SEENG1050

Web Technologies

Lecture 11: Encryption

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

FACULTY OF ENGINEERING AND BUILT ENVIRONMENT

www.newcastle.edu.au

Lecture Overview

- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

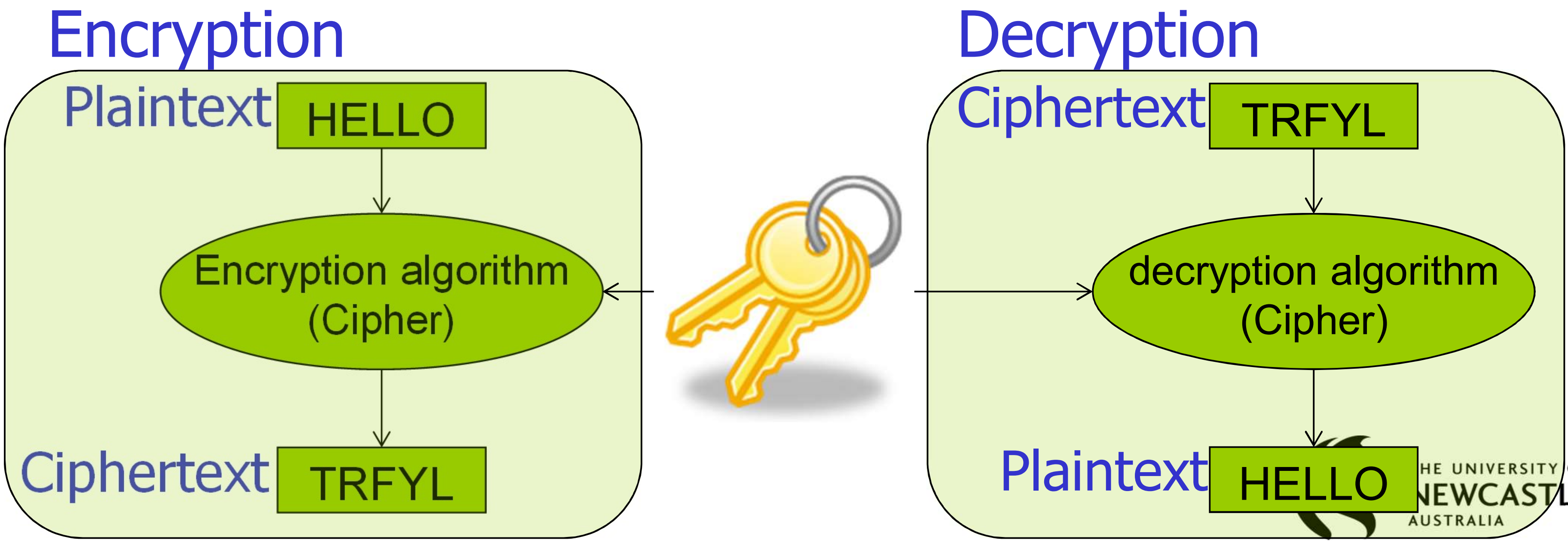
Protecting Internet Communications: Encryption

- Encryption: The process of transforming plain text or data into cipher text that cannot be read by anyone other than the sender and receiver
- Purpose: Secure stored information and information transmission
- To encrypt or decrypt, you need

  an algorithm + a key

Cryptography

- Plaintext - a message
- Ciphertext - an encrypted message
- Encryption - plaintext → ciphertext
- Decryption - ciphertext → plaintext
- Cipher - encryption algorithm



Encryption and the Internet

- Not an early priority for Internet developers
  - Most protocols work on plaintext messages
  - Developed mostly through free work at the University level
  - Open design, sharing and public information were the driving forces for the Internet

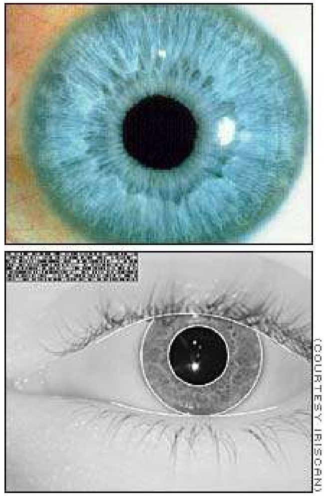
Encryption and the Internet

- Priorities have changed in the last 10 years
  - More company, government and private (i.e., secret) information being transmitted
- Not a new problem – Military networks have long been used for secret information
  - Usually isolated on a separate network, shielded from the Internet

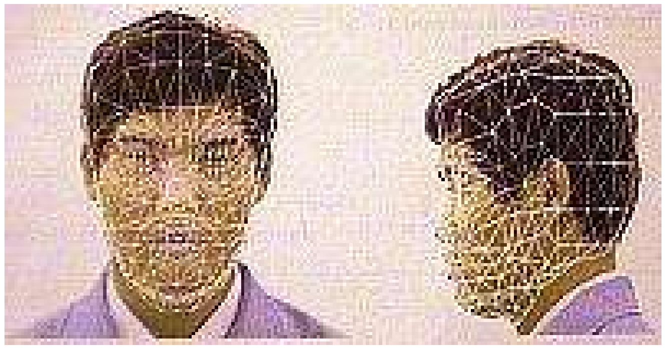
3 Fundamental Reasons for Cryptography

1. Authentication

- The receiver of the message can ascertain and validate its origin. Such schemes based on:
  - Something you “know” (password)
  - Something you “have” (swipe card, *Secure*Id or other device)
  - Something you “are” (voiceprint, fingerprint, retinal scan – biometrics, face detection)



(COURTESY IRISCAN)



3 Fundamental Reasons for Cryptography

2. Integrity

- The receiver can verify that the message was not modified during the transmission

3. Non-repudiation

- The sender cannot deny that they sent the message

3? Fundamental Reasons for Cryptography

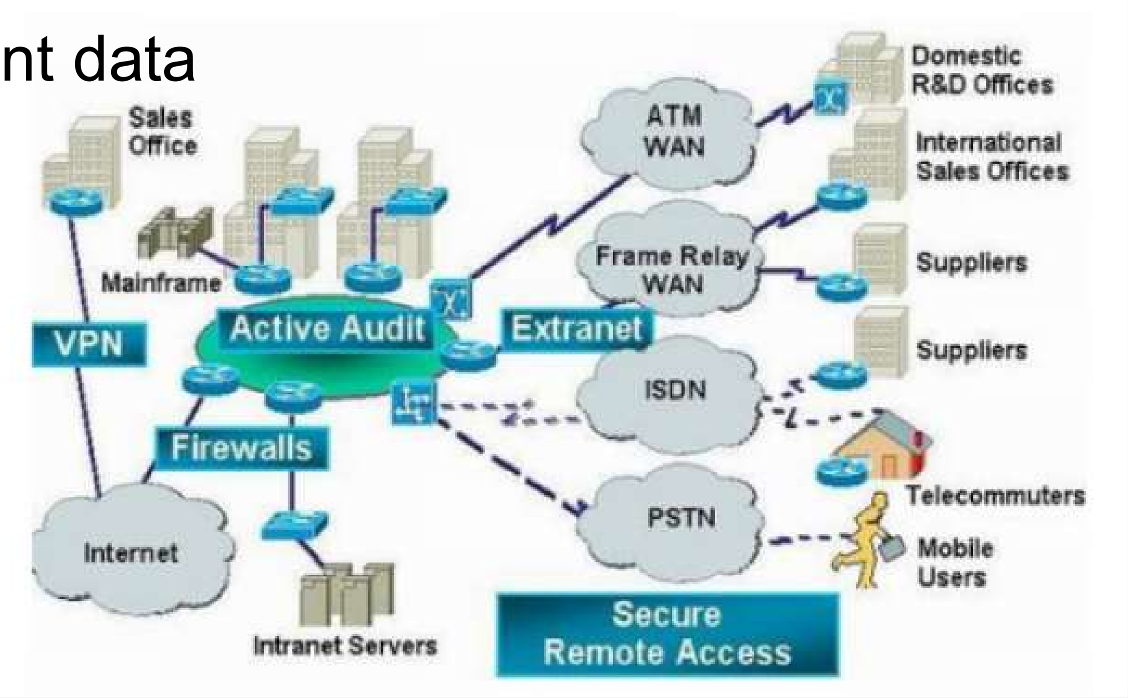
…okay…

4. Confidentiality

- A message can be sent without “eavesdroppers” being able to read it

Who Needs Cryptography?

- Electronic Funds Transfer – between banks
- Electronic Funds Transfer – EFTPOS, ATMs
- Credit card number across the web
- Bank records and other financial data
- Business/Government communications
- Product development data
- Your personal files
- Student grades



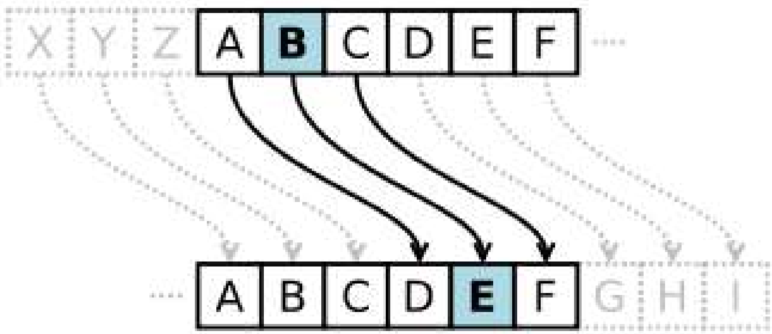
A VERY Simple Example

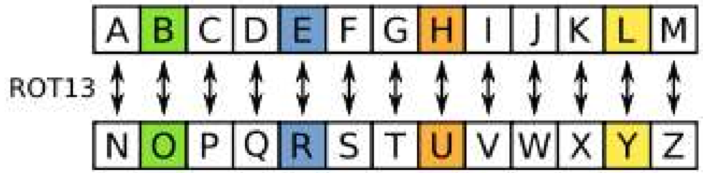
- Plaintext: `This is a secret sentence`
- Ciphertext: `Uijt jt b tfdsfu tfoufodf`
- This is a very basic transposition of letters
  - a→b, b→c, c→d, …
- The "key" is the number of letters to transpose
  - Caesar cipher used a shift of 3
- What is the basic problem with this method?
  - A very small set of permutations available – only 26
  - Only have to "try" 26 keys to find the correct one

