

# Public Key Cryptography

Sender

# Public Key Crypto

Receiver

**Message  
Source**

**Message  
Destination**

- ❑ Sender's Public Key
- ❑ Sender's Private Key

- ❑ Receiver's Public Key
- ❑ Receiver's Private key

❑ Encrypt with  
Sender's Private Key



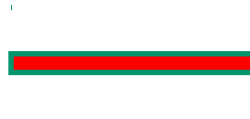
❑ Decrypt with Sender's  
Public Key

❑ Encrypt with  
Receiver's Public key



❑ Decrypt with  
Receiver's Private Key

❑ Encrypt with  
Sender's Public Key



❑ Decrypt with  
Sender's Private Key

❑ Encrypt with  
Receiver's Private Key



❑ Decrypt with  
Receiver's Public Key

# Public Key Cryptography

- ❑ Two keys
  - o Sender uses recipient's **public key** to encrypt
  - o Receiver uses his **private key** to decrypt
- ❑ Based on **trap door, one way function**
  - o Easy to compute in one direction
  - o Hard to compute in other direction
  - o “Trap door” used to create keys

# Asymmetric Encryption

Two keys are involved.  
One is Public: can be send / published.  
One is Private: Known only to the owner.

The concept of asymmetric encryptions rests on assumption that it is extremely difficult to find the factors of a very large number that is a product of two prime numbers.  
Let  $N = p \times q$ ; where both  $p$  and  $q$  are large prime numbers.  
It is difficult to factors  $N$



Lost your Private key?  
Get bankrupted  
in seconds!

# Public Key Cryptography

- ❑ Two keys
  - o Sender uses recipient's **public key** to encrypt
  - o Receiver uses his **private key** to decrypt
- ❑ Based on **trap door, one way function**
  - o Easy to compute in one direction
  - o Hard to compute in other direction
  - o “Trap door” used to create keys

