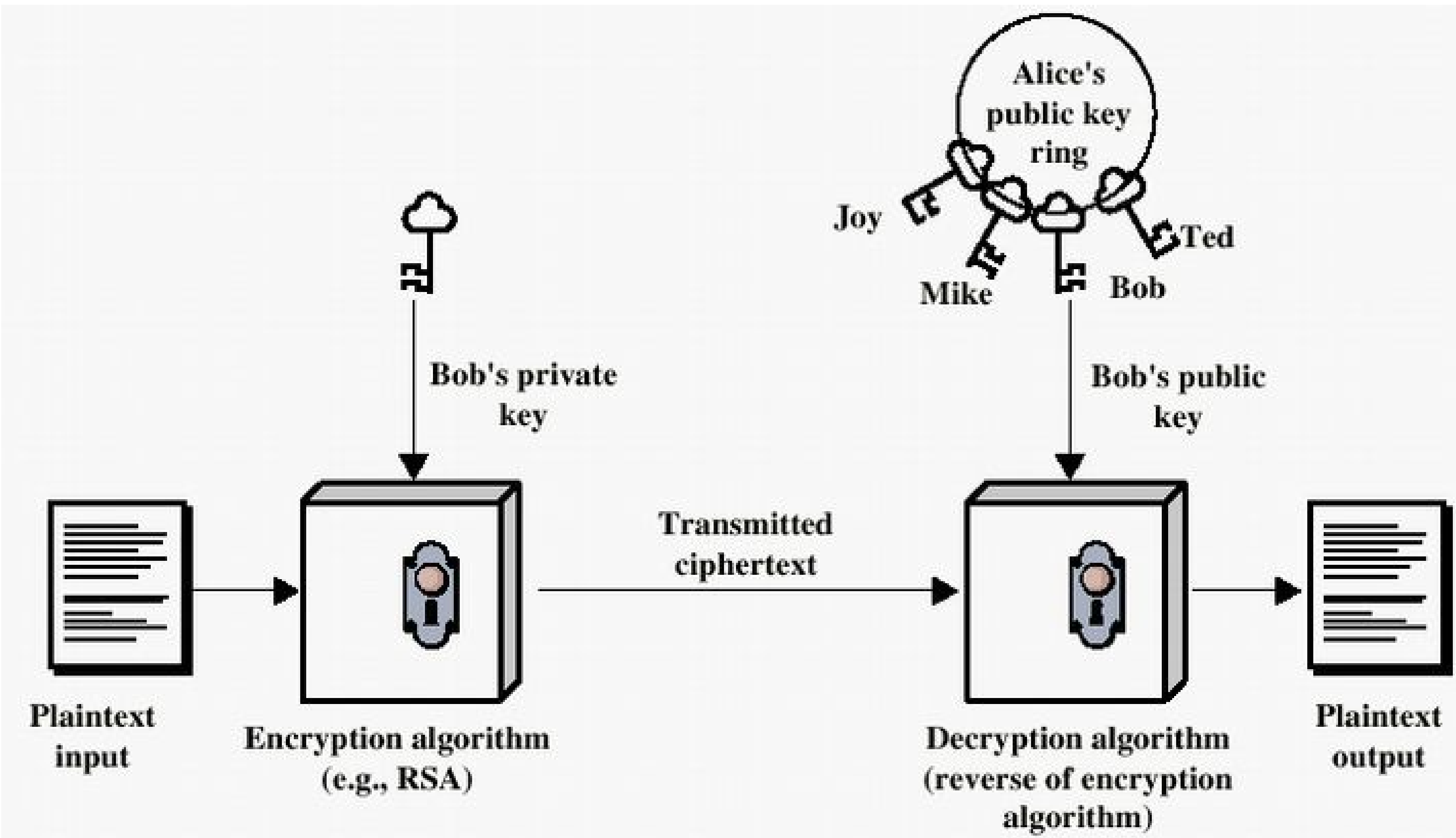
Applications for Public-Key Cryptosystems

- Three categories:
  - **Encryption/decryption:** The sender encrypts a message with the recipient's public key.
  - **Digital signature:** The sender ”signs” a message with its private key.
  - **Key exchange:** Two sides cooperate two exchange a session key.