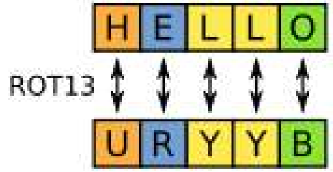
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

A VERY Simple Example

- Plaintext: `This is a secret sentence`
- Ciphertext: `Uijt jt b tfdsfu tfoufodf`
-
-







- What is the basic problem w
  - A very small set of permutations available – only 26
  - Only ... “try” 26 keys to find the ... one

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

A VERY Simple Example

- Plaintext: `This is a secret sentence`
- Ciphertext: `Uijt jt b tfdsfu tfoufodf`
- This is a very basic transposition of letters
  - a→b, b→c, c→d, …
- The "key" is the number of letters to transpose
  - Caesar cipher used a shift of 3
- What is the basic problem with this method?
  - A very small set of permutations available – only 26
  - Only have to "try" 26 keys to find the correct one

Mono-alphabetic Substitution

- Where a letter of plaintext always produces the same letter of ciphertext
- Plaintext: `a b c d e f g h i j k l m`
- Ciphertext: `Q R S K O W E I P L T U Y`
- There are 26! different cipher alphabets! Far too many to try them all
- But it is still not secure. Why?
  - The letter frequencies and underlying patterns are unchanged
    - e is the most common letter, now O will be
    - the is a very common word, …

Mono-alphabetic Substitution

Ftt bdv bfhvq efno ukvj f tnmvbnxv ic bdv fkvjfyv Fxvjnlfz fjv qevzb ic bdv yukvjzxvzb nz tvqq bdfz f qvluzo.

Pnx Mnviny V, F, N, B, Z – E, T, A, I, N, bdv – the

```
    the t  e          e       et  e     the
Ftt bdv bfhvq efno ukvj f tnmvbnxv ic bdv

  e   e   e         e   e t     the    e   e t
fkvjfyv Fxvjnlfz fjv qevzb ic bdv yukvjzxvzb

    e   th       e
nz tvqq bdfz f qvluzo.

                                               e
                                     Pnx Mnviny
```

```
 All the ta e    a      e  a l  et   e     the
Ftt bdv bfhvq efno ukvj f tnmvbnxv ic bdv

a e a e A e   an a e   ent     the     e n ent
fkvjfyv Fxvjnlfz fjv qevzb ic bdv yukvjzxvzb

 n le   than a   e  n
nz tvqq bdfz f qvluzo.

                                                 e
                                        Pnx Mnviny
```

Transposition Ciphers

Cryptography and Network Security Principles and Practice, 2nd Edition – by William Stallings

4	3	1	2	5	6	7
A	T	T	A	C	K	P
O	S	T	P	O	N	E
D	U	N	T	I	L	T
W	O	A	M	X	Y	Z

- It is fairly straightforward to break
- Cipher text needs to be placed in a matrix and play around with the column positions
- More than one stage of transposition can make it significantly difficult

Plain text: attack postponed until two am

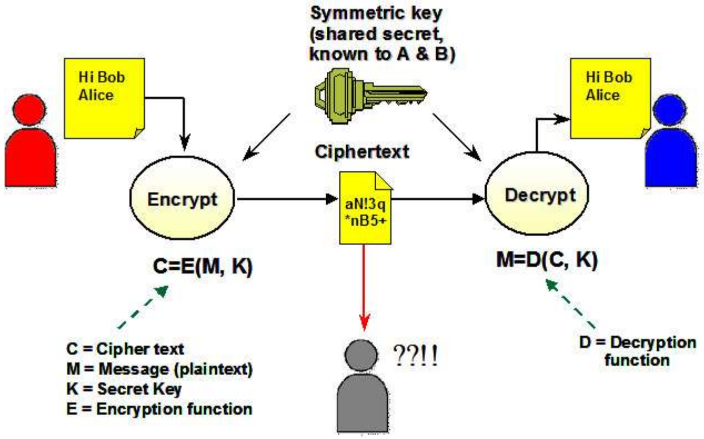
Cipher text: ttnaaptmtsuoaodwcoixknlypetz

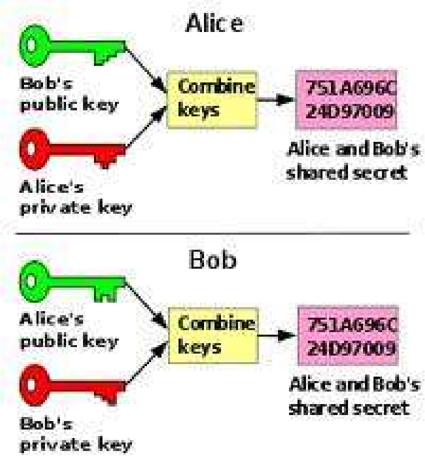
Lecture Overview

- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

Secret key vs Public key

- Secret key: symmetric cryptography, the same key is used for both encryption and decryption.
  - Data Encryption Standard (DES)
- Pubic key: each user has a *public key* and a *private key.*
  - *RSA* (Rivest, Shamir, Adleman)





Lecture Overview

- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

Secret Key Encryption

- Both sender and receiver share a common secret – the key
- Simple example: XOR

MESSAGE:	00111010010010
KEY:	01110110100101
CIPHERTEXT:	

Sender does this XOR and sends the *ciphertext*

CIPHERTEXT:	01001100110111
KEY:	01110110100101
MESSAGE:	

Receiver does this XOR and reads the *plaintext*

- Note: XOR is its own inverse

Secret Key Encryption

- If the message is longer than the key?
  - One solution: just repeat the key
- DES (Data Encryption Standard)
  - Uses both transposition and substitution ciphers
  - Encrypts 64-bit blocks of data with 56-bit key
- IDEA (International Data data with 128-bit key
  - Used in PGP program (Pretty Good Privacy)

How do you send the key?

Lecture Overview

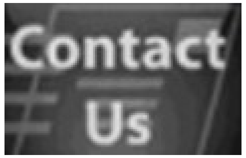
- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

Key management: Communication Example

- Alice wants to send Bob secret messages via the Internet, but knows that Lucy is listening
- One strategy:
  - Alice encrypts message *m* into ciphertext *c* using key *k*, and sends *c* to Bob
  - Bob decrypts *c* into the original plaintext *m* using key *k*
  - Even if Lucy intercepts *c*, she still needs key *k* to read the message
- Drawbacks:
  - Alice and Bob must agree on the key *k* to use

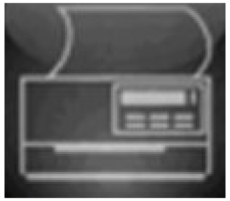
Key Management

- How should Alice send a key to Bob?
  - Should she email it?
    - Is email secure?



  - Should she phone him with it?
    - Is the phone line secure?
    - How complex is this key?
    - Will he be there when she calls?

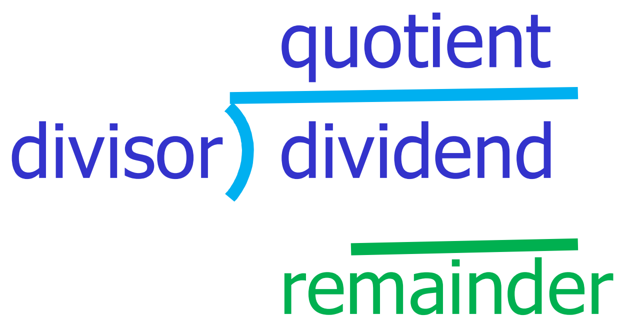
  - Should she fax it?
  - Should she post it?



- All methods of key transfer are open to interception – some are more secure than others

Key Management: Modulo

- Mathematical operator
  - Write as MOD (C, C++, Java uses %)
  - Is the "remainder" after an integer division
- 5/2 ('normal') = 2.5, 5/2 (integer) = 2
- 5 MOD 2 = 1, 15 MOD 5 = 0
- 23 MOD 6 = 5, 13 MOD 6 = 1



Key Management

- Key-exchange protocol – Diffie-Hellman algorithm

1. Alice and Bob agree on a large prime number n and another number g – *these are not necessarily secret*
2. Alice generates a random number x – *which is secret* –and sends to Bob the value $X = (g^x) \bmod n$
3. Bob generates a random number y – *which is secret* – and sends to Alice the value $Y = (g^y) \bmod n$
4. Alice receives Y and computes $K_x = (Y^x) \bmod n$
5. Bob receives X and computes $K_y = (X^y) \bmod n$

K_x is equal to K_y

Key Management

- Key-exchange protocol – Diffie-Hellman algorithm

1. Alice and Bob agree on a large prime number n and another number g – *these are not necessarily secret*

   $n=23,\ g=5$

2. Alice generates a random number x – *which is*

secret –and sends to Bob the value $X = g^x \bmod n$

   $x = 6,\ X = 5^6\ mod\ 23 = 15{,}625\ mod\ 23 = \mathbf{8}$

3. Bob generates a random number y – *which is secret*

– and sends to Alice the value $Y = (g^y) \bmod n$

   $y = 15,\ Y = 5^{15}\ mod\ 23 = 30{,}517{,}578{,}125\ mod\ 23 = \mathbf{19}$

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Key Management

- Key-exchange protocol – Diffie-Hellman algorithm

1. Alice and Bob agree on a large prime number n and another number g – *these are not necessarily secret*

   $n=23,\ g=5$

2. Alice generates a random number x – *which is secret* –and sends to Bob the value $X = g^x \bmod n$

   $x = 6,\ X = 5^6 \bmod 23 = 15{,}625 \bmod 23 = 8$

3. Bob generates a random number y – *which is secret* – and sends to Alice the value $Y = (g^y) \bmod n$

   $y = 15,\ Y = 5^{15} \bmod 23 = 30{,}517{,}578{,}125 \bmod 23 = 19$

4. Alice receives Y and computes $K_x = (Y^x) \bmod n$

$$\boldsymbol{Kx} = 19^6 \bmod 23 = 47{,}045{,}881 \bmod 23 = \mathbf{2}$$

5. Bob receives X and computes $K_y = (X^y) \bmod n$

$$\boldsymbol{Ky} = 8^{15} \bmod 23 = 35{,}184{,}372{,}088{,}832 \bmod 23 = \mathbf{2}$$