Sender

# This is Asymmetric Encryption Type A

Receiver

**Message Source**

**X**

1. Senders has a message "X" to send to the receiver

2. Senders asks for Receiver's Public Key

3. Receiver's Public Key

**4. Cipher Text**

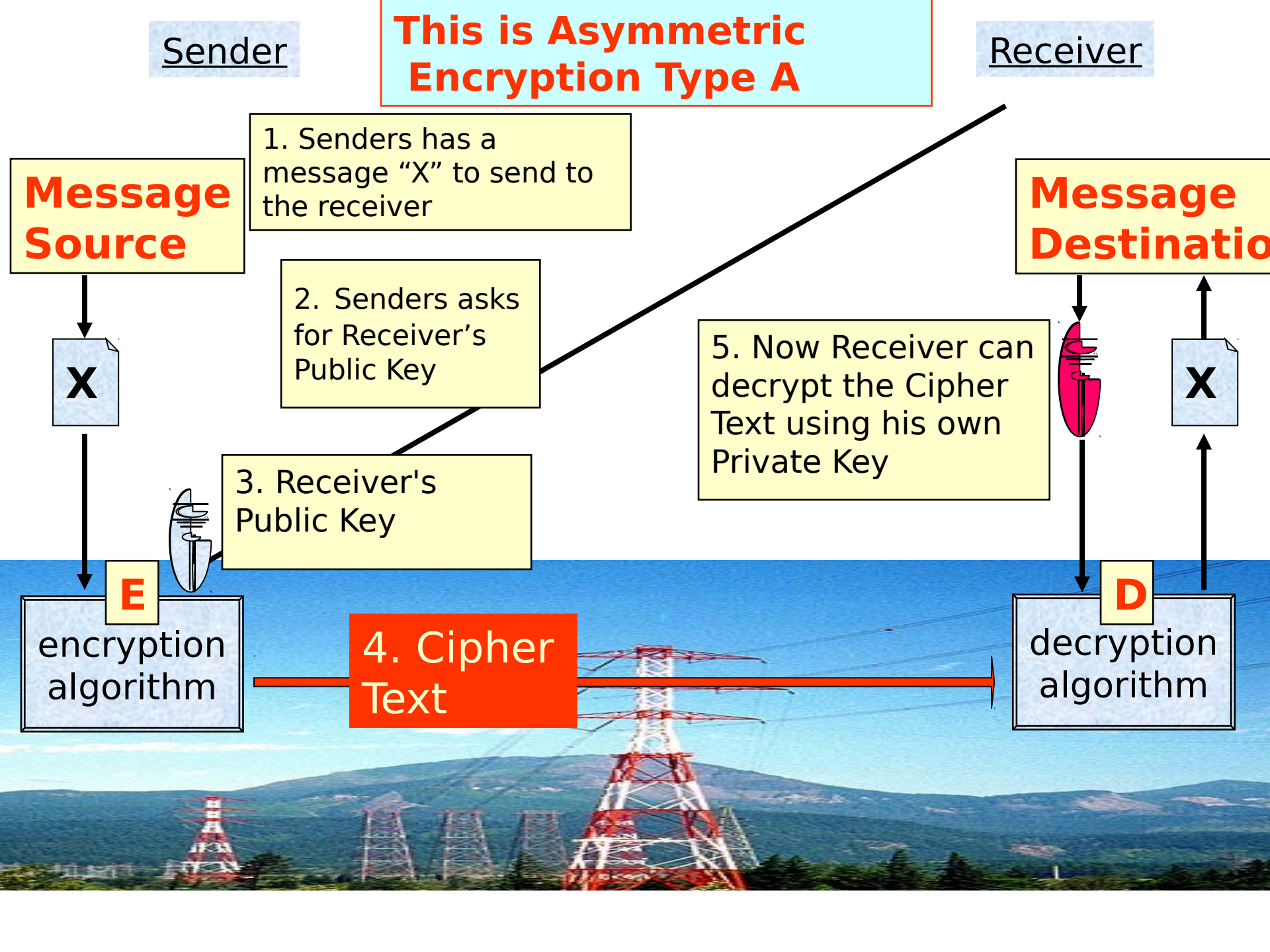
**E**  
encryption algorithm

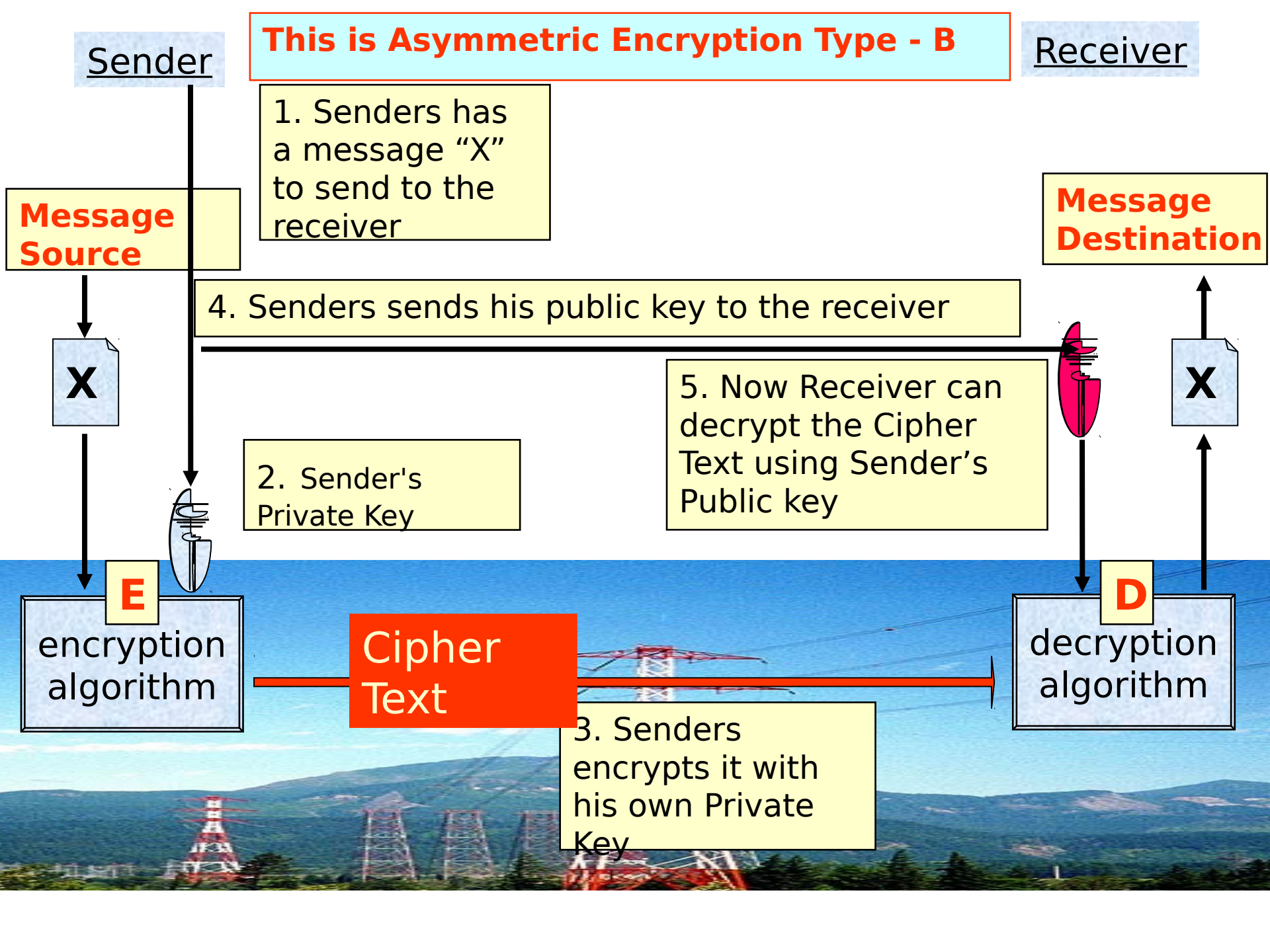
**Message Destination**

**X**

5. Now Receiver can decrypt the Cipher Text using his own Private Key

**D**  
decryption algorithm







Integrity ?  
Authenticity?

Type A v/s Type B

Secrecy ?  
Privacy ?

**No**

- ❑ **Type A !**
- ❑ **Keys used:**
- ❑ **Receiver's pair**
  - Receiver's Public key
  - Receiver's Private key
- ❑ **Achievement:**
  - Secrecy, Privacy
- ❑ **(Receiver's public key could be with many)**

**No**

- ❑ **Type B !**
- ❑ **Keys used**
- ❑ **Sender's pair**
  - Sender's Public Key
  - Sender's Private key
- ❑ **Achievement:**
  - Integrity
  - Authenticity
- ❑ **(Sender's public key could be with many)**



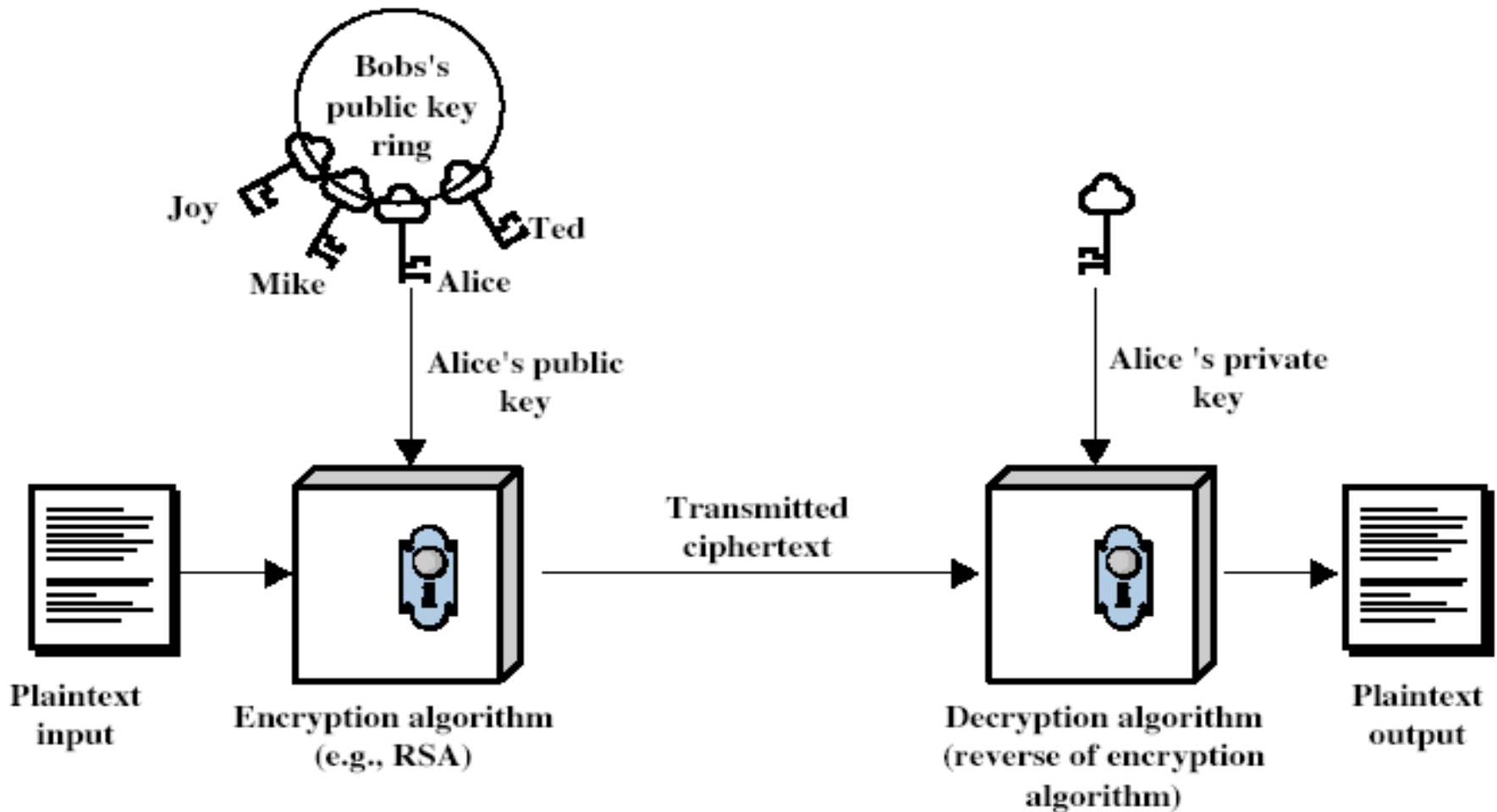
# Public-Key Cryptography

## Public-key/two-key/asymmetric

cryptography involves the use of **two** keys:

