

Cryptography

Reference:

Network Security

PRIVATE Communication in a PUBLIC World.

by Kaufman, Perlman & Speciner.

Secret Key Cryptography