K_x is equal to K_y

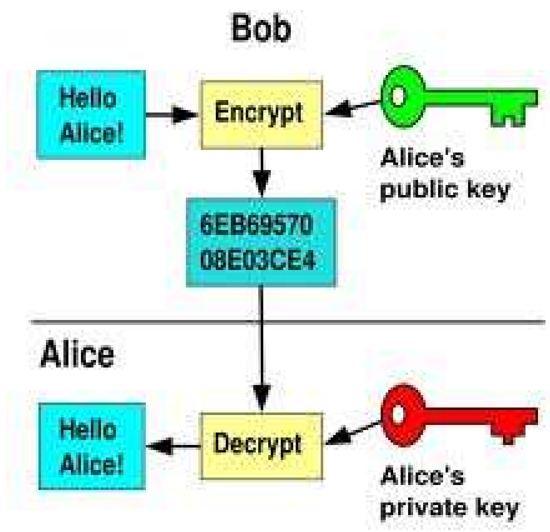
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Lecture Overview

- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

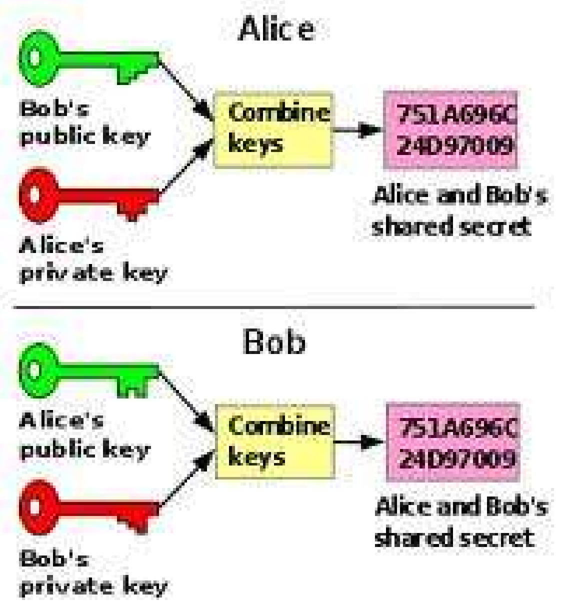
Public-key Encryption

- Each user has a pair of keys
  - Public key – public knowledge – used for encryption
  - Private key – known only to its owner –used for decryption and signing
- Bob wants to send a secure message to Alice – he encrypts it with Alice's public key
- Alice receives the message and decrypts it using her private key
- Each key is one way
- eg) RSA Cryptographic



RSA Cryptographic System

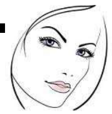
- Rivest, Shamir, Adleman (1978)
- Alice and Bob don't need to hide the key from Lucy before communicating securely
- Bob can openly distribute keys which can be used to send him a secure message
- Bob doesn't need to give different keys to different people



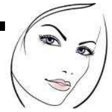
RSA – How it works

- Alice's Setup 
  - Picks two large prime numbers p and q
  - Multiplies p and q to obtain n
  - $w = (p - 1)(q - 1)$
  - Chooses e, such that e and w are relatively prime (no common factor).
  - Chooses d such that $1 = d \times e \bmod w$ ($d = e^{-1} \bmod w$)
  - Public key is: <e, n> ➔ send this key to Bob
  - Private key is: <d, n> ➔ keep this key in a safe place
  - Message code m, secret code c
    - $c = m^e \bmod n$: *encryption*
      - *Bob encrypts a message m for Alice*
    - $m = c^d \bmod n$: *decryption*
      - *Alice receives and decrypts ciphertext c*

RSA – How it works

- Alice's Setup 
  - Picks two large prime numbers p and q
    - p = 47 and q = 71
  - Multiplies p and q to obtain n
    - $n = p*q =$ **3337**
  - $w = (p - 1)(q - 1) = 46 * 70 =$ **3220**
  - Chooses e, such that e and w are relatively prime (no common factor).
    - $1 < e < w$, GCD (e,w) = 1
    - 79 (Extended Euclidean Algorithm's Table Method)
  - Chooses d such that 1 = d x e mod w ($d = e^{-1}$ *mod* w)
    - $d = 79^{-1}$ *mod 3220 = 1019*
  - Public key is: <*e, n*> - <79, 3337>
  - Private key is: <*d, n*> - <1019, 3337>
  - Message code m, secret code c
    - $c = m^e$ mod n
      - m = hello [104_{m1} 101_{m2} 108_{m3} 108_{m4} 111_{m5}]
      - c = [104^{79} mod 3337, 101^{79} mod 3337, 108^{79} mod 3337, 108^{79} mod 3337, 111^{79} mod 3337]
      - c = [2.2+e159 mod 3337, 2.19+e158 mod 3337, 4.36+e160 mod 3337, 4.36+e160 mod 3337, 3.80 + 161 mod 3337]
      - c = [2893, 1113, 1795, 1795, 2237]
    - $m = c^d$ mod n
      - m = [2893^{1019}, 1113^{1019} mod 3337, 1795^{1019} mod 3337, 1795^{1019} mod 3337, 2237^{1019} mod 3337]
      - m = [104, 101, 108, 108, 111]

RSA – How it works

- Alice's Setup 
  - Picks two large prime numbers *p* and *q*
    - p = 47 and q = 71
  - Multiplies *p* and *q* to obtain *n*
    - $n=p*q=3337$
  - $w = (p - 1)(q - 1) = 46 * 70 = 3220$
  - Chooses *e*, such that *e* and w are relatively prime (no common factor).
    - $1 < e < w$, GCD (e,w) = 1
    - 79 (Extended Euclidean Algorithm's Table Method)
  - Chooses *d* such that $1 = d \times e \bmod w$ (**$d = e^{-1} \bmod w$**)
    - $d = 79^{-1} \bmod 3220 =$ ***1019***
  - Public key is: <*e, n*> - <79, 3337>
  - Private key is: <*d, n*> - <1019, 3337>
  - Message code *m*, secret code *c*
    - $c = m^e \bmod n$
      - m = hello [104_{m1} 101_{m2} 108_{m3} 108_{m4} 111_{m5}]
      - $c = [104^{79} \bmod 3337, 101^{79} \bmod 3337, 108^{79} \bmod 3337, 108^{79} \bmod 3337, 111^{79} \bmod 3337]$
      - c = [2.2+e159 mod 3337, 2.19+e158 mod 3337, 4.36+e160 mod 3337, 4.36+e160 mod 3337, 3.80 + 161 mod 3337]
      - c = [2893, 1113, 1795, 1795, 2237]
    - $m = c^d \bmod n$
      - $m = [2893^{1019}, 1113^{1019} \bmod 3337, 1795^{1019} \bmod 3337, 1795^{1019} \bmod 3337, 2237^{1019} \bmod 3337]$
      - m = [104, 101, 108, 108, 111]

RSA – How it works

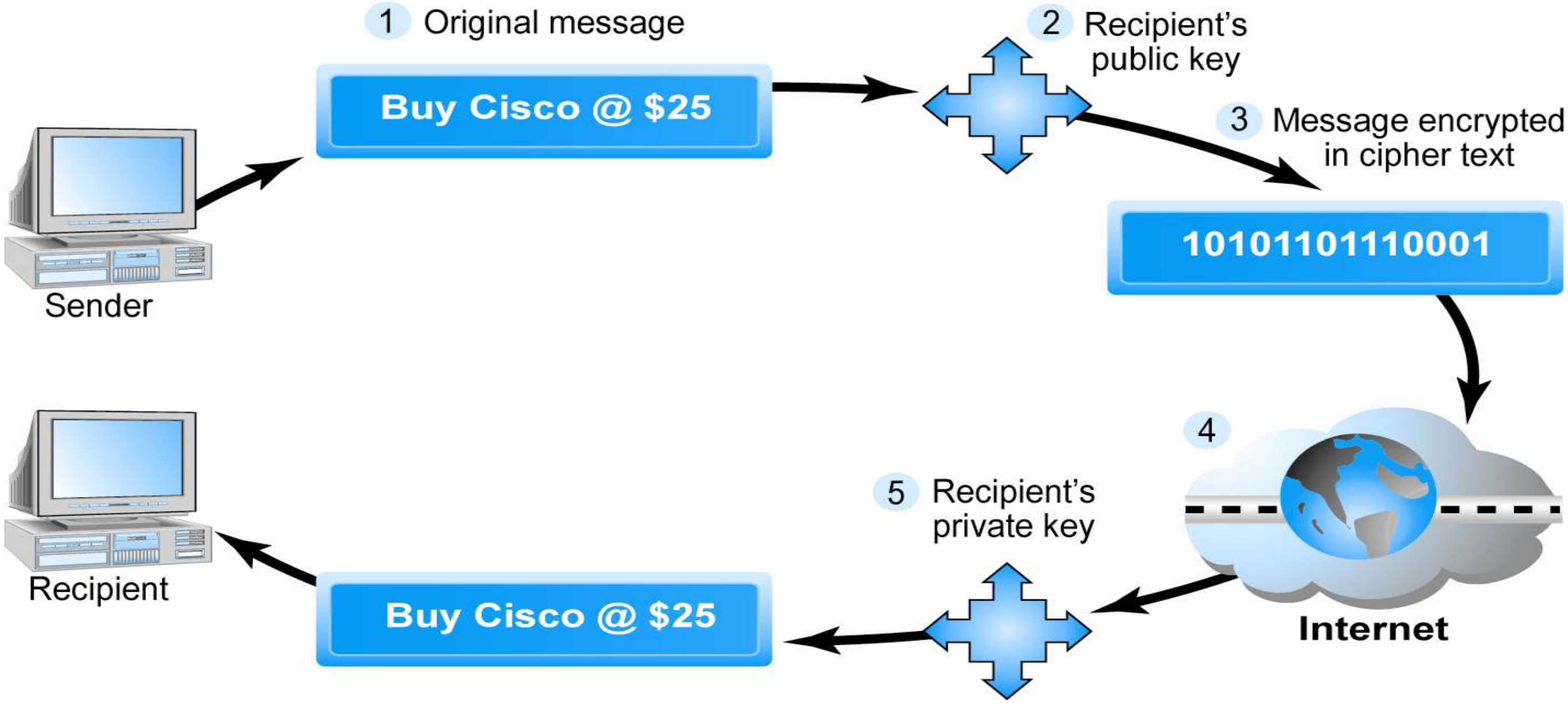
- Bob encrypts a message 
  - Picks two large prime numbers *p* and *q*
    - p = 47 and q = 71
  - Multiplies *p* and *q* to obtain *n*
    - *n=p*q=3337*
  - *w* = (*p* - 1)(*q* - 1) = 46 * 70 = 3220
  - Chooses **e**, such that *e* and w are relatively prime (no common factor).
    - *1 < e < w,* GCD (e,w) = 1
    - 79 (Extended Euclidean Algorithm's Table Method)
  - Chooses *d* such that 1 = *d* x *e* mod *w* (*d* = e^{-1} *mod w)*
    - $d = 79^{-1} \bmod 3220 = 1019$
  - Public key is: <*e, n*> - <79, 3337>
  - Private key is: <*d, n*> - <1019, 3337>