Encryption using Public-Key system



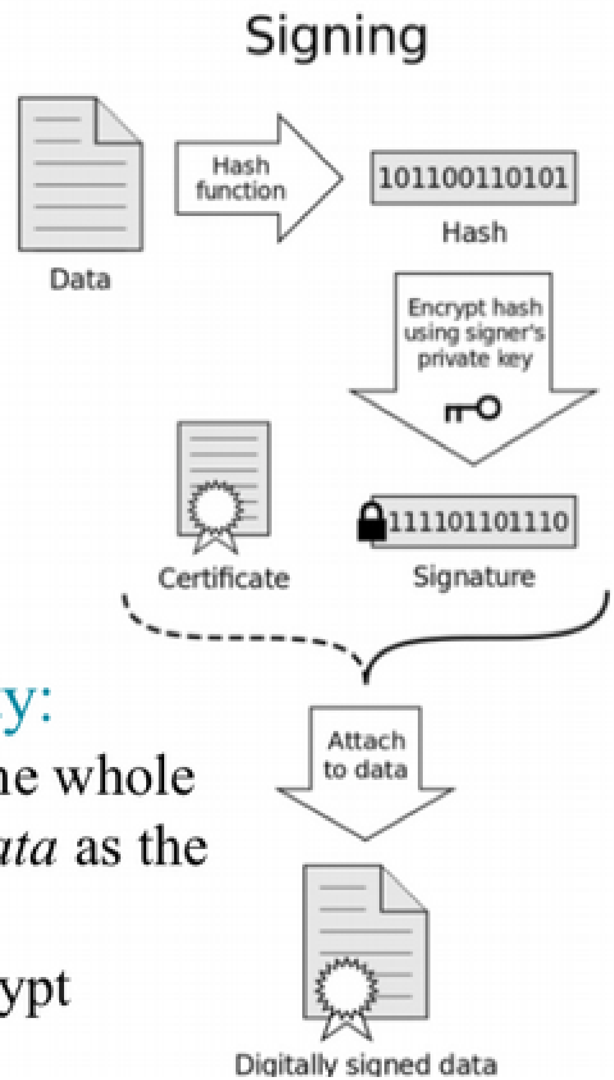
Authentication using Public-Key System

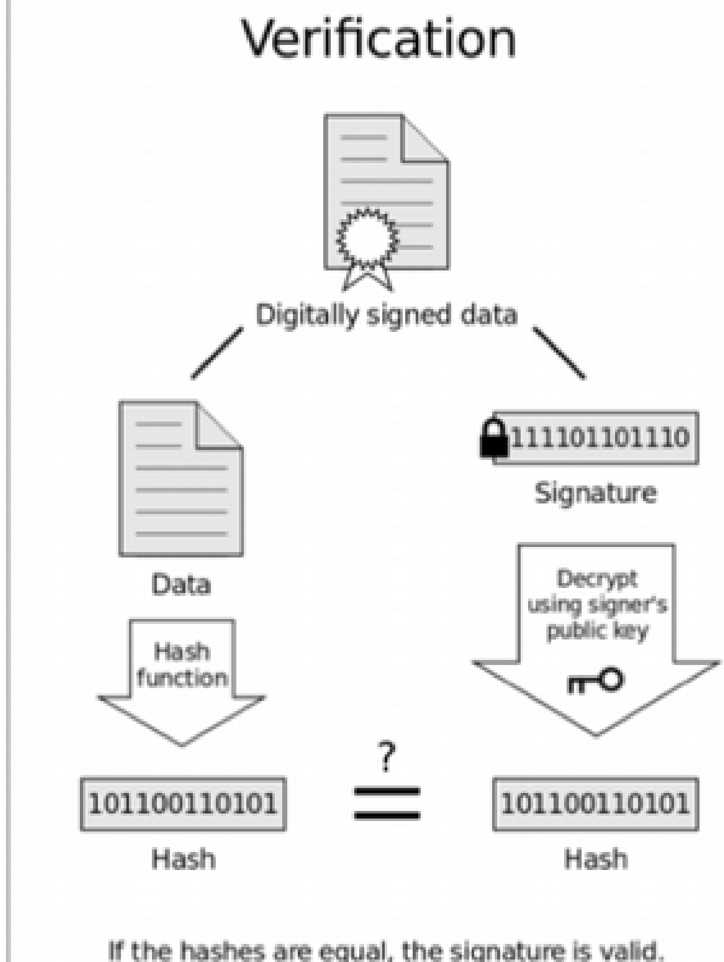


Public-Key Cryptographic Algorithms

- **RSA** - Ron Rives, Adi Shamir and Len Adleman at MIT, in 1977.
  - RSA is a block cipher
  - The most widely implemented
- **Diffie-Hellman**
  - Exchange a secret key securely
  - Compute discrete logarithms
- **Elliptic Curve Cryptography (ECC)**

Typical Digital Signature





For confidentiality:

- Need to encrypt the whole *digitally signed data* as the plaintext.
- Four encrypt/decrypt operations!

Signature Creation

- **Generate Public/Private key pair**

```
openssl genrsa -out mykey.pem
openssl rsa -in mykey.pem -pubout >mypub.pem
```

- **Create the signature**

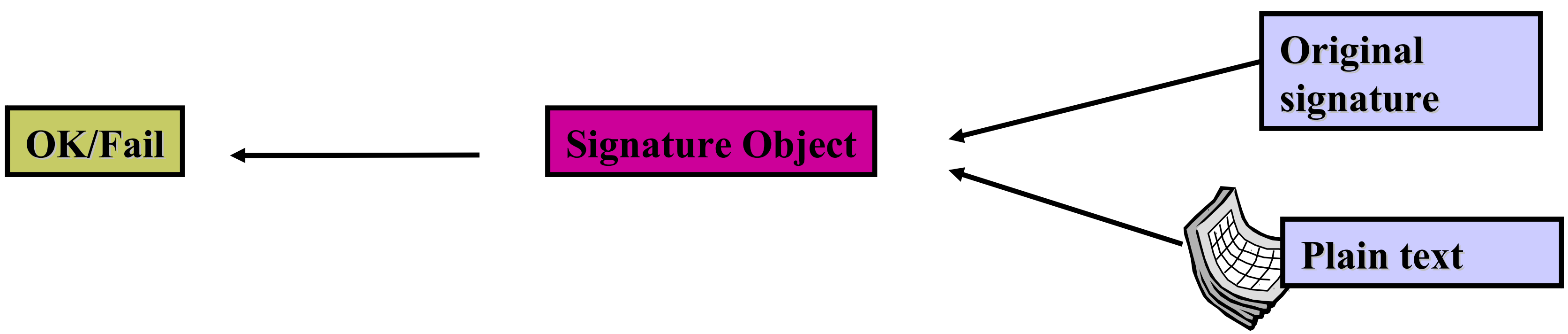
```
openssl dgst -sha1 -sign mykey.pem
-out mysign.sha1 jethavanaya.jpg
```

Signature Verification

- Retrieves the Public key

- Verify the signature

```
openssl dgst -sha1 -verify mypub.pem
-signature mysign.sha1 jethavanaya.jpg
```



Key measure: Encryption strength

The mathematic background of ECC is more complex than other cryptographic systems

- Geometry, abstract algebra, number theory

ECC provides greater security and more efficient performance than the first generation public key techniques (RSA and Diffie-Hellman)

- Mobile systems
- Systems required high security level (such as 256 bit AES)

Bits of Security	Symmetric Key Algorithm	Corresponding RSA Key Size	Corresponding ECC Key Size
80	Triple DES (2 keys)	1024	160
112	Triple DES (3 keys)	2048	224
128	AES-128	3072	256
192	AES-192	7680	384
256	AES-256	15360	512