- ▮ a **public-key**, which may be known by anybody, and can be used to **encrypt messages**, and **verify signatures**
- ▮ a **private-key**, known only to the recipient, used to **decrypt messages**, and **sign** (create) **signatures**
- ▮ **Asymmetric** because
  - ▮ those who encrypt messages or verify signatures **cannot** decrypt messages or create signatures

# Public-Key Cryptography



# ▯ Public-Key Characteristics

- ▯ Public-Key algorithms rely on two keys with the characteristics that it is:
  - ▯ computationally infeasible to find decryption key knowing only algorithm & encryption key
  - ▯ computationally easy to encrypt/decrypt messages when the relevant (encrypt/decrypt) key is known
  - ▯ either of the two related keys can be used for encryption, with the other used for decryption (in some schemes)

RSA

# RSA

- ❑ Invented by Cocks (GCHQ), independently, by Rivest, Shamir and Adleman
- ❑ Let  $p$  and  $q$  be two large prime numbers
- ❑ Let  $N = pq$  be the **modulus**
- ❑ Choose  $e$  relatively prime to  $(p-1)(q-1)$
- ❑ Find  $d$  s.t.  $ed = 1 \bmod (p-1)(q-1)$
- ❑ **Public key** is  $(N, e)$
- ❑ **Private key** is  $(N, d)$

# RSA

- ❑ To encrypt message  $M$  compute
  - o  $C = M^e \bmod N$
- ❑ To decrypt  $C$  compute
  - o  $M = C^d \bmod N$
- ❑ Recall that  $e$  and  $N$  are public
- ❑ If attacker can factor  $N$ , he can use  $e$  to easily find  $d$  since  $ed = 1 \bmod (p-1)(q-1)$
- ❑ Factoring the modulus breaks RSA
- ❑ It is not known whether factoring is the only way to break RSA

# Simple RSA Example

## □ Example of RSA

- o Select “large” primes  $p = 11$ ,  $q = 3$
- o Then  $N = pq = 33$  and  $(p-1)(q-1) = 20$
- o Choose  $e = 3$  (relatively prime to 20)
- o Find  $d$  such that  $ed = 1 \pmod{20}$ , we find that  $d = 7$  works

□ **Public key:**  $(N, e) = (33, 3)$

□ **Private key:**  $d = 7$



# Simple RSA Example

□ **Public key:**  $(N, e) = (33, 3)$

□ **Private key:**  $d = 7$

□ Suppose message  $M = 8$

□ Ciphertext  $C$  is computed as

$$C = M^e \bmod N = 8^3 = 512 = 17 \bmod 33$$

□ Decrypt  $C$  to recover the message  $M$  by

$$M = C^d \bmod N = 17^7 = 410,338,673 = 12,434,505$$

$$\star 33 + 8 = 8 \bmod 33$$

# RSA Example

- $p = 11, q = 7, n = 77, \Phi(n) = 60$
- $d = 13, e = 37$  ( $ed = 481; ed \bmod 60 = 1$ )
- Let  $M = 15$ . Then  $C \equiv M^e \bmod n$ 
  - $C \equiv 15^{37} \bmod 77 = 71$
- $M \equiv C^d \bmod n$ 
  - $M \equiv 71^{13} \bmod 77 = 15$

# RSA Example 2

- Parameters:
  - $p = 3, q = 5, n = pq = 15$
  - $\Phi(n) = ?$
- Let  $e = 3$ , what is  $d$ ?
- Given  $M=2$ , what is  $C$ ?
- How to decrypt?

# RSA Signatures (cont.)

## Signing message M

- Verify  $0 < M < n$
- Compute  $S = M^d \bmod n$

## Verifying signature S

- Use public key  $(e, n)$
- Compute  $S^e \bmod n = (M^d \bmod n)^e \bmod n = M$

Note: in practice, a hash of the message is signed and not the message itself.

# Diffie-Hellman

# Diffie-Hellman

- ❑ Invented by Williamson (GCHQ) and, independently, by D and H (Stanford)
- ❑ A “key exchange” algorithm
  - o Used to establish a shared symmetric key
- ❑ Not for encrypting or signing
- ❑ Security rests on difficulty of **discrete log** problem: given  $g$ ,  $p$ , and  $g^k \bmod p$  find  $k$

# Diffie-Hellman

- Let  $p$  be prime, let  $g$  be a **generator**
  - For any  $x \in \{1, 2, \dots, p-1\}$  there is  $n$  s.t.  $x = g^n \bmod p$
- Alice selects secret value  $a$
- Bob selects secret value  $b$
- Alice sends  $g^a \bmod p$  to Bob
- Bob sends  $g^b \bmod p$  to Alice
- Both compute shared secret  $g^{ab} \bmod p$
- Shared secret can be used as symmetric key



# Diffie-Hellman

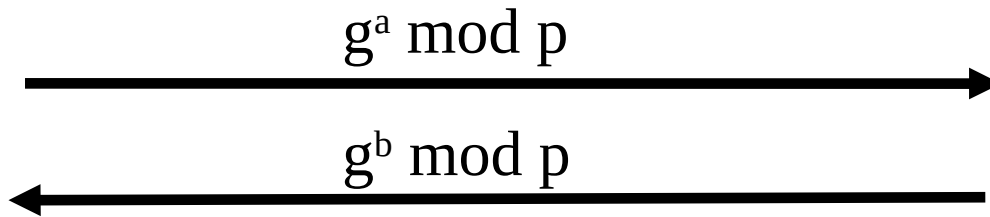
- Suppose that Bob and Alice use  $g^{ab} \bmod p$  as a symmetric key
- Trudy can see  $g^a \bmod p$  and  $g^b \bmod p$
- Note  $g^a g^b \bmod p = g^{a+b} \bmod p \neq g^{ab} \bmod p$
- If Trudy can find  $a$  or  $b$ , system is broken
- If Trudy can solve **discrete log problem**, then she can find  $a$  or  $b$

# Diffie-Hellman

- **Public:**  $g$  and  $p$
- **Secret:** Alice's exponent  $a$ , Bob's exponent  $b$