Public key is: <*e, n*> - <79, 3337>

Private key is: <*d, n*> - <1019, 3337>

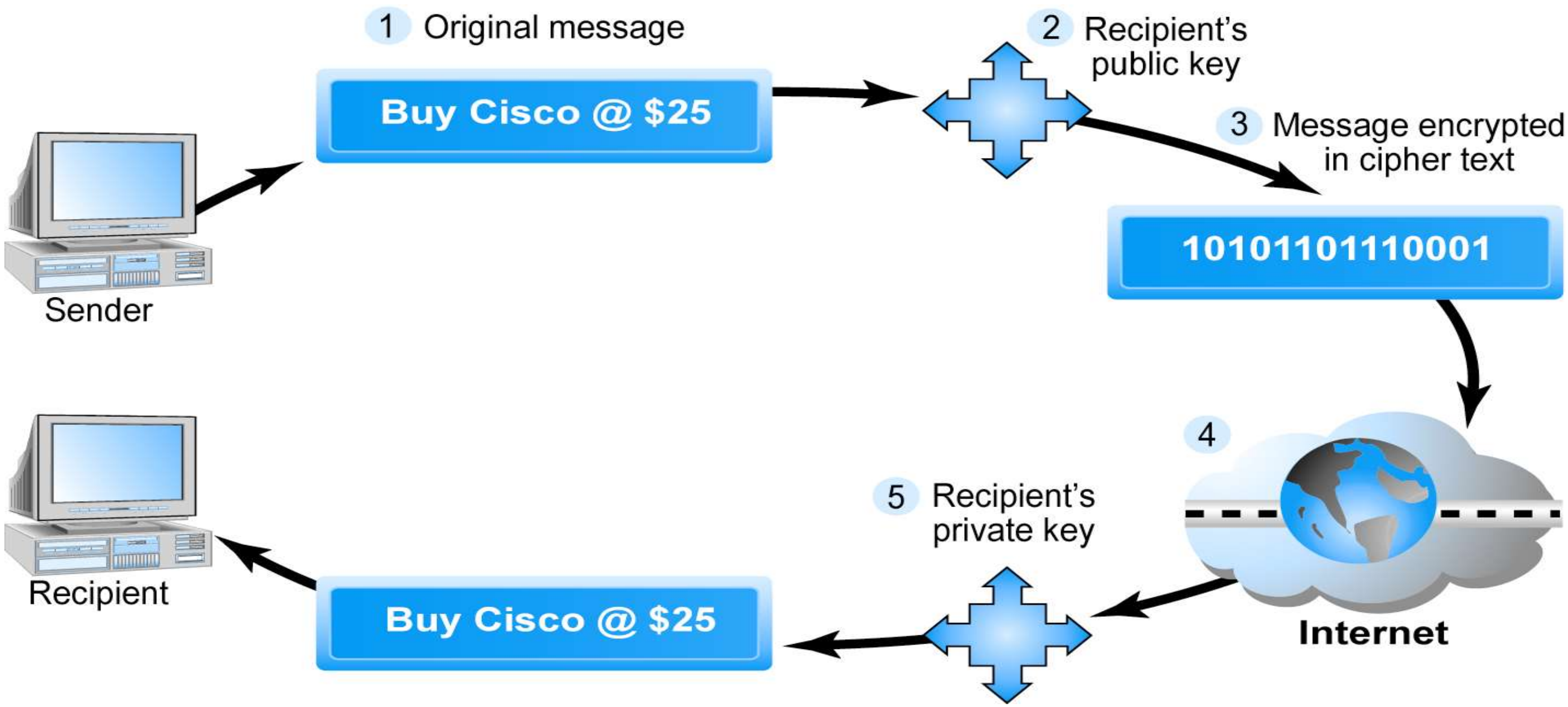
  - Message code *m*, secret code *c*
    - $c = m^e \bmod n$
      - *m* = ***h e l l o*** $[104_{m1}\ 101_{m2}\ 108_{m3}\ 108_{m4}\ 111_{m5}]$
      - $c = [104^{79} \bmod 3337,\ 101^{79} \bmod 3337,\ 108^{79} \bmod 3337,\ 108^{79} \bmod 3337,\ 111^{79} \bmod 3337]$
      - *c = [2.2+e159 mod 3337, 2.19+e158 mod 3337, 4.36+e160 mod 3337, 4.36+e160 mod 3337, 3.80 + 161 mod 3337]*
      - *c = [2893, 1113, 1795, 1795, 2237]*
- Alice decrypts the message 
    - $m = c^d \bmod n$
      - $m = [2893^{1019} \bmod 3337,\ 1113^{1019} \bmod 3337,\ 1795^{1019} \bmod 3337,\ 1795^{1019} \bmod 3337,\ 2237^{1019} \bmod 3337]$
      - *m = [104, 101, 108, 108, 111]* ***h e l l o***

RSA – public key encryption



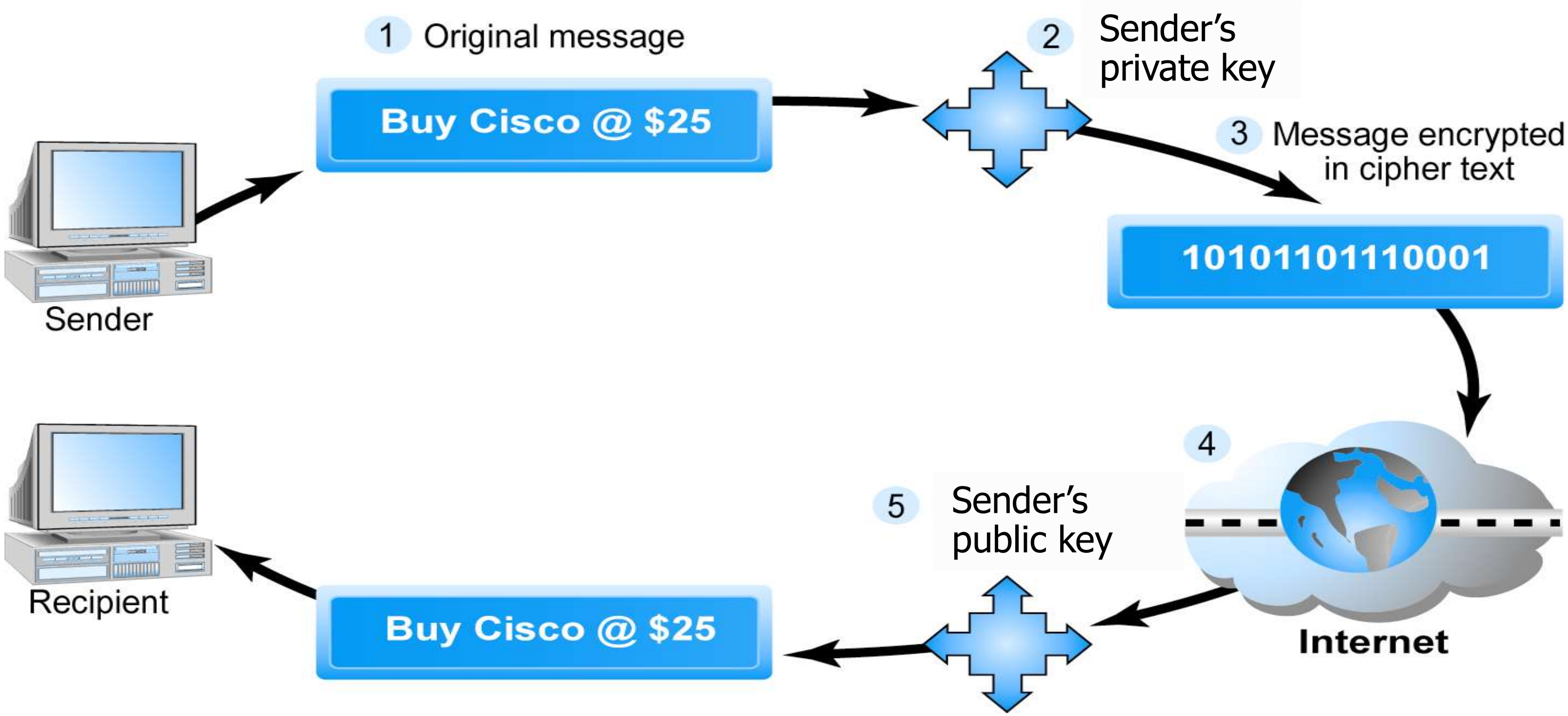
RSA – public key encryption – Confidentiality