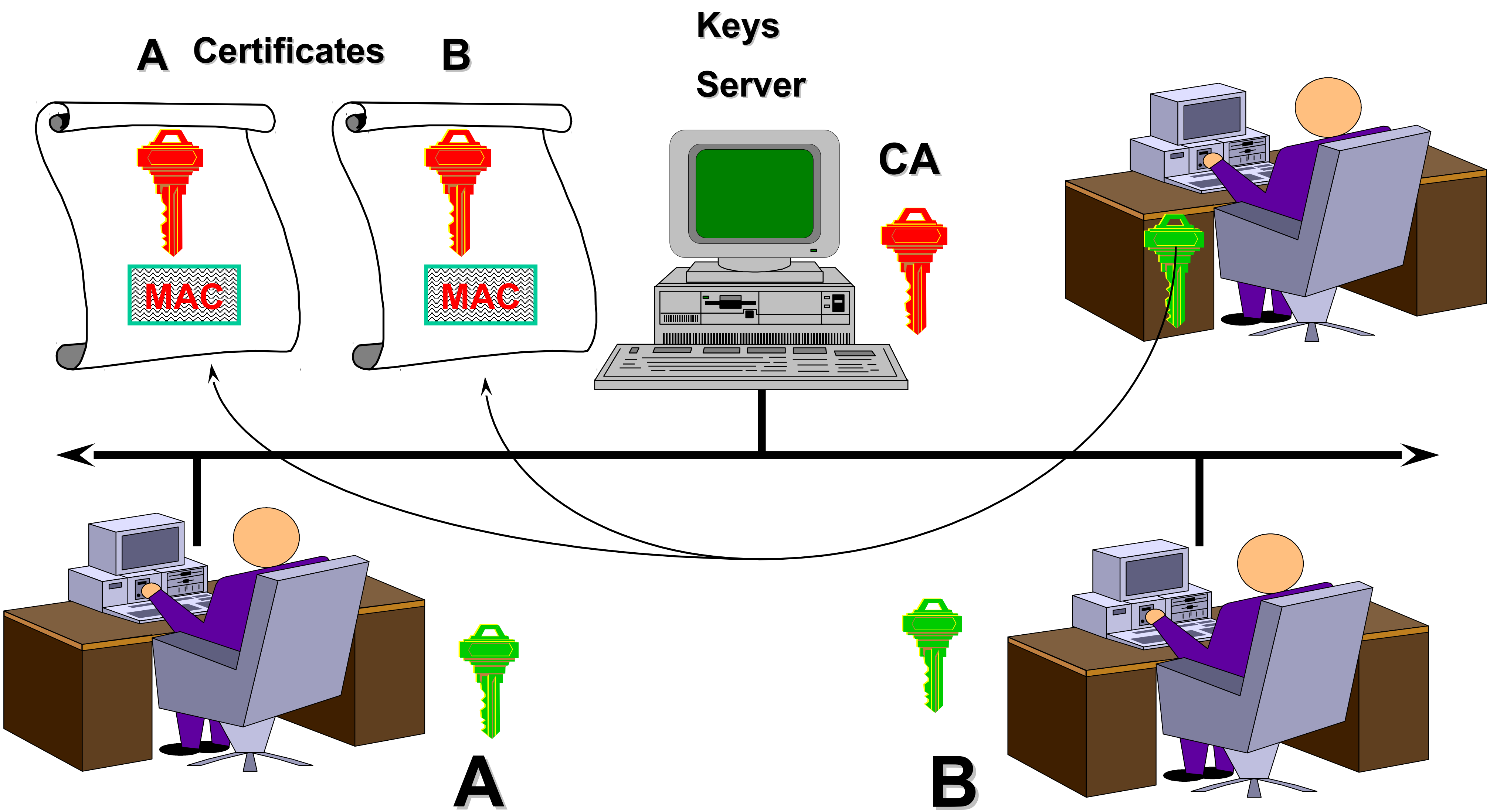
Hybrid Encryption

- Why is symmetric key encryption still used?
  - Performance
  - Also cryptographic reasons

In practice one uses hybrid encryption...