Alice,  $a$

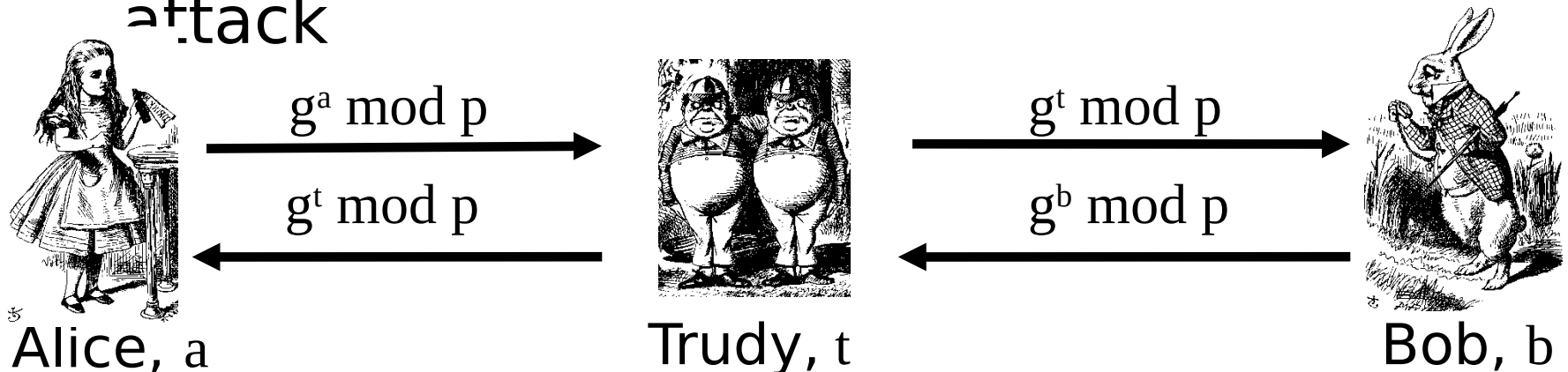


Bob,  $b$

- Alice computes  $(g^b)^a = g^{ba} = g^{ab} \bmod p$
- Bob computes  $(g^a)^b = g^{ab} \bmod p$
- Could use  $K = g^{ab} \bmod p$  as symmetric key

# Diffie-Hellman

- Subject to man-in-the-middle (MiM) attack



- Trudy shares secret  $g^{at} \bmod p$  with Alice
- Trudy shares secret  $g^{bt} \bmod p$  with Bob
- Alice and Bob don't know Trudy exists!

# Diffie-Hellman

- ❑ How to prevent MiM attack?
  - o Encrypt DH exchange with symmetric key
  - o Encrypt DH exchange with public key
  - o Sign DH values with private key
  - o Other?
- ❑ You **MUST** be aware of MiM attack on Diffie-Hellman

# Uses for Public Key Crypto

# Uses for Public Key Crypto

- ❑ Confidentiality
  - o Transmitting data over insecure channel
  - o Secure storage on insecure media
- ❑ Authentication
- ❑ Digital signature provides integrity and **non-repudiation**
  - o No non-repudiation with symmetric keys

# No-non-repudiation

- ❑ Alice orders 100 shares of stock from Bob
- ❑ Alice computes **MAC** using symmetric key
- ❑ Stock drops, Alice claims she did not order
- ❑ Can Bob prove that Alice placed the order?
- ❑ **No!** Since Bob also knows symmetric key, he could have forged message
- ❑ **Problem:** Bob knows Alice placed the order, but he can't prove it



# Non-repudiation

- Alice orders 100 shares of stock from Bob
- Alice **signs** order with her private key
- Stock drops, Alice claims she did not order
- Can Bob prove that Alice placed the order?
- **Yes!** Only someone with Alice's private key could have signed the order
- This assumes Alice's private key is not stolen (revocation problem)

# Sign and Encrypt vs Encrypt and Sign

# Public Key Notation

- **Sign** message  $M$  with Alice's **private key**:  $[M]_{\text{Alice}}$
- **Encrypt** message  $M$  with Alice's **public key**:  $\{M\}_{\text{Alice}}$
- Then

$$\{[M]_{\text{Alice}}\}_{\text{Alice}} = M$$

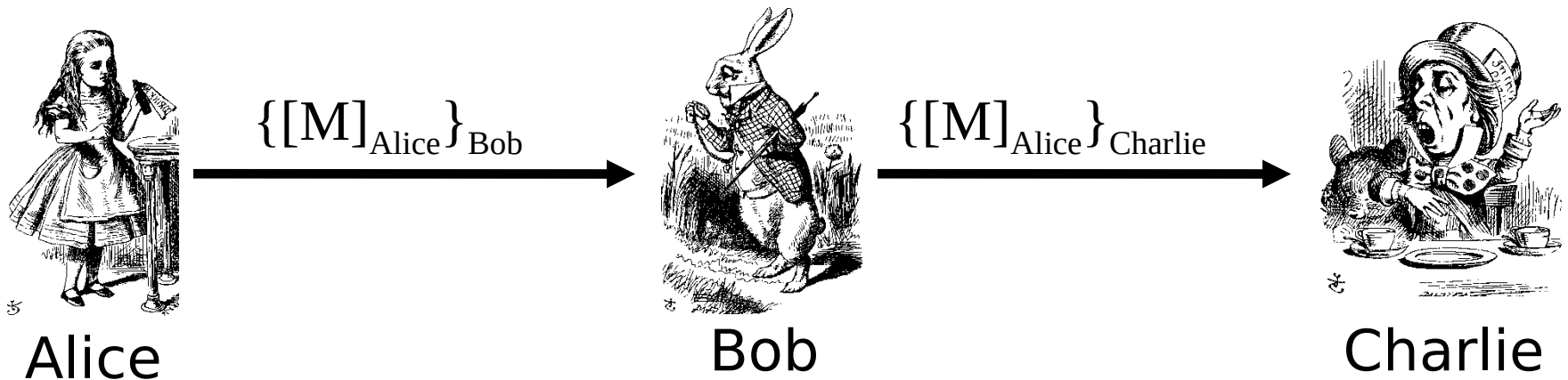
$$[\{M\}_{\text{Alice}}]_{\text{Alice}} = M$$

# Confidentiality and Non-repudiation

- ❑ Suppose that we want confidentiality and non-repudiation
- ❑ Can public key crypto achieve both?
- ❑ Alice sends message to Bob
  - **Sign and encrypt**  $\{[M]_{\text{Alice}}\}_{\text{Bob}}$
  - **Encrypt and sign**  $[\{M\}_{\text{Bob}}]_{\text{Alice}}$
- ❑ Can the order possibly matter?

# Sign and Encrypt

□  $M = \text{"I love you"}$



□ **Q:** What is the problem?

□ **A:** Charlie misunderstands crypto!

# Encrypt and Sign

- $M$  = “My theory, which is mine....”



Alice

$[\{M\}_{Bob}]_{Alice}$



Charlie

$[\{M\}_{Bob}]_{Charlie}$



Bob

- **Note** that Charlie cannot decrypt  $M$
- **Q:** What is the problem?
- **A:** Bob misunderstands crypto!

# How do I digitally sign?

- ❑ It has nothing to do with your physical signature !
- ❑ It doesn't need any pen or paper!
- ❑ But it does the work of a Physical signature!

Then what is it like:

- First take a Message Digest (MD) or simply Hash of the text matter that you wish to send, using any Message Digest algorithm such as MD5, SHA

Secure Hash Algorithm

- Then encrypt that with your Private key
- And you get your Digital Signature !
- Simple. Isn't it?
- Well ..... not exactly !!