Intercepted message cannot be read



RSA – public key encryption – Authentication

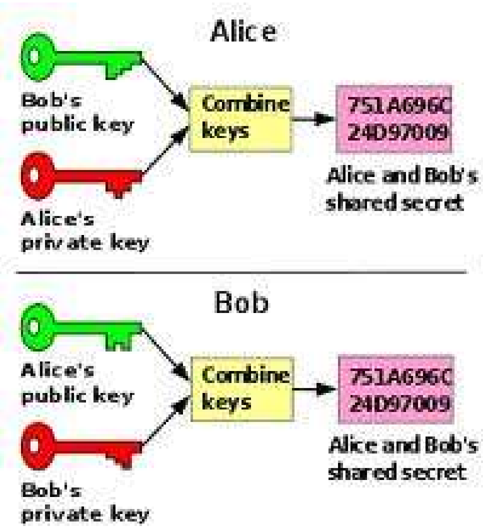
We should know who sent the message

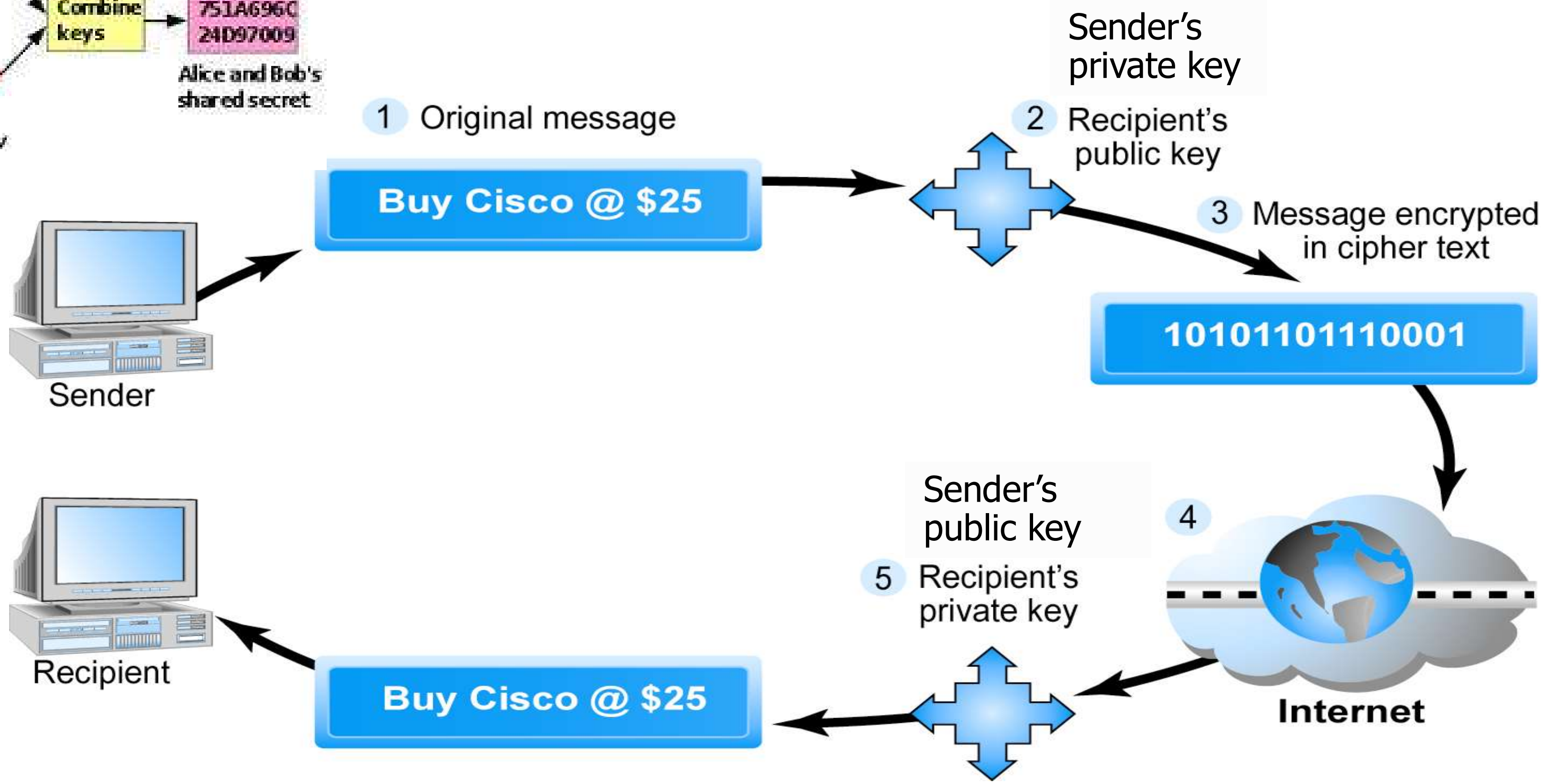


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

RSA – public key encryption

Confidentiality
Authentication



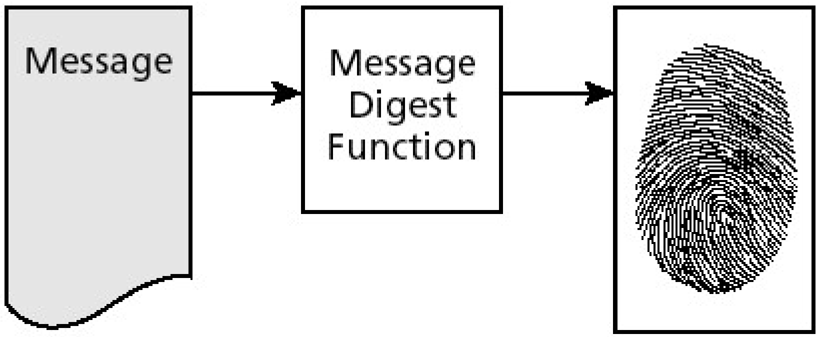


Lecture Overview

- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

Data Integrity: Message digest

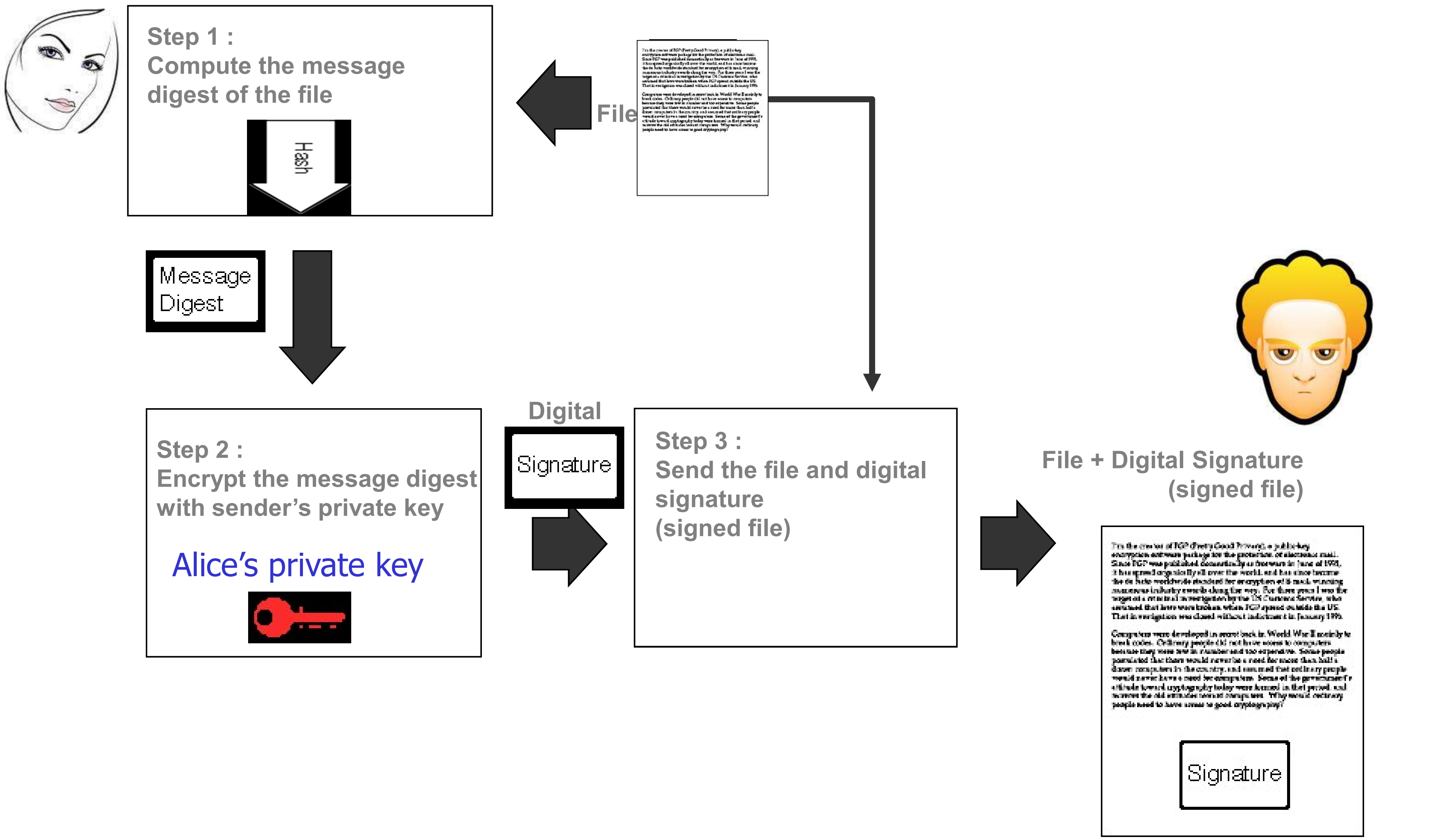
- In some cases, we may only concern with data integrity.
- As it is slow to perform encryption, it may not be necessary to encrypt all messages.
- A message digest algorithm can generate an almost unique message digest (looks like a "fingerprint") for a message.
- A popular message digest algorithm is MD5.