  - A one-time random key is generated (“session key”)
  - This is used to symmetrically encrypt the message
  - The symmetric session key is encrypted through public key encryption and sent to the other party together with the (encrypted) message

Certificate Authority



Certificates Infrastructure

Certificate Authority

- Trusted, 3rd party organization
- CA (Certificate Authority) guarantees that the individual granted a certificate is who he/she claims to be
- CA usually has arrangement with financial institution to confirm identity
- Critical to data security and electronic commerce
- Well known organisation establish themselves to act as certificate authorities. Verisign, CREN, etc.
- One can then obtain X.509 public key certificates from them by submitting satisfactory evidence of their identity.

Certificate Standards

X.509

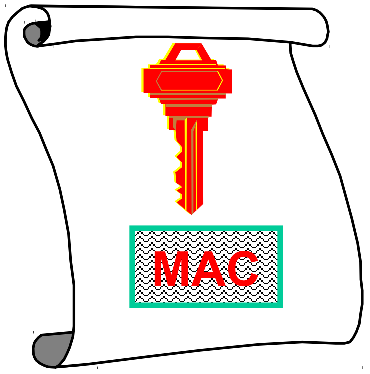
- Most widely used standard for certificates.
- Part of the X.500 standard for the construction of global directories of names and attributes.
- X.509 is used in cryptography as a format definition for free standing certificates.
- Public key is bound to a named entity called a subject.
- Binding is in the signature, which is issued by an Issuer.

X.509 Certificate Format

Subject:	Distinguished Name, Public Key
Issuer:	Distinguished Name, Signature
Validity Period:	Not Before, Not After
Admin Info:	Version, Serial
Extended Info: …	

Internal Structure of Certificate

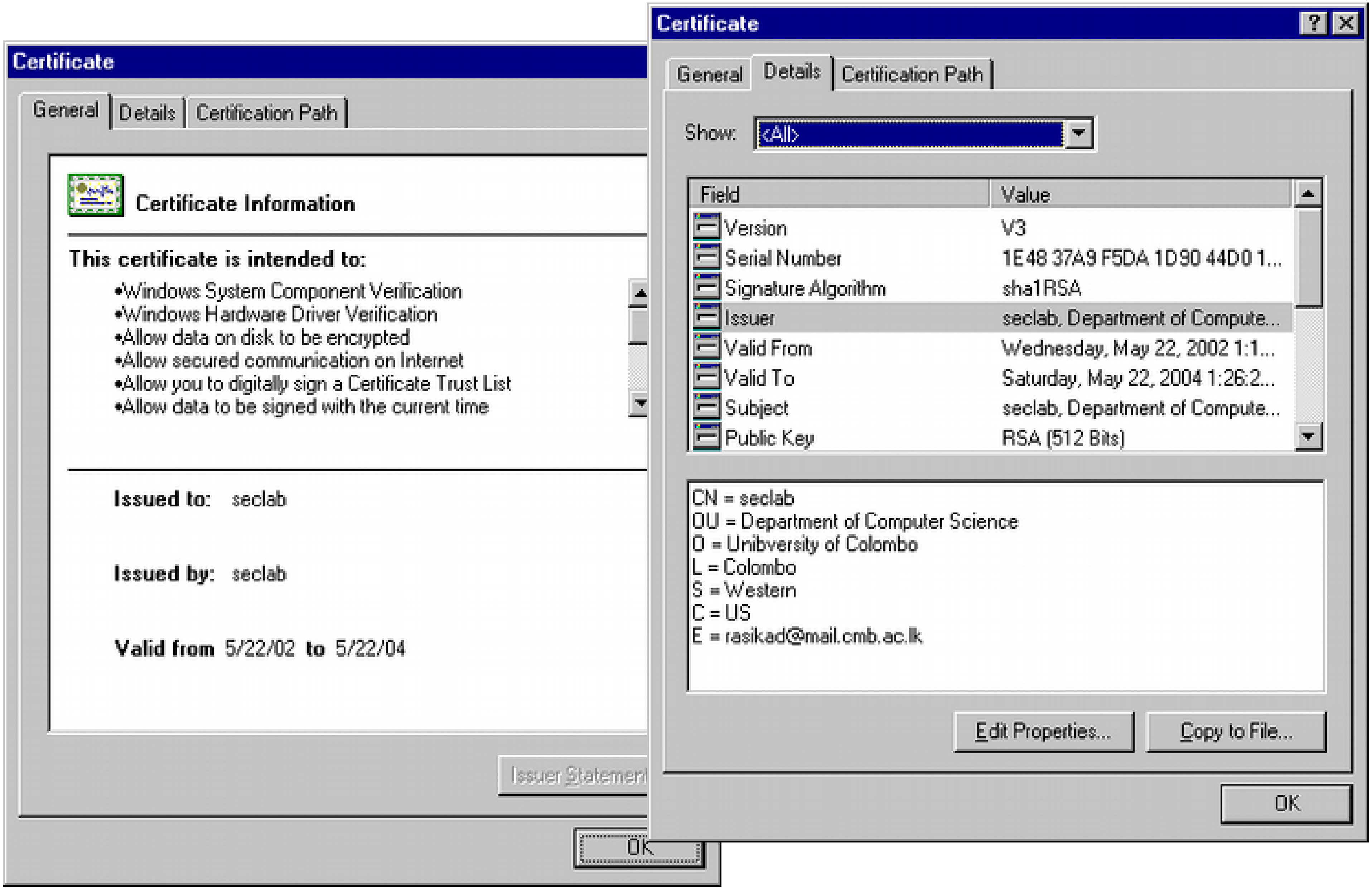
- Version
- Serial Number
- Signature Algorithm
- Issuer
- Subject
- Validity
- Subject Public Key Information
- Extensions
- Signature



