Asymmetric key Cryptography

Symmetric Encryption

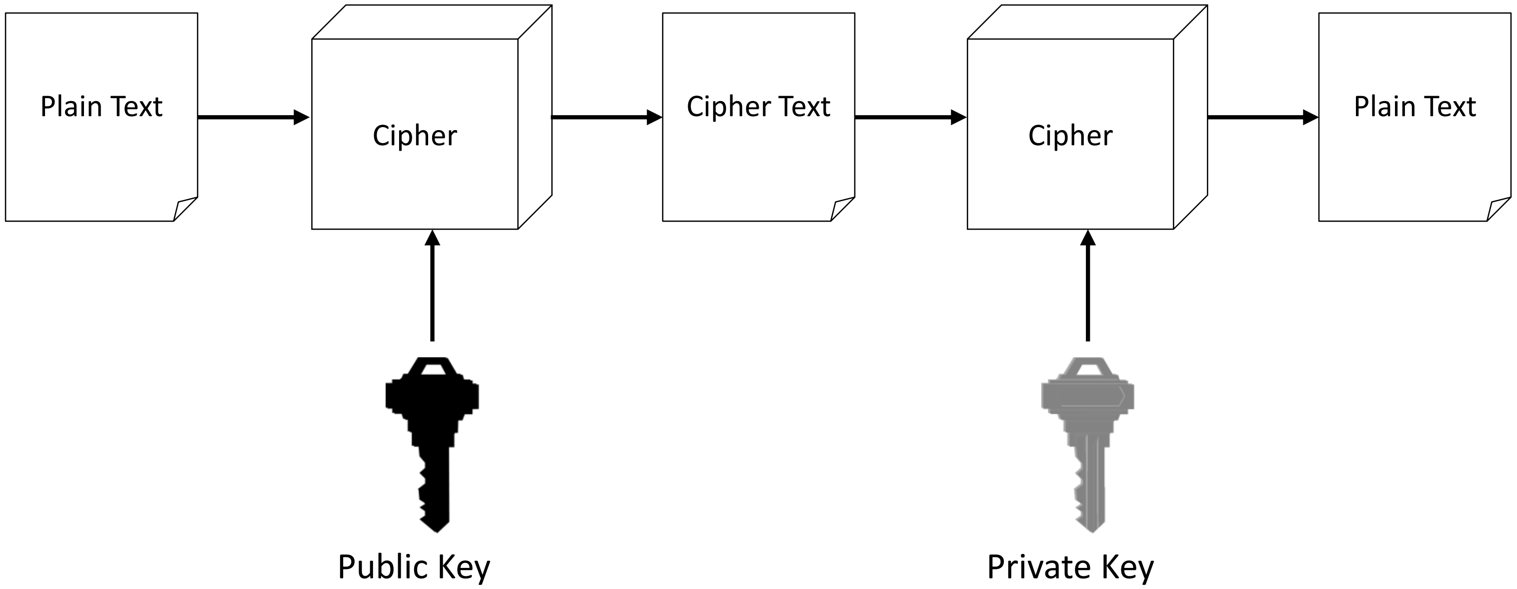
Limitations

- Any exposure to the secret key compromises secrecy of ciphertext
- A key needs to be delivered to the recipient of the coded message for it to be deciphered
  - Potential for eavesdropping attack during transmission of key

Asymmetric Encryption

Basics

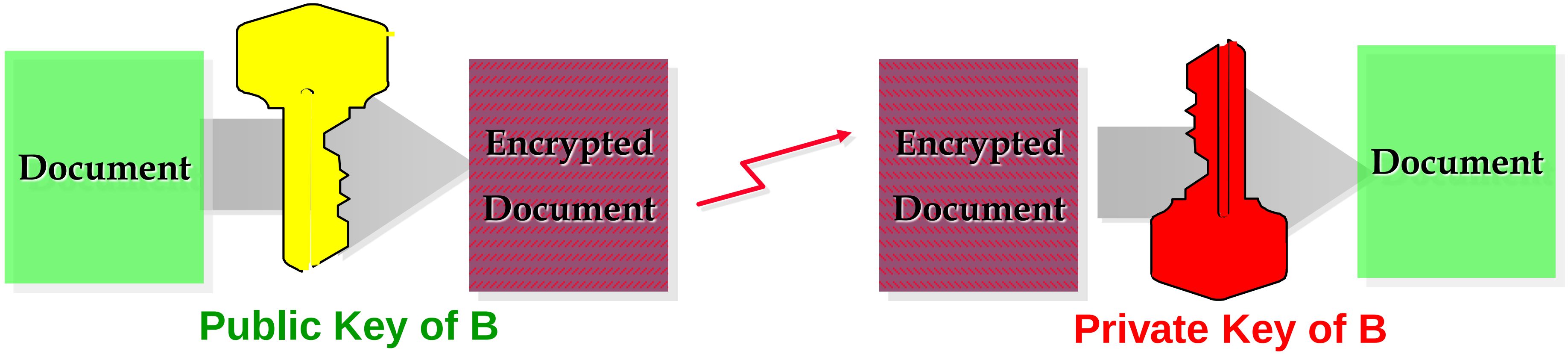
- Uses a pair of keys for encryption
  - Public key for encryption
  - Private key for decryption
- Messages encoded using public key can only be decoded by the private key
  - Secret transmission of key for decryption is not required
  - Every entity can generate a key pair and release its public key



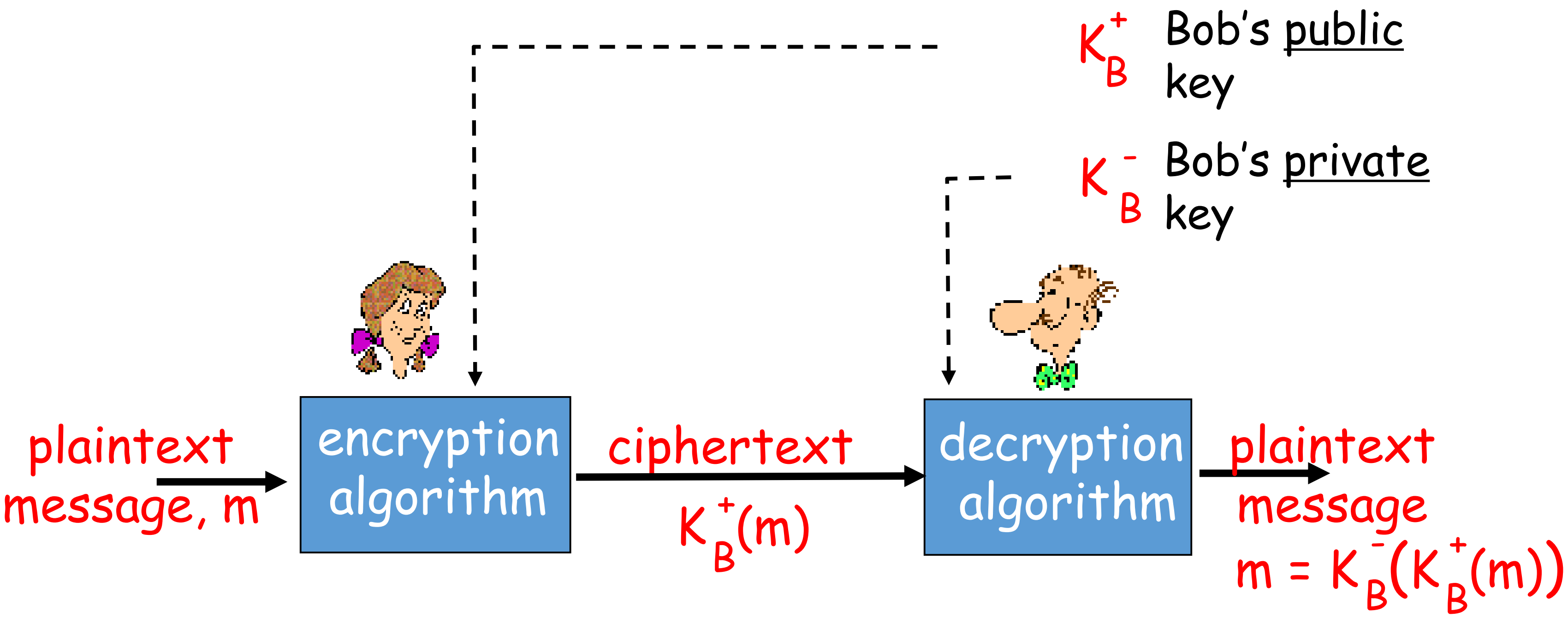
Public Key Cryptography

Encryption Technologies

Confidentiality



Public key cryptography



K_B^+ Bob's <u>public</u> key

K_B^- Bob's <u>private</u> key

plaintext message, m

encryption algorithm

ciphertext

$K_B^+(m)$

decryption algorithm

plaintext message

$m = K_B^-(K_B^+(m))$

The Process (revisited)

- A encrypts the message using his own private key. (Noone else knows A’s private key)

- A encrypts the message using B’s public key. (everyone knows B’s public key)

- A sends this message to B

The Process (revisited)

- B decrypts the message using his private key.
  (Only B knows his private key)

- B decrypts the message using A’s public key key.
  (Everyone knows A’s public key)

Applications in Judiciary

1. Instant posting of judgment on the web.
2. Secured electronic communications within judiciary
3. Authentic archiving of Judicial records
4. Submission of affidavits
5. Giving certified copies of the Judgment

Applications in Telecommunications

A. Subscribers

- Subscriber’s services management
  - STD/ISD, Opening, Closing, Initializing Password
- Shifting of telephones, Accessories (Clip, Cordless)
- Small Payments through telephones bills
  - Books, gifts, Internet purchases
- Mobile Authentication of SMS
  - Share market trading, Intra/Inter office instructions
- Mobile Phones as Credit cards
  - Mobile operator can venture into credit card business

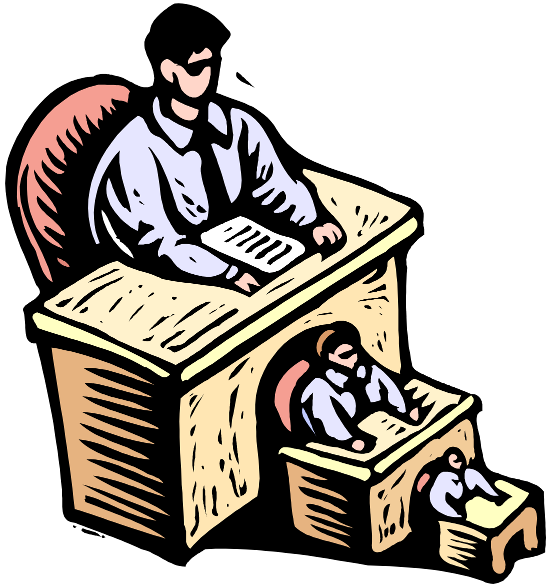
Applications in Telecommunications *(contd.)*

B. Internal

- Intra/Inter offices authentic communications
  - OBs, approvals, Instructions, requests
- Procurement of material
  - Calling/Receiving bids, Purchase orders, Payment instructions
- Network Management functions
  - Change of configuration, Blocking/unblocking routes

E-Governance

- Empowering Citizens
  a) Transparency
  b) Accountability
  c) Elimination of Intermediatory
  d) Encouraging Citizens to exercise their Rights



Government Online

1. Issuing forms and licences
2. Filing tax returns online
3. Online Government orders/treasury orders
4. Registration
5. Online file movement system
6. Public information records
7. E-voting
8. Railway reservations & ticketing
9. E-education
10. Online money orders

Asymmetric Encryption

Types

- Two most popular algorithms are RSA & El Gamal
  - RSA
    - Developed by Ron Rivest, Adi Shamir, Len Adelman
    - Both public and private key are interchangable
    - Variable Key Size (512, 1024, or 2048 buts)
    - Most popular public key algorithm
  - El Gamal
    - Developed by Taher ElGamal
    - Variable key size (512 or 1024 bits)
    - Less common than RSA, used in protocols like PGP

Public key encryption algorithms

Requirements:

1. need $K_B^+(\cdot)$ and $K_B^-(\cdot)$ such that

$$K_B^-(K_B^+(m)) = m$$

2. given public key K_B^+, it should be impossible to compute private key K_B^-

RSA: Rivest, Shamir, Adelson algorithm

RSA: Choosing keys

1. Choose two large prime numbers *p, q.* (e.g., 1024 bits each)

2. Compute $n = pq, \quad z = (p-1)(q-1)$

3. Choose e (with $e<n$) that has no common factors with z. (e, z are "relatively prime").

4. Choose d such that $ed-1$ is exactly divisible by z. (in other words: $ed \bmod z = 1$).

5. *Public* key is $\underbrace{(n,e)}_{K_B^+}$. *Private* key is $\underbrace{(n,d)}_{K_B^-}$.

RSA: Encryption, decryption

0. Given (n,e) and (n,d) as computed above

1. To encrypt bit pattern, m, compute

   $c = m^e \bmod n$ (i.e., remainder when m^e is divided by n)

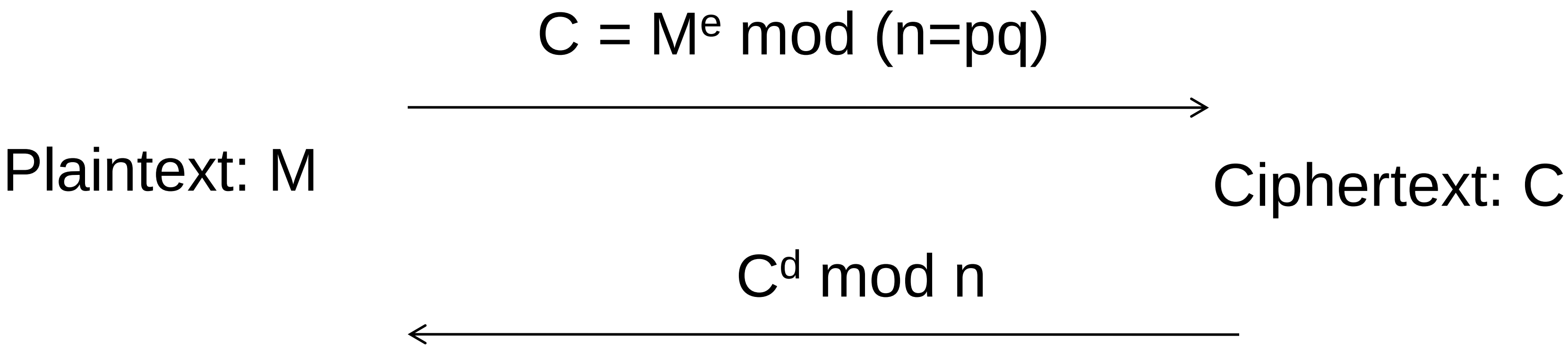
2. To decrypt received bit pattern, c, compute

   $m = c^d \bmod n$ (i.e., remainder when c^d is divided by n)

Magic happens!

$$m = (\underbrace{m^e \bmod n}_{c})^d \bmod n$$

RSA in action



$C = M^e \bmod (n=pq)$

Plaintext: M

Ciphertext: C

$C^d \bmod n$

From n, difficult to figure out p,q

From (n,e), difficult to figure d.

From (n,e) and C, difficult to figure out M s.t. $C = M^e$

RSA example:

Bob chooses $p=5$, $q=7$. Then $n=35$, $z=24$.

$e=5$ (so e, z relatively prime).

$d=29$ (ed mod(z) = 1, or $ed-1$ exactly divisible by z).

Keys generated are

Public key: (35,5)

Private key is (35, 29)

	letter	m	m^e	$c = m^e \bmod n$
encrypt:	L	12	248832	17

	c	c^d	$m = c^d \bmod n$	letter
decrypt:	17	481968572106750915091411825223071697	12	L

Example Contd..

- Encrypt the word love using ($c = m^e \bmod n$)
  - Assume that the alphabets are between 1 & 26

Plain Text	Numeric Representation	m^e	Cipher Text ($c = m^e \bmod n$)
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

Asymmetric Encryption

RSA

- Decrypt the word love using ($m = c^d \bmod n$)
  - n = 35, c=29

Cipher Text	**c^d**	**($m = m^e \bmod n$)**	**Plain Text**
17	481968572106750915091411825223072000	17	l
15	12783403948858939111232757568359400	15	o
22	8526433190865377019561944997211100000000	22	v
10	100000000000000000000000000000	10	e

Example 2

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

RSA: another important property

The following property will be *very* useful later:

$$\underbrace{K_B^-(K_B^+(m))}_{\substack{\text{use public key}\\ \text{first, followed}\\ \text{by private key}}} = m = \underbrace{K_B^+(K_B^-(m))}_{\substack{\text{use private key}\\ \text{first, followed}\\ \text{by public key}}}$$

Result is the same!
This leads to
Digital Signatures

Digital Signatures: The Problem

- Consider the real-life example where a person pays by credit card and signs a bill; the seller verifies that the signature on the bill is the same with the signature on the card
- Contracts, they are valid if they are signed.
- Signatures provide non-repudiation.
  - ensuring that a party in a dispute cannot repudiate, or refute the validity of a statement or contract.
- Can we have a similar service in the electronic world?
  - Does Message Authentication Code provide non-repudiation? Why?

Digital signatures

- A digital signature
  - Allows the receiver to authenticate the identity of the sender
  - Prevents the sender from later claiming that he did not sent the message
  - Prevents the receiver from constructing the message that appears as if it came from the sender

Digital Signatures and Hash

- Very often digital signatures are used with hash functions, hash of a message is signed, instead of the message.

3.4 HASH FUNCTIONS

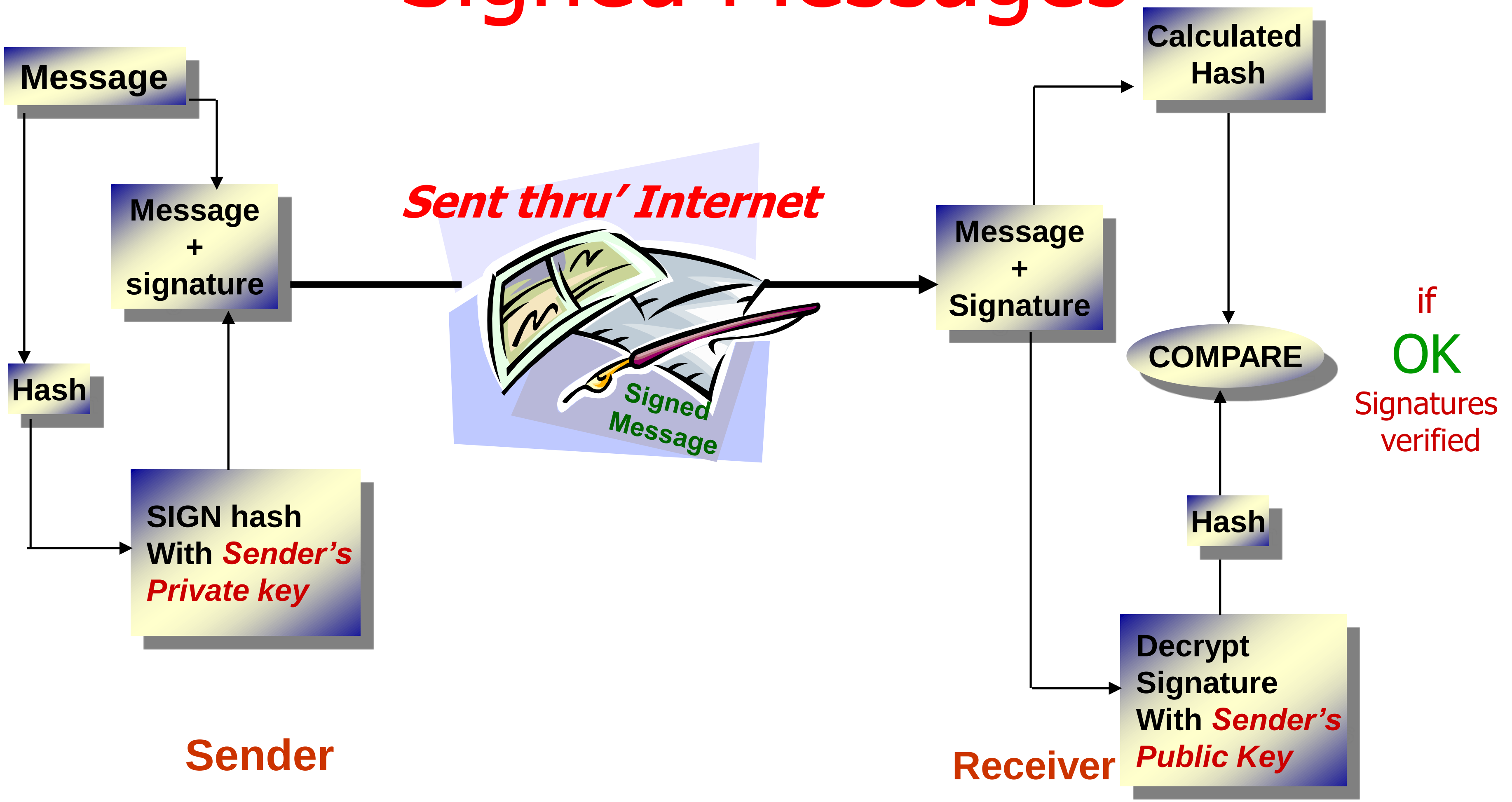
- Condenses arbitrary message to fixed size

  $h = H(M)$
- Usually assume hash function is public
- Hash used to detect changes to message
- Want a cryptographic hash function
  - Computationally infeasible to find data mapping to specific hash (one-way property)
  - Computationally infeasible to find two data to same hash (collision-free property)

Digital Signatures

- Digital signatures: One party generates signature, many parties can verify.
- Digital Signature: a data string which associates a message with some originating entity.
- Digital Signature Scheme:
  - a signing algorithm: takes a message and a (private) signing key, outputs a signature
  - a verification algorithm: takes a (public) key verification key, a message, and a signature
- Provides:
  - Authentication, Data integrity, Non-Repudiation

Signed Messages



Message

Message + signature

Hash

SIGN hash With *Sender's Private key*

Sent thru' Internet

Message + Signature

Calculated Hash

COMPARE

if OK Signatures verified

Hash

Decrypt Signature With *Sender's Public Key*

Sender

Receiver

Digital signatures

- Step 1: A encrypts the plaintext (PT1) to ciphertext (CT1) using B’s public key

  PT1 → CT1

- Step 2: A creates a message digest by hashing and then the digital signature by encrypting the digest with A’s private key.

  PT1 → MD1 → DS1

- Step 3: A sends both the ciphertext (CT1) and digital signature (DS1) to B. B receives both.

A’s end	B’s end
CT1 DS1	CT2 DS2

Digital signatures

- Step 4: B decrypts ciphertext received in step 3 by using B's private key to get the original plaintext message.

  CT2 → PT2

- How do we know PT2 = PT1?

- Comparing PT1 and PT2 is not a wise thing.

- Step 5: B obtains a message digest (MD2) by decrypting A's digital signature received in step 3 by using A's public key. Hope MD1 = MD2.

- Step 6: B creates its own message digest (MD3) using the same hashing algorithm on the plaintext message (PT2). If MD2 = MD3 B concludes that the message must have come from A and it has not been tempered with.

Requirements of PKC

- Computationally easy
  - To generate public and private key pair
  - To encrypt the message using encryption key
  - To decrypt the message using decryption key
- Computational infeasible
  - To compute the private key using public key
  - To recover the plaintext using ciphertext and public key
- The encryption and decryption can be applied in either order

Trapdoor One Way Function

- PKC boils down to need for trapdoor one way function
- One way function:
  - Maps a domain into a range st every function value has a unique inverse
  - The calculation of the function is easy
    - $Y = f(X)$ easy
  - The calculation of the inverse is infeasible
    - $X = f'(Y)$ infeasible

Trapdoor One-way Function

- Trapdoor one way function
  - Easy to compute in one direction
    - $Y=f_k(X)$
  - Infeasible in other direction if k is unknown
    - $X = f'_k(y)$
  - The calculation of the inverse is easy if the key is known
    - $X = f'_k(y)$

Possible Attacks

- Brute force
  - Use large keys
    - Trade-off: speed (not linearly depend on key size)
    - Confined to small data encryption: signature, key management
- Compute the private key from public key
  - Not proven that is not feasible for most protocols!
- Probable message attack
  - Encrypt all possible messages using encryption key
  - Compare with the ciphertext to find the matched one!
  - If data is small, feasible, regardless of key size of PKC

Asymmetric Encryption

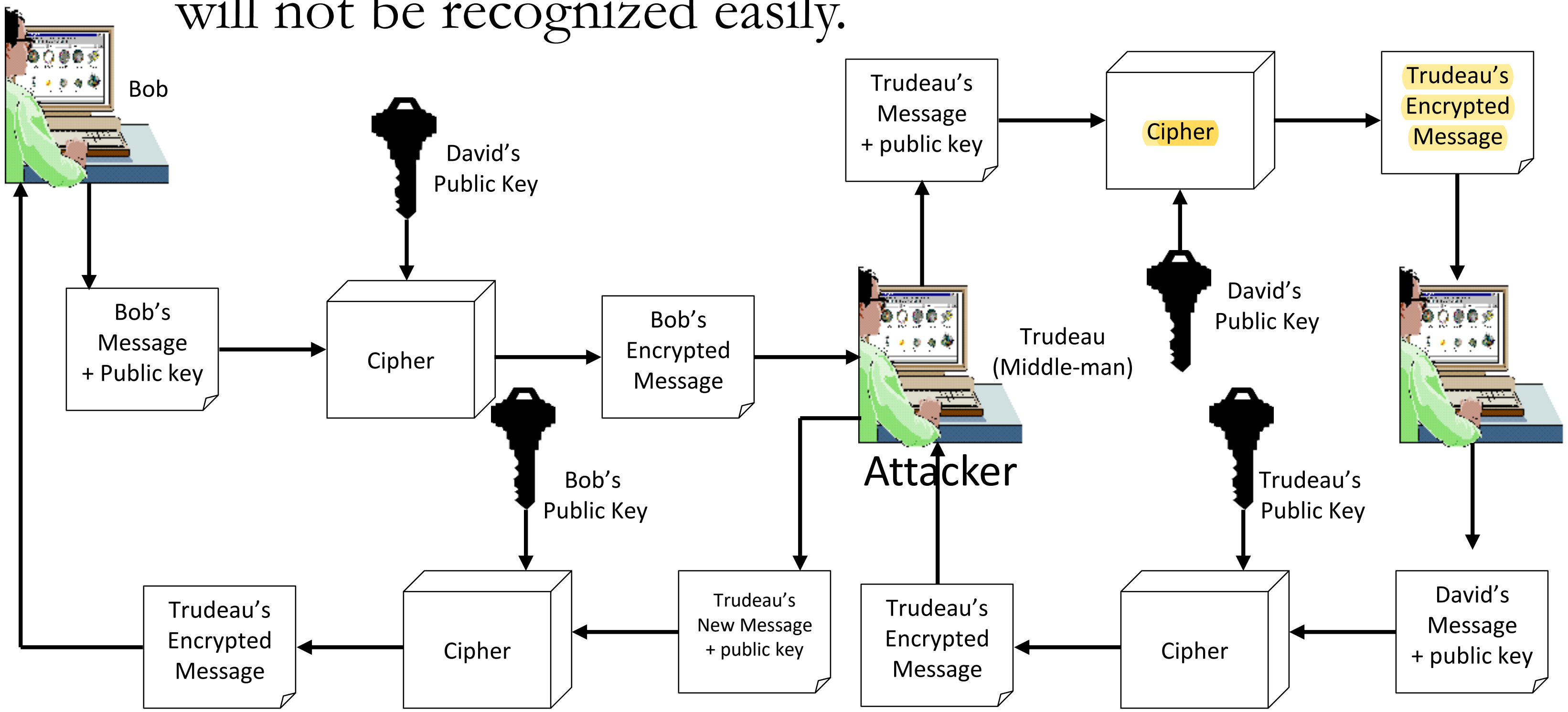
Weaknesses

- Efficiency is lower than Symmetric Algorithms
  - A 1024-bit asymmetric key is equivalent to 128-bit symmetric key
- Potential for man-in-the middle attack
- It is problematic to get the key pair generated for the encryption

Asymmetric Encryption

Man-in-the-middle Attack

- Hacker could generate a key pair, give the public key away and tell everybody, that it belongs to somebody else. Now, everyone believing it will use this key for encryption, resulting in the hacker being able to read the messages. If he encrypts the messages again with the public key of the real recipient, he will not be recognized easily.



Advantages and Disadvantages of Cryptographic Systems

Characteristic	Symmetric	Asymmetric
Key used	Same key is used	Two different keys
Speed of encryption/decryption	Very fast	Slower
Size of resulting encrypted text	Usually same as or less than the original size	More than the original plain text size
Key agreement/exchange	A big problem	No problem
Number of keys required	Equals about the square of the number of participants, so scalability is an issue	Same as the number of participants
Usage	Mainly for encryption and decryption (confidentiality), cannot be used of DS (integrity and non-repudiability)	Can be used for encryption and decryption (confidentiality) as well as for DS (integrity and non-repudiability)