Data integrity: message digest - Digital Signatures

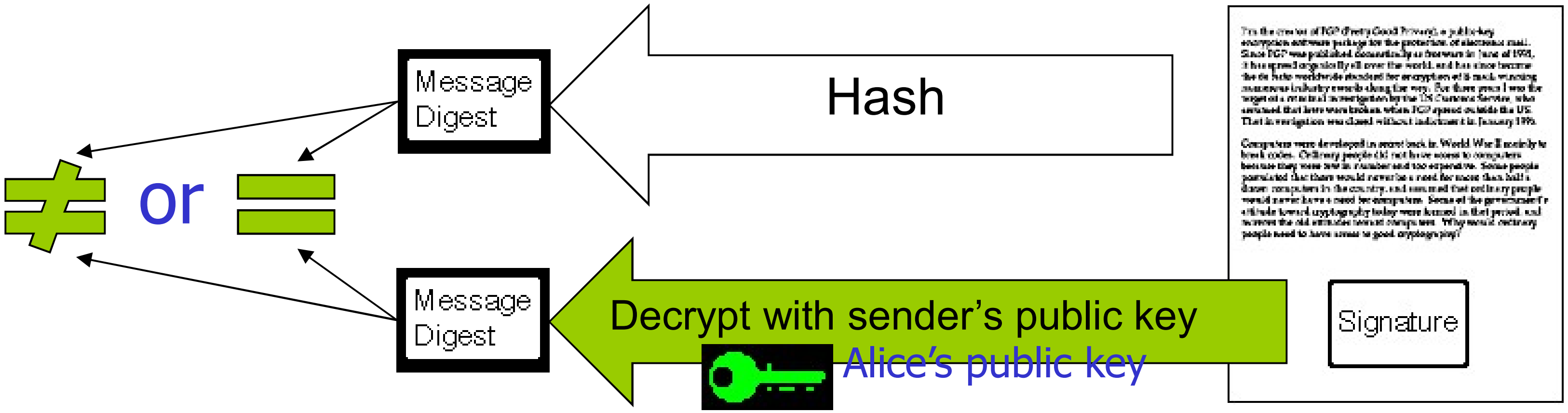
- Can use **one-way hash functions**
  - Creates a hash value (checksum) of the message
  - The hash value is relatively small
  - Given a message, it's easy to generate the hash value
  - Given the hash value, it's difficult to reconstruct the message
  - Given the hash value, it's difficult to find a message which has the same hash value

Steps in digital signature generation



Digital Signatures and Public Keys

- Secure Hash Algorithm (SHA)
  -  Bob decrypts the digital signature using Alice's public key
  - He then computes the hash code and compares it with the decrypted value
  - If they match he has ensured the integrity of the message
  - He has also ensured non-repudiation, as only Alice could have signed it with her private key!



Lecture Overview

- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

So, what are the problems?

- Strong cryptography:
  - Methods that are considered safe in this practical sense
  - Methods which can only be broken (guessed) after a long time (1000s of years)
- Weak cryptography:
  - Methods that can be broken in a practical time frame
  - DES 64 is now considered Weak Cryptography
    - due to its small key size

So, what are the problems?

- Security of private keys
  - Your keys are usually stored in a file on disk
  - They are usually password protected – another level of security
- Counterfeit keys
  - How do you know the public key you receive really belongs to the claimed owner?
  - A variation on the man-in-the-middle attack

Counterfeit Keys

A real public key in a public directory is replaced with a **counterfeit key** by Lucy

Alice (unknowingly) uses this key to send to Bob

Message encrypted with **counterfeit key** is sent across the Internet

Message is intercepted en route by Lucy

Message is decoded with the private key that matches the **counterfeit public key**

The decoded message is encoded again – this time using Bob's real public key

New message is sent out onto the Internet with forged e-mail headers

Encrypted message arrives to Bob which looks like it comes from Alice

Bob receives the message, decodes it and thinks nothing is wrong!

Lecture Overview

- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

Public-Key Certificates

- One Solution: Trusted Third Party
  - Bob physically goes to a "Certificates Authority"
    - http://www.necs.gov.au/Digital-Signature-Certificate-Suppliers/default.aspx
  - Presents his public key and valid ID
  - Issued with a Cert which include:
    - Bob's distinguished name (DN):
      - C=County O=Organization CN=Authority
    - Bob's public key
    - Issuer's distinguished name (DN)
    - Validity Period
    - Serial Number
    - Issuer's digital signature

Public-Key Certificates

-  Alice asks for Bob’s certificate
- Alice verifies the certificate
  – Must know the public key of the Issuer
  – Issuer is well-known and has published key
  – Or the issuer’s key can be certified by another issuer whom Alice knows and trusts
  – This chain of certification is called the *public key infrastructure*
  – Web browsers come with a few certificates installed
- Alice extract Bob’s public key and can now send him a message

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Lecture Overview

- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

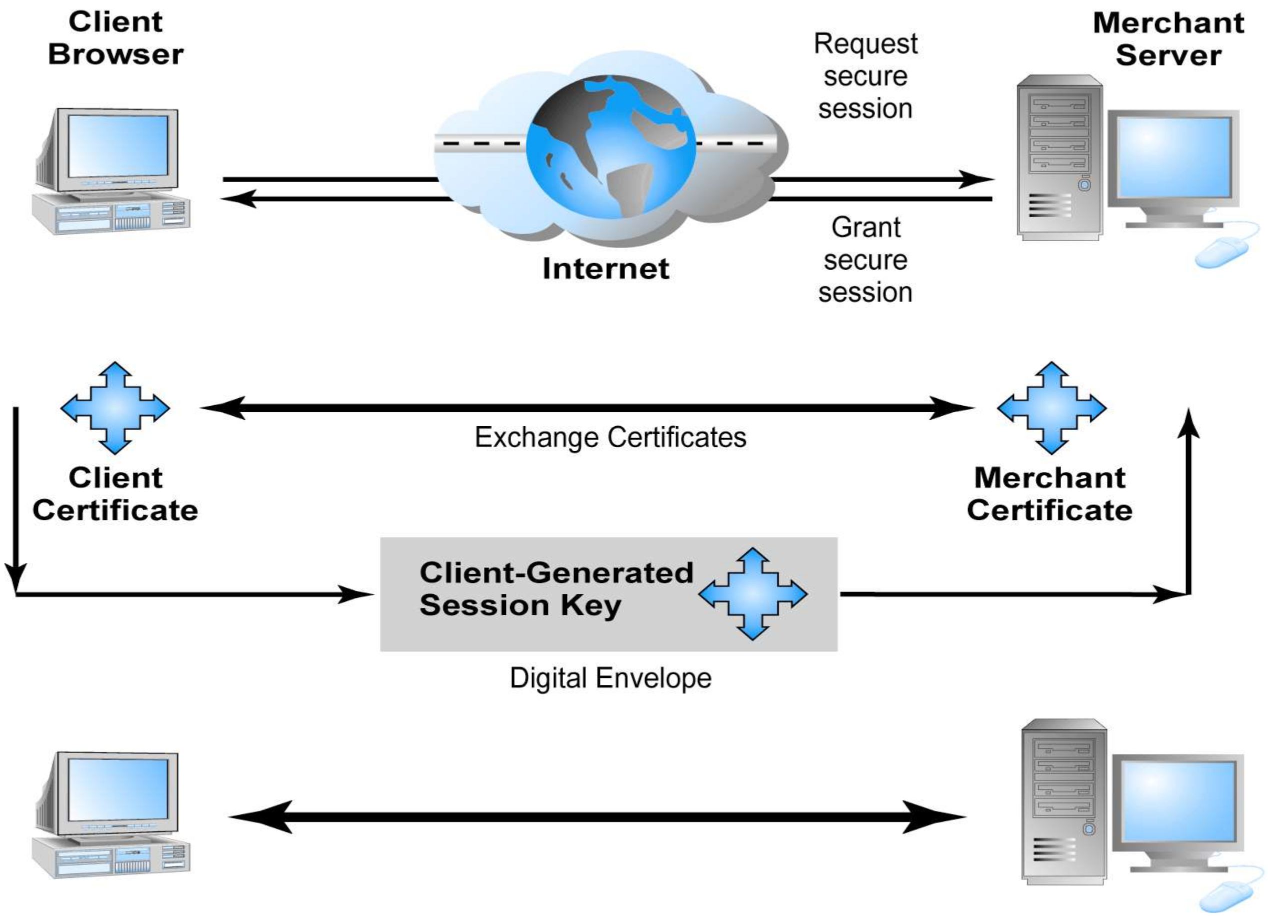
Application (HTTP)
Security (SSL)
Transport (TCP)
Network (IP)
Data link (PPP)
Physical (modem, ADSL, cable TV)

Secure connection: Secure Sockets Layer (SSL)

- Used to create a secure and reliable communication channel
  - The connection is private
    - Encryption is used after the initial handshake to define the cryptographic protocol
    - Secret key encryption (=symmetric cryptography) is used for data encryption
  - Sender/Receiver can authenticate each others identity
  - The connection is reliable

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Secure Negotiated Sessions Using SSL



Session ID and methods of encryption negotiated.