Structure of Distinguish Name

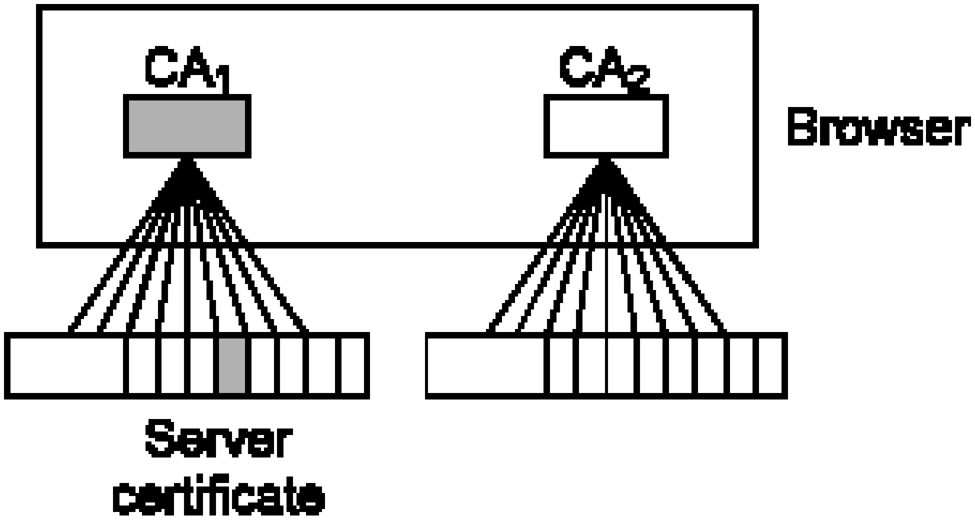
- Country Name
- State and Province Name
- Locality Name
- Organization Name
- Organization Unit Name
- Common Name
- Email Address
- URL

Root Certificate



CA Hierarchy in Practice

Flat or Clayton's hierarchy



CA certificates are hard-coded into web browsers or email software

- Later software added the ability to add new CAs to the hardcoded initial set

Dr. Kasun De Zoysa
e-mail: kasun@ucsc.cmb.ac.lk