- Let us learn it again step by step



# Confidentiality and Non-repudiation

- ❑ Suppose that we want confidentiality and non-repudiation
- ❑ Can public key crypto achieve both?
- ❑ Alice sends message to Bob
  - **Sign and encrypt**  $\{[M]_{\text{Alice}}\}_{\text{Bob}}$
  - **Encrypt and sign**  $[\{M\}_{\text{Bob}}]_{\text{Alice}}$
- ❑ Can the order possibly matter?

# Digital Signature

X

- ❑ **What is Digital Signature?**
- ❑ Digital Signature is a process by which
  - o the contents of a message and
  - o The identity of the sender can be verified.

**Digital Signature is implemented using**

- An asymmetric encryption cipher &
- A hash function

Yes. This is  
**my**  
message!



And this is  
indeed  
**me!!**

# Digital Signatures

Each individual generates his own key pair  
[Public key known to everyone & Private key only to the owner]



Private Key – Used for making digital signature

Public Key – Used to verify the digital signature

# RSA Key pair

(including Algorithm identifier)  
[2048 bit]



## Private Key

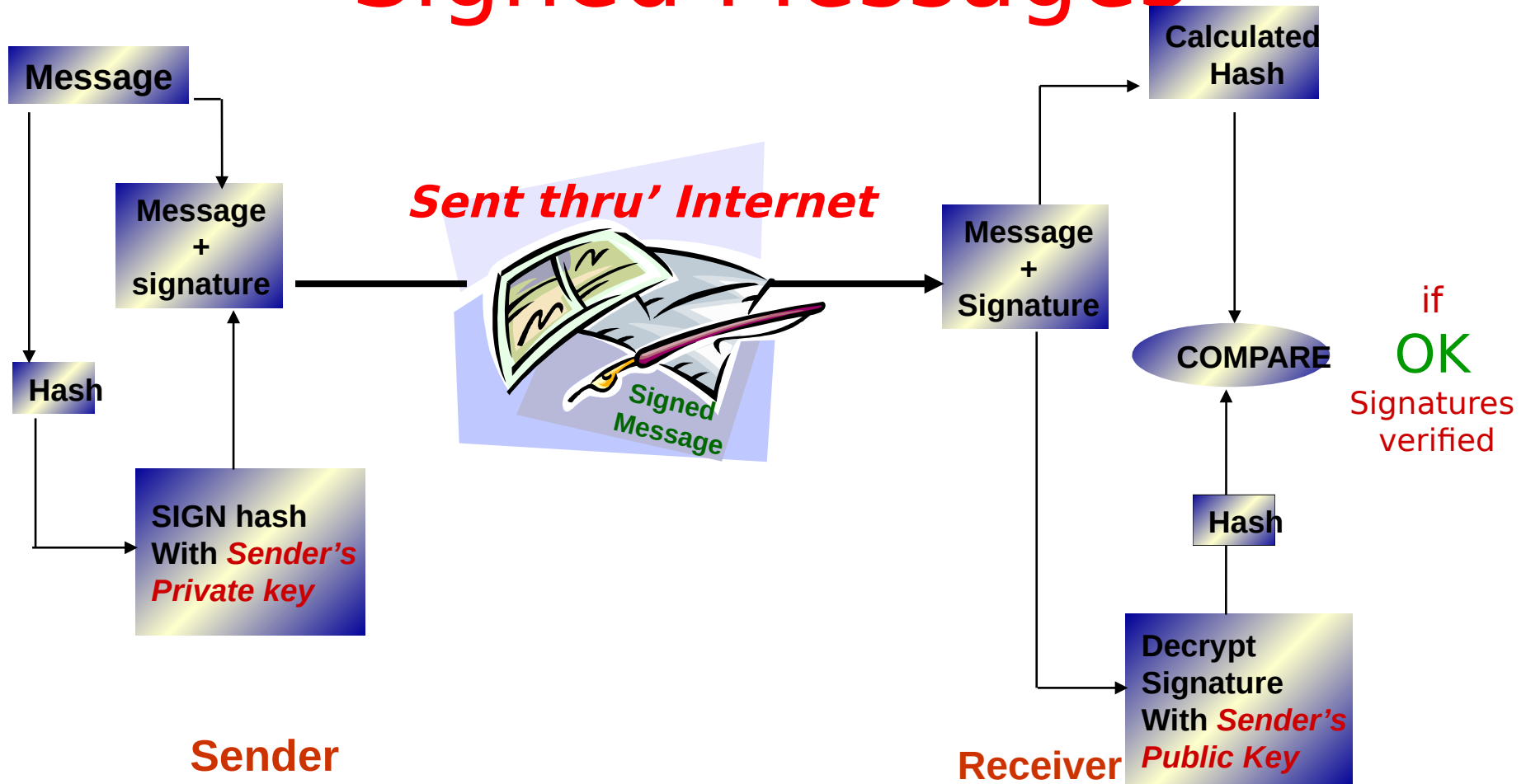
```
3082 010a 0282 0101 00b1 d311 e079 5543 0708 4ccb 0542 00e2 0d83 463d e493 bab6
06d3 0d59 bd3e c1ce 4367 018a 21a8 efbc ccd0 a2cc b055 9653 8466 0500 da44 4980
d854 0aa5 2586 94ed 6356 ff70 6ca3 a119 d278 be68 2a44 5e2f cfcc 185e 47bc 3ab1
463d 1ef0 b92c 345f 8c7c 4c08 299d 4055 eb3c 7d83 deb5 f0f7 8a83 0ea1 4cb4 3aa5
b35f 5a22 97ec 199b c105 68fd e6b7 a991 942c e478 4824 1a25 193a eb95 9c39 0a8a
cf42 b2f0 1cd5 5ffb 6bed 6856 7b39 2c72 38b0 ee93 a9d3 7b77 3ceb 7103 a938 4a16
6c89 2aca da33 1379 c255 8ced 9cbb f2cb 5b10 f82e 6135 c629 4c2a d02a 63d1 6559
b4f8 cdf9 f400 84b6 5742 859d 32a8 f92a 54fb ff78 41bc bd71 28f4 bb90 bcff 9634
04e3 459e a146 2840 8102 0301 0001
```

## Public Key

```
3082 01e4 f267 0142 0f61 dd12 e089 5547 0f08 4ccb 0542 00e2 0d83 463d e493 bab6
0673 0d59 bf3e c1ce 4367 012a 11a8 efbc ccd0 a2cc b055 9653 8466 0500 da44 4980
d8b4 0aa5 2586 94ed 6356 ff70 6ca3 a119 d278 be68 2a44 5e2f cfcc 185e 47bc 3ab1
463d 1df0 b92c 345f 8c7c 4c08 299d 4055 eb3c 7d83 deb5 f0f7 8a83 0ea1 4cb4 3aa5
b35f 5a22 97ec 199b c105 68fd e6b7 a991 942c e478 4824 1a25 193a eb95 9c39 0a8a
cf42 b250 1cd5 5ffb 6bed 6856 7b39 2c72 38b0 ee93 a9d3 7b77 3ceb 7103 a938 4a16
6c89 2aca da33 1379 c255 8ced 9cbb f2cb 5b10 f82e 6135 c629 4c2a d02a 63d1 6559
b4f8 cdf9 f400 84b6 5742 859d 32a8 f92a 54fb ff78 41bc bd71 28f4 bb90 bcff 9634
04de 45de af46 2240 8410 02f1 0001
```



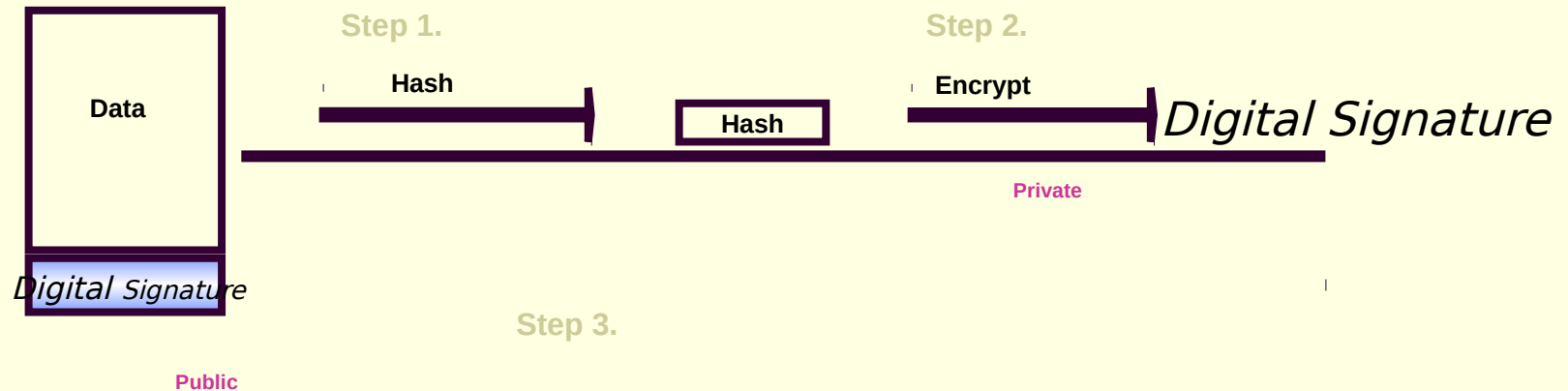
# Signed Messages



# What is a Digital Signature ?

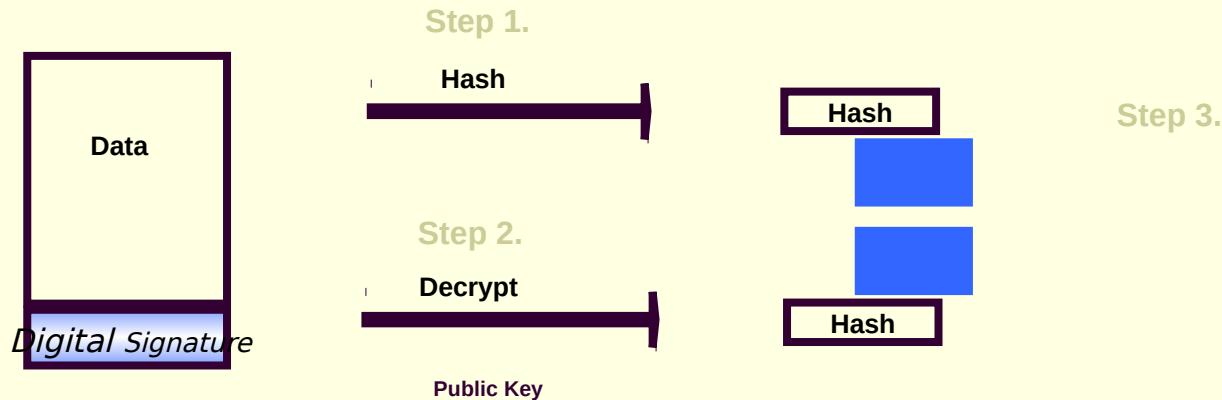
- A Digital Signature is the result of **encrypting** the Hash of the data to be exchanged.
- A Hash (or Message Digest) is the process of mathematically reducing a data stream down to a fixed length field.
- The Hash uniquely represents the original data.
- The probability of producing the same Hash with two sets of different data is  $<.001\%$ .
- Signature Process is opposite to Encryption Process
  - Private Key is used to Sign (encrypt) Data
  - Public Key is used to verify (decrypt) Signature

# Digital Signature Process



- Step 1. Hash (digest) the data using one of the supported Hashing algorithms, e.g., MD2, MD5, or SHA-1.
- Step 2. Encrypt the hashed data using the sender's private key.
- Step 3. Append the signature (and a copy of the sender's public key) to the end of the data that was signed.

# Signature Verification Process



- Step 1. Hash the original data using the same hashing algorithm.
- Step 2. Decrypt the digital signature using the sender's public key. All digital signatures contain a copy of the signer's public key.
- Step 3. Compare the results of the hashing and the decryption. If the values match then the signature is verified. If the values do not match, then the data or signature was probably modified in transit.



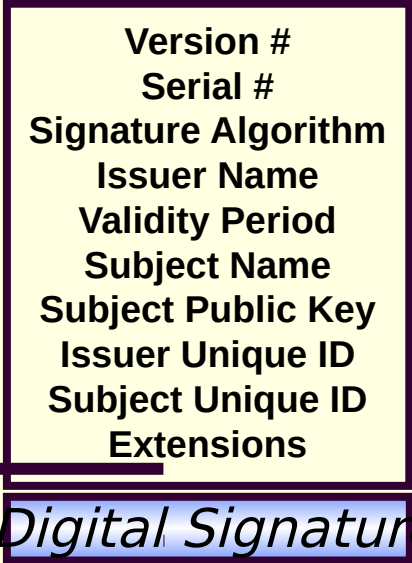
# The Critical Questions

- How can the recipient know with certainty the sender's public key? (to validate a digital signature)
- How can the sender know with certainty
- the recipient's public key? (to send an
- encrypted message)
-

# Digital Certificates

- A Digital Certificate is simply an X.509 defined data structure with a Digital Signature. The data represents who owns the certificate, who signed the certificate, and other relevant information

X.509 Certificate



The diagram shows a rectangular box representing the X.509 Certificate. Inside the box, the following fields are listed from top to bottom: Version #, Serial #, Signature Algorithm, Issuer Name, Validity Period, Subject Name, Subject Public Key, Issuer Unique ID, Subject Unique ID, and Extensions. To the left of the box, there is a tilted rectangular stamp that reads 'CA Authorized'. Below the box, there is a blue rectangular area labeled 'Digital Signature'.

Version #  
Serial #  
Signature Algorithm  
Issuer Name  
Validity Period  
Subject Name  
Subject Public Key  
Issuer Unique ID  
Subject Unique ID  
Extensions

*Digital Signature*

- When the signature is generated by a Certification Authority (CA), the signature can be viewed as trusted.
- Since the data is signed, it can not be altered without detection.
- Extensions can be used to tailor certificates to meet the needs of end applications.

# Digital Certificates

~~~~~  
~~~~~  
~~~~~  
**Digital  
Signature**

- Before two parties exchange data using Public Key cryptography, each wants to be sure that the other party is authenticated
- Before B accepts a message with A's Digital Signature, B wants to be sure that the public key belongs to A and not to someone masquerading as A on an open network
- One way to be sure, is to use a trusted third party to authenticate that the public key belongs to A. Such a party is known as a **Certification Authority (CA)**
- Once A has provided proof of identity, the Certification Authority creates a message containing A's name and public key. This message is known as a **Digital Certificate**.

# Public Key Certificate

- ❑ Contains name of user and user's public key (and possibly other info)
- ❑ Certificate is **signed** by the issuer (such as VeriSign) who vouches for it
- ❑ Signature on certificate is verified using signer's public key

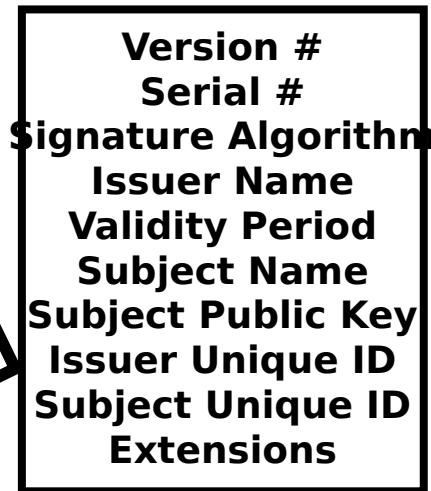
# Certificate Authority

- Certificate authority (CA) is a trusted 3rd party (TTP) that issues and signs cert's
  - Verifying signature verifies the identity of the owner of corresponding private key
  - Verifying signature does **not** verify the identity of the source of certificate!
  - Certificates are public!
  - Big problem if CA makes a mistake (a CA once issued Microsoft certificate to someone else!)
  - Common format for certificates is X.509

# Digital Certificates

- A Digital Certificate is simply an X.509 defined data structure with a Digital Signature. The data represents who owns the certificate, who signed the certificate, and other relevant information

## X.509 Certificate



*Digital Signature*

- When the signature is generated by a Certification Authority (CA), the signature can be viewed as trusted.
- Since the data is signed, it can not be altered without detection.
- Extensions can be used to tailor certificates to meet the needs of end applications.

# How does a digital Certificate look like?

On whose name is the certificate issued?

What is the length of this public key?

Version

Serial

No.

Algorithm

Issuer

Validity

Public Key

NetScape®  
Certificate Management  
System

Enrollment Retrieval  
Certificate 0x014

[Check  
Request  
Status](#)

[List  
Certificates](#)

[Search  
Certificates](#)

[Import CA  
Certificate](#)

[Chain](#)

[Import](#)

[Certificate](#)

[Export](#)

[Certificate](#)

Certificate contents

Certificate:  
Data:

Version: v3

Serial Number: 0x14

Signature Algorithm: SHA1withRSA - 1.2.840.113549.1.1.5

Issuer: CN=Certificate Manager,OU=Internet Security Group,O=Orisis,

Validity:

Before: Wednesday, February 27, 2002 6:08:39 PM IST

After: Thursday, February 27, 2003 6:08:39 PM IST

E=akahate@indiatimes.com,CN=Atul Kahate,UID=akahate,OU=Per

Public Key Info:

Algorithm: RSA - 1.2.840.113549.1.1.1

Public Key:

Exponent: 65537

Public Key Modulus: (1024 bits) :

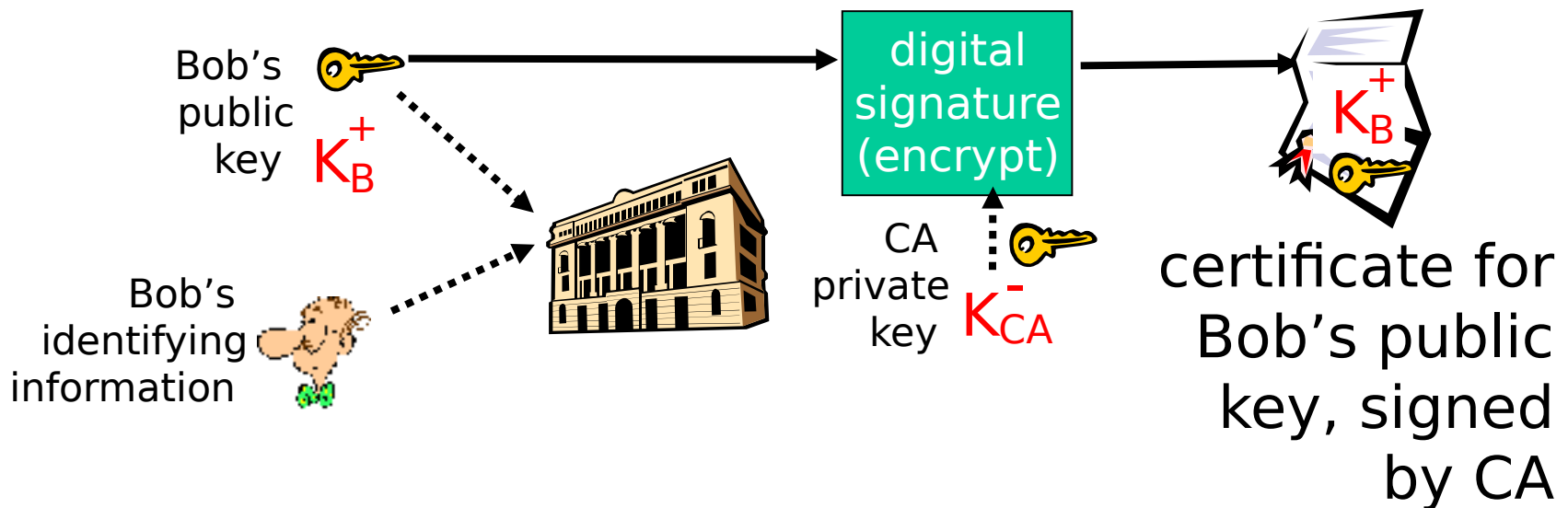
C8:15:3E:E6:75:C1:9D:9B:26:06:95:CB:7C:8D:ED:C3:  
3F:6D:76:3C:BD:8D:36:CA:F7:C5:4A:17:D4:F8:D4:82:  
02:B7:0D:54:A7:5E:8B:BC:B2:C9:80:5F:86:96:59:44:  
BA:AE:7B:78:0E:45:53:04:8A:A5:1D:8A:ED:C1:A8:53:

16  
Numbers  
per row x 8  
rows x 8  
bits per two  
digit  
Hexadecim

- We will learn more about it too, later

# Certification Authorities

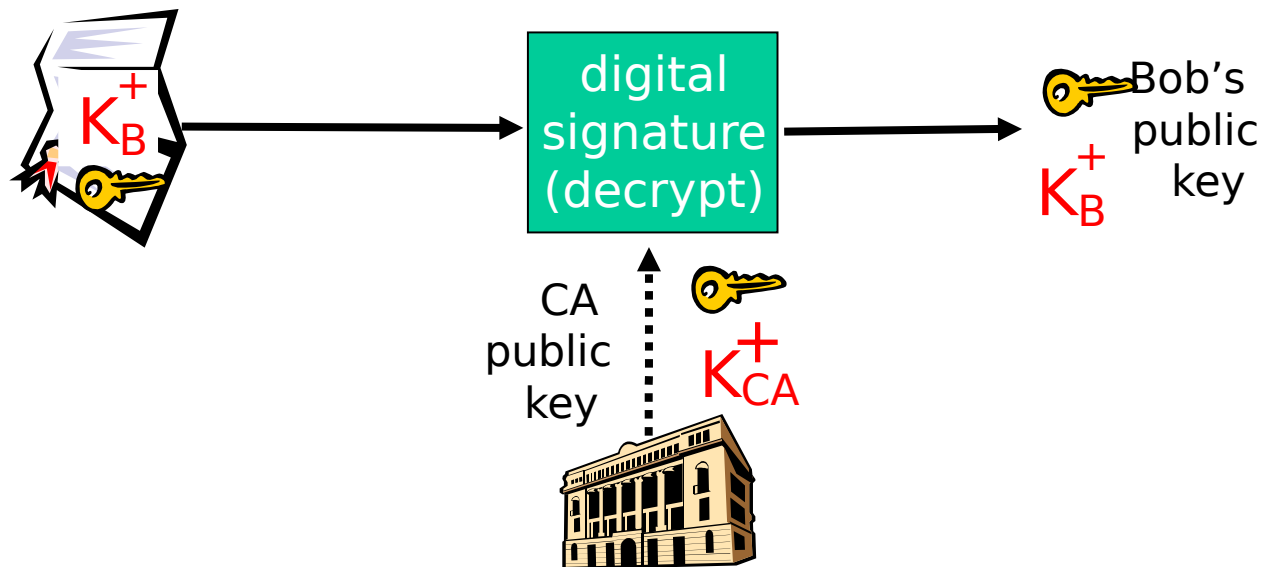
- ❑ **Certification authority (CA):** binds public key to particular entity, E.
- ❑ E (person, router) registers its public key with CA.
  - o E provides “proof of identity” to CA.
  - o CA creates certificate binding E to its public key.
  - o certificate containing E’s public key digitally signed by CA
    - CA says “this is E’s public key”





# Certification Authorities

- When Alice wants Bob's public key:
  - o gets Bob's certificate (Bob or elsewhere).
  - o apply CA's public key to Bob's certificate, get Bob's public key



# Public Key Infrastructure

# PKI

- ❑ Public Key Infrastructure (PKI) consists of all pieces needed to securely use public key cryptography
  - o Key generation and management
  - o Certificate authorities
  - o Certificate revocation (CRLs), etc.
- ❑ No general standard for PKI
- ❑ We consider a few “trust models”

# PKI Players

- Registration Authority (RA) to identity proof users
- Certification Authorities (CA) to issue certificates and CRL's
- Repositories (publicly available databases) to hold certificates and CRLs

# Certification Authority (CA)

## ***Certification Authority***

- Trusted (Third) Party
- Enrolls and Validates Subscribers
- Issues and Manages Certificates
- Manages Revocation and Renewal of Certificates
- Establishes Policies & Procedures

**Certification Authority = Basis of Trust**

# Registration Authority (RA)

- Enrolling, de-enrolling, and approving or rejecting requested changes to the certificate attributes of subscribers.
- Validating certificate applications.
- Authorizing requests for key-pair or certificate generation and requests for the recovery of backed-up keys.
- Accepting and authorizing requests for certificate revocation or suspension.
- Physically distributing personal tokens to and recovering obsolete tokens from people authorized to hold and use them.

# Certificate Policy (CP) is ...

- the basis for trust between unrelated entities
- not a formal “contract” (but implied)
- a framework that both informs and constrains a PKI implementation
- a statement of what a certificate means
- a set of rules for certificate holders
- a way of giving advice to Relying

# Certificate Revocation Lists

- ❑ CA periodically publishes a data structure called a certificate revocation list (CRL).
- ❑ Described in X.509 standard.
- ❑ Each revoked certificate is identified in a CRL by its serial number.
- ❑ CRL might be distributed by posting at known Web URL or from CA's own X.500 directory entry.



# PKI Trust Models

- ❑ Monopoly model
  - o One universally trusted organization is the CA for the known universe
  - o Favored by VeriSign
  - o Big problems if CA is ever compromised
  - o Big problem if you don't trust the CA!

# PKI Trust Models

- ❑ Oligarchy
  - o Multiple trusted CAs
  - o This approach used in browsers today
  - o Browser may have 80 or more certificates, just to verify signatures!
  - o User can decide which CAs to trust

# PKI Trust Models

- ❑ Anarchy model
  - o Everyone is a CA!
  - o Users must decide which “CAs” to trust
  - o This approach used in PGP (Web of trust)
  - o Why do they call it “anarchy”? Suppose cert. is signed by Frank and I don’t know Frank, but I do trust Bob and Bob says Alice is trustworthy and Alice vouches for Frank. Should I trust Frank?
- ❑ Many other PKI trust models

# From where can I get Digital Certificate



1

**Safescrypt Ltd.**  
II Floor, Tidel Park  
4 Canal Bank Road  
Taramani,

Chennai - 600 113



# From where can I get Digital Certificate



2

**National Informatics Centre**  
Ministry of Communications and  
Information Technology  
A-Block CGO Complex,  
Lodhi Road, New Delhi -110 003



# From where can I get Digital Certificate

**i-trust** PKI Services

IDRBT Certifying Authority

Licensed by Controller of Certifying Authorities, Government of India.

3

**Institute of Development & Research in Banking Technology (IDRBT)**

IDRBT, Road No. 1,  
Hyderabad,  
Andhra Pradesh - 500 057

Trust &

on

**INFINET**



CONTROLLER OF  
CERTIFYING AUTHORITIES





# From where can I get Digital Certificate



**CERTIFYING AUTHORITY**

Recognized by the controller of Certifying Authorities

**TATA CONSULTANCY SERVICES**

**4**

**Tata Consultancy Services Ltd.**  
IT Consulting and Software Service  
11th Floor, Air India Building,  
Nariman Point, Mumbai - 400 021



# From where can I get Digital Certificate



5

**Mahanagar Telephone Nigam Limited.**

Jeevan Bharati Tower-1, 3rd Floor,  
124, Connaught Circus,  
New Delhi, Pin-110001





# From where can I get Digital Certificate

**6**

**Director General,  
DG of Systems and Data  
Management,  
Customs and Central Excise.  
5th Floor, Hotel Samrat, Kautilya Marg,  
Chanakya Puri, New Delhi-110021**



# From where can I get Digital Certificate

**Gujarat Narmada Valley Fertilizers Company Limited**



**(n)Code Solutions (A division of Gujarat Narmada Valley Fertilisers Company Limited)**

301, GNFC Tower, Bodak Dev,  
Ahmedabad - 380 054, Gujarat, INDIA