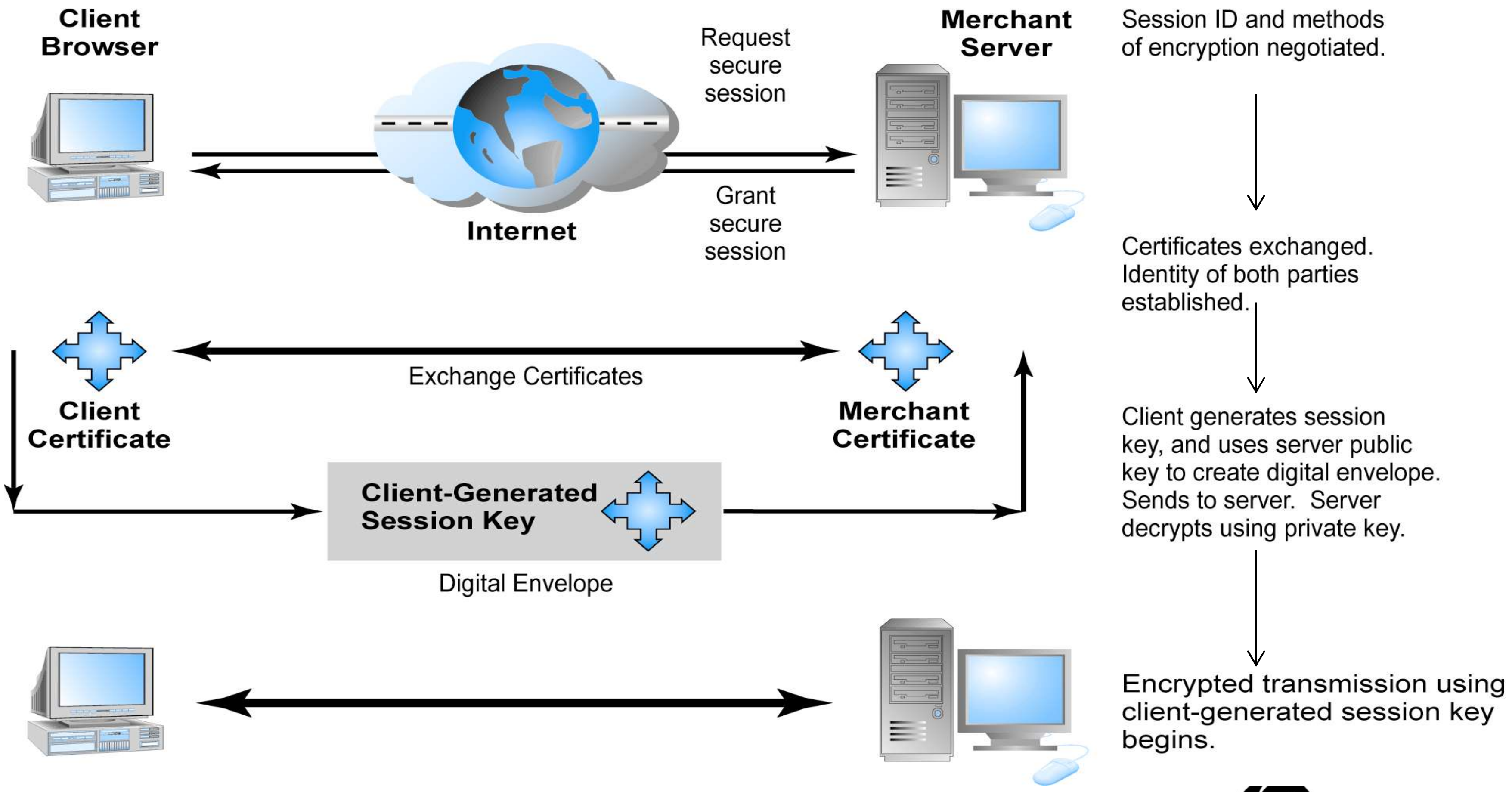
Certificates exchanged. Identity of both parties established.

Client generates session key, and uses server public key to create digital envelope. Sends to server. Server decrypts using private key.

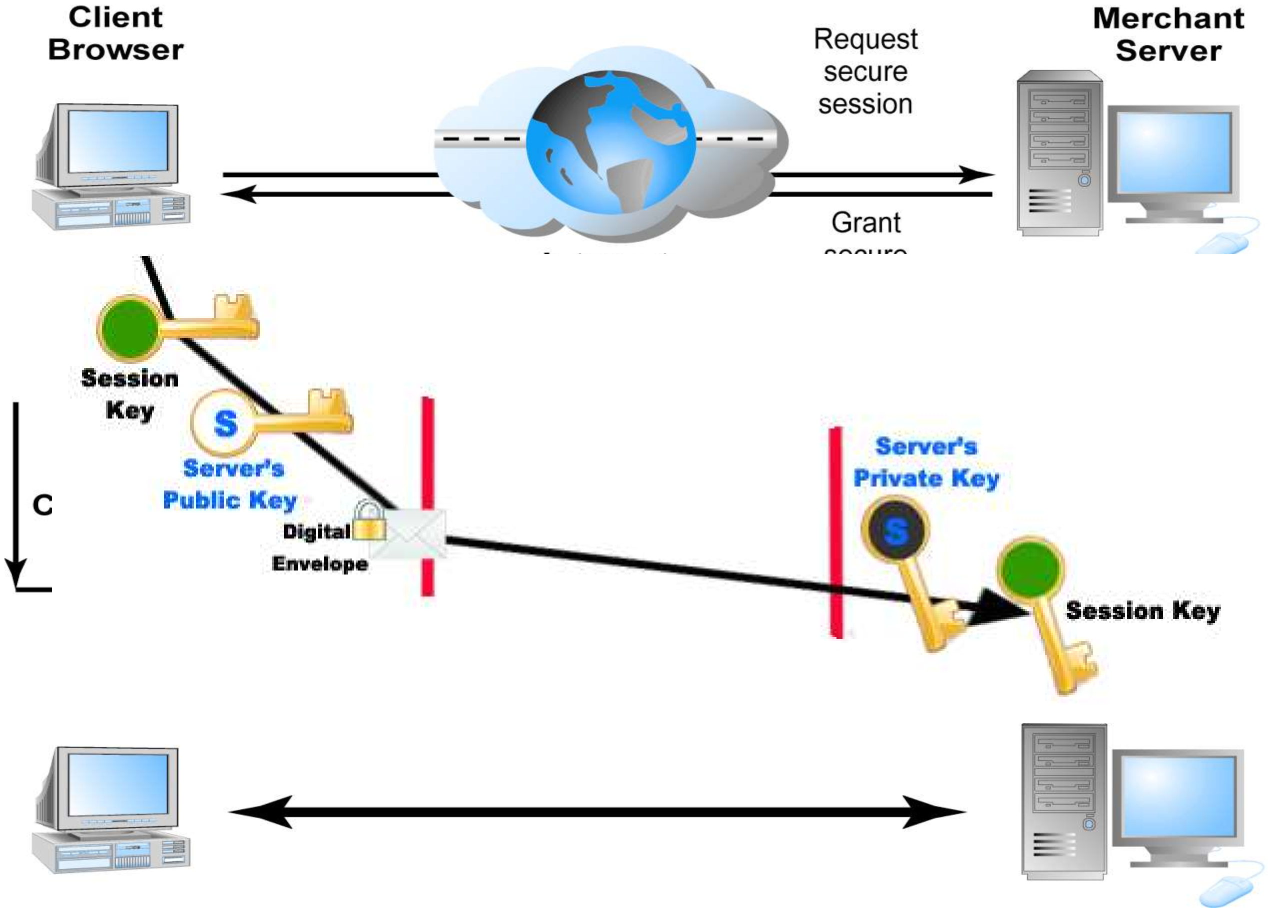
Encrypted transmission using client-generated session key begins.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Secure Negotiated Sessions Using SSL



Secure Negotiated Sessions Using SSL



Session ID and methods of encryption negotiated.

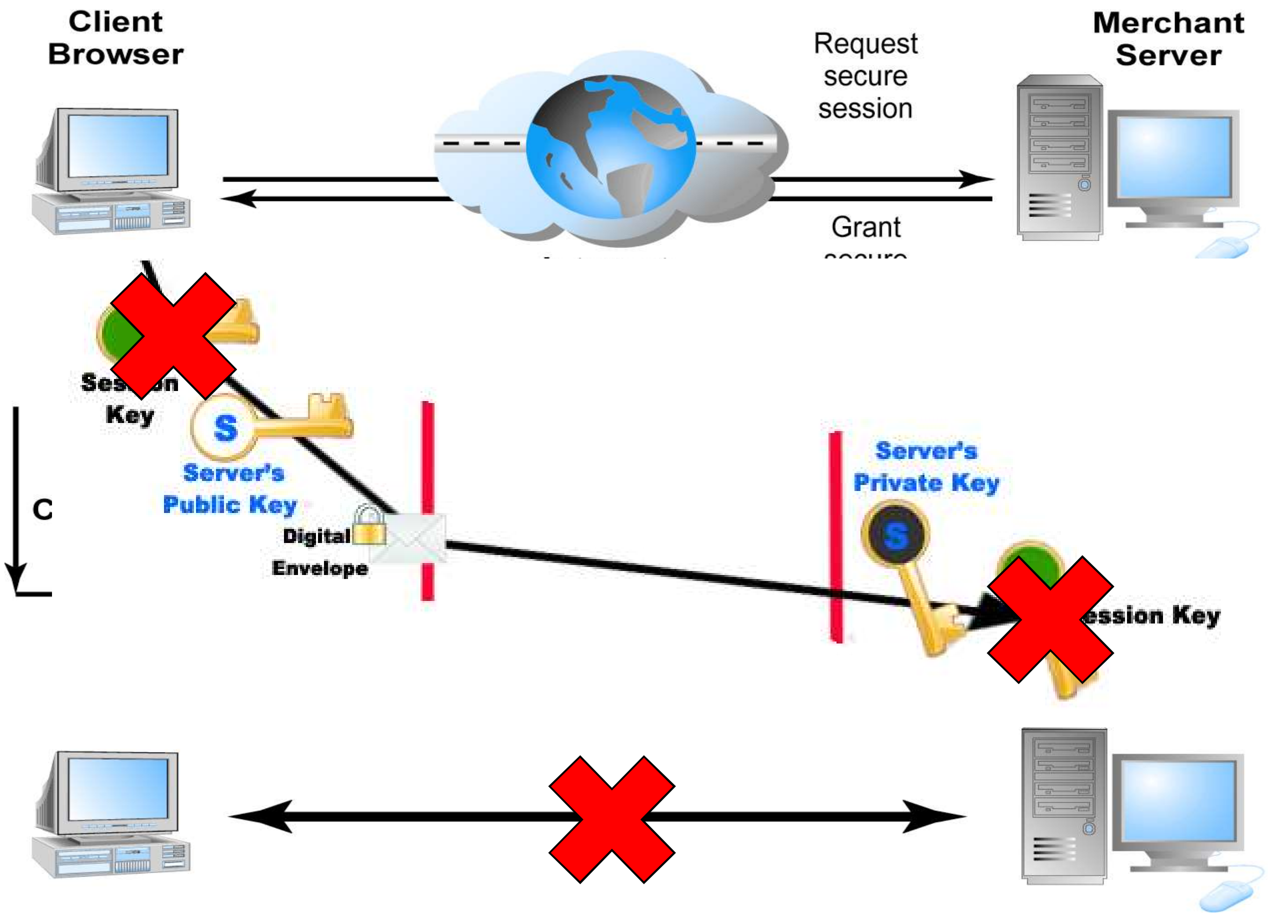
Certificates exchanged. Identity of both parties established.

Client generates session key, and uses server public key to create digital envelope. Sends to server. Server decrypts using private key.

Encrypted transmission using client-generated session key begins.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Secure Negotiated Sessions Using SSL



Session ID and methods of encryption negotiated.

Certificates exchanged. Identity of both parties established.

Client generates session key, and uses server public key to create digital envelope. Sends to server. Server decrypts using private key.

Encrypted transmission using client-generated session key begins.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

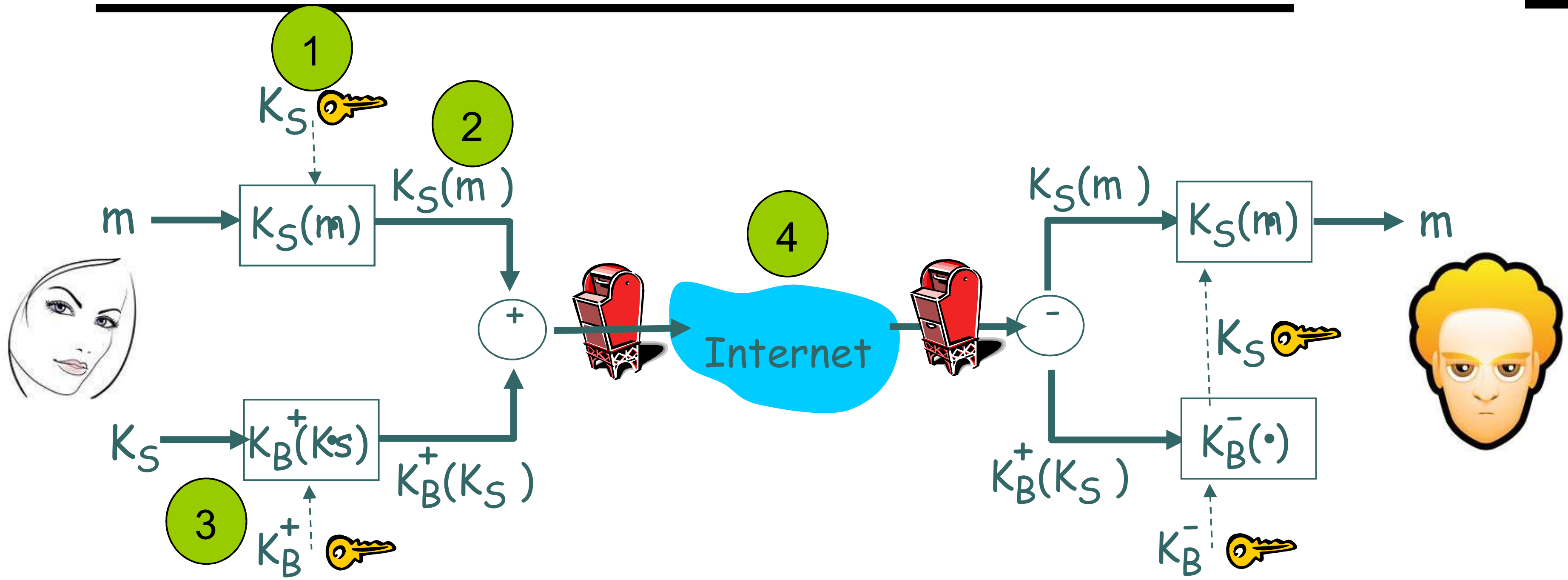
Lecture Overview

- Protecting Internet Communications
  - Foundation of Cryptography
  - Secret Key versus Public Key
  - Key Management
  - Cryptography for Data Integrity
  - Strong Cryptography
  - Public-key Certificates
  - Secure Connection: SSL
  - Popular System: PGP

Popular System: Pretty Good Privacy (PGP)

- A popular cryptographic system
- Freeware: Open PGP and variants:
  - www.openpgp.org, www.gnupg.org
- Based on RSA
  - Public keys for encrypting session keys / verifying signatures
  - Private keys for decrypting session keys / creating signatures

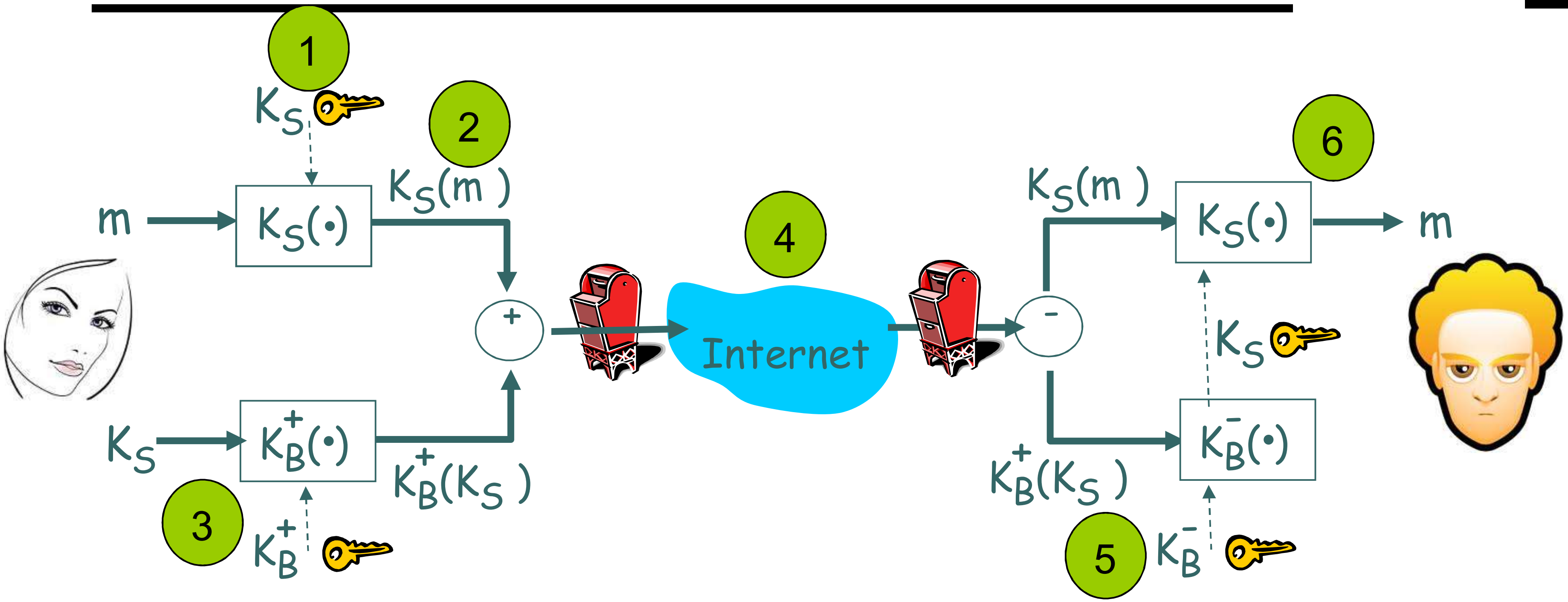
Popular System: Pretty Good Privacy (PGP)



- Alice:

1. generates random *symmetric* private key, K_S.
2. encrypts message with K_S (for efficiency)
3. also encrypts K_S with Bob's public key.
4. sends both $K_S(m)$ and $K_B(K_S)$ to Bob.

Popular System: Pretty Good Privacy (PGP)



- Bob:

5 uses his private key to decrypt and recover K_S

6 uses K_S to decrypt $K_S(m)$ to recover m

Legal Issues

- Some countries place restrictions on…
  - Who can use cryptographic software
  - The export of cryptographic software
- Laws are constantly changing
  - http://en.wikipedia.org/wiki/Export_of_cryptography
  - http://www.efa.org.au/Issues/Crypto/cryptfaq.html

    *"There are currently no direct controls limiting the **import** of cryptographic software or hardware to Australia, nor for the **domestic use** of cryptography **within** Australia (export of cryptographic technology is a different story). Even so, there are some limits in place or currently planned which can have a similar effect."*

Questions?