

# Cryptography

Reference:  
Network Security  
PRIVATE Communication in a PUBLIC World.  
by Kaufman, Perlman & Speciner.

---

---

---

---

---

---

---

---

# Secret Key Cryptography

- Single key used to encrypt and decrypt.
- Key must be known by both parties.
- Assuming we live in a hostile environment (otherwise - why the need for cryptography?), it may be hard to share a secret key.

---

---

---

---

---

---

---

---

# Public Key Cryptography (a.k.a. asymmetric cryptography)

- Relatively new field - 1975 (as far as we know, the NSA is not talking).
- Each entity has 2 keys:
  - private key (a secret)
  - public key (well known).

---

---

---

---

---

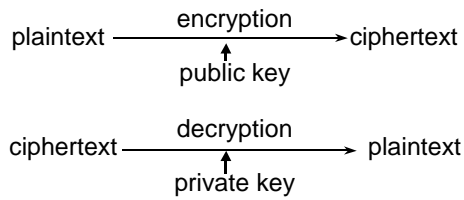
---

---

---

## Using Keys

- Private keys are used for decrypting.
- Public keys are used for encrypting.



Netprog: Cryptography

4

---

---

---

---

---

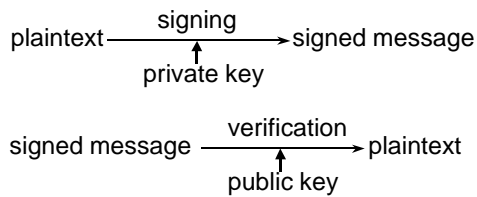
---

---

---

## Digital Signature

- Public key cryptography is also used to provide digital signatures.



Netprog: Cryptography

5

---

---

---

---

---

---

---

---

## Transmitting over an insecure channel.

Alice wants to send Bob a private message.

$A_{\text{public}}$  is Alice's public key.

$A_{\text{private}}$  is Alice's private key.

$B_{\text{public}}$  is Bob's public key.

$B_{\text{private}}$  is Bob's private key.

Netprog: Cryptography

6

---

---

---

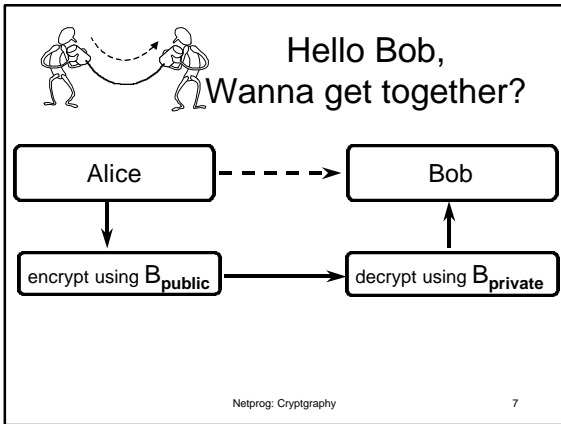
---

---

---

---

---




---

---

---

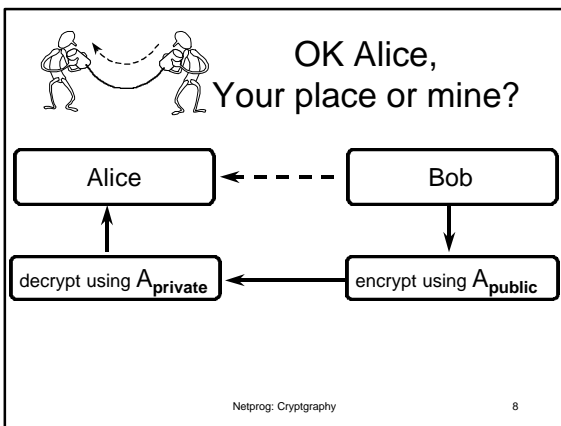
---

---

---

---

---




---

---

---

---

---

---

---

---

### Bob's Dilemma

- Nobody can read the message from Alice, but anyone could produce it.
- How does Bob know that the message was really sent from Alice?
- Bob may be comforted to know that only Alice can read his reply.

Netprog: Cryptography 9

---

---

---

---

---

---

---

---

## Alice can sign her message!

- Alice can create a digital signature and prove she sent the message (or someone with knowledge of her private key).
- The signature can be a message digest encrypted with  $A_{\text{private}}$ .

---

---

---

---

---

---

---

---

## Message Digest

- Also known as “hash function” or “one-way transformation”.
- Transforms a message of any length and computes a fixed length string.
- We want it to be hard to guess what the message was given only the digest.
  - Guessing is always possible.

---

---

---

---

---

---

---

---

## Alice's Signature

- Alice feeds her original message through a hash function and encrypts the message digest with  $A_{\text{private}}$ .
- Bob can decrypt the message digest using  $A_{\text{public}}$ .
- Bob can compute the message digest himself.
- If the 2 message digests are identical, Bob knows Alice sent the message.

---

---

---

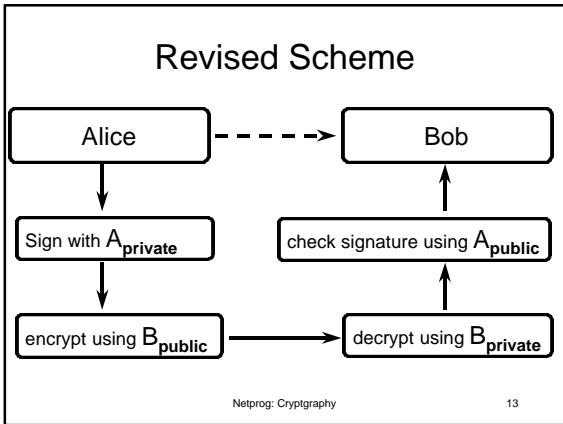
---

---

---

---

---




---

---

---

---

---

---

---

---

### Why the digest?

- Alice could just encrypt her name, and then Bob could decrypt it with  $A_{\text{public}}$ .
- Why wouldn't this be sufficient?

Netprog: Cryptography 14

---

---

---

---

---

---

---

---

### Implications

- Suppose Alice denies she sent the message?
- Bob can prove that only someone with Alice's key could have produced the message.

Netprog: Cryptography 15

---

---

---

---

---

---

---

---

## Another possible problem

- Suppose Bill receives a message from Alice including a digital signature.  
*"meet me at the library tonight"*
- Bill sends the same message to Joe so that it looks like the message came from Alice.
- Bill includes the digital signature from the message Alice sent to him.
- Joe is convinced Alice sent the message!

---

---

---

---

---

---

---

---

## Solution?

- Always start your messages with:
  - Dear Bill,
- Create a digest from the encrypted message and sign that digest.
- There are many other schemes as well.

---

---

---

---

---

---

---

---

## Speed

- Secret key encryption/decryption algorithms are much faster than public key algorithms.
- Many times a combination is used:
  - use public key cryptography to share a secret key.
  - use the secret key to encrypt the bulk of the communication.

---

---

---

---

---

---

---

---

## Secure Protocols

- There are a growing number of applications for secure protocols:
  - email
  - electronic commerce
  - electronic voting
  - homework submission

---

---

---

---

---

---

---

---

## Secure Protocols

- Many application protocols include the use of cryptography as part of the application level protocol.
  - The cryptographic scheme employed is part of the protocol.
  - If stronger cryptographic tools become available we need to change the protocol.

---

---

---

---

---

---

---

---

## SSL and TLS

- Secure Sockets Layer (SSL) is a different approach - a new layer is added that provides a secure channel over a TCP only link.
- TLS is Transport Layer Security (IETF standard based on SSL).

---

---

---

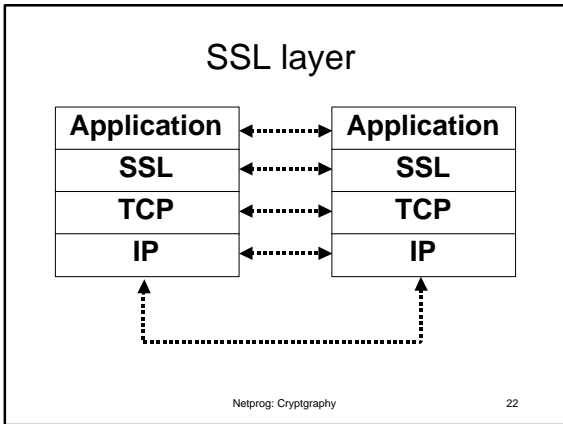
---

---

---

---

---




---

---

---

---

---

---

---

---

- ### Advantages of SSL/TLS
- Independent of application layer
  - Includes support for negotiated encryption techniques.
    - easy to add new techniques.
  - Possible to switch encryption algorithms in the middle of a session.
- Netprog: Cryptography 23

---

---

---

---

---

---

---

---

- ### HTTPS Usage
- HTTPS is HTTP running over SSL.
    - used for most secure web transactions.
    - HTTPS server usually runs on port 443.
    - Include notion of verification of server via a certificate.
    - Central trusted source of certificates.
- Netprog: Cryptography 24

---

---

---

---

---

---

---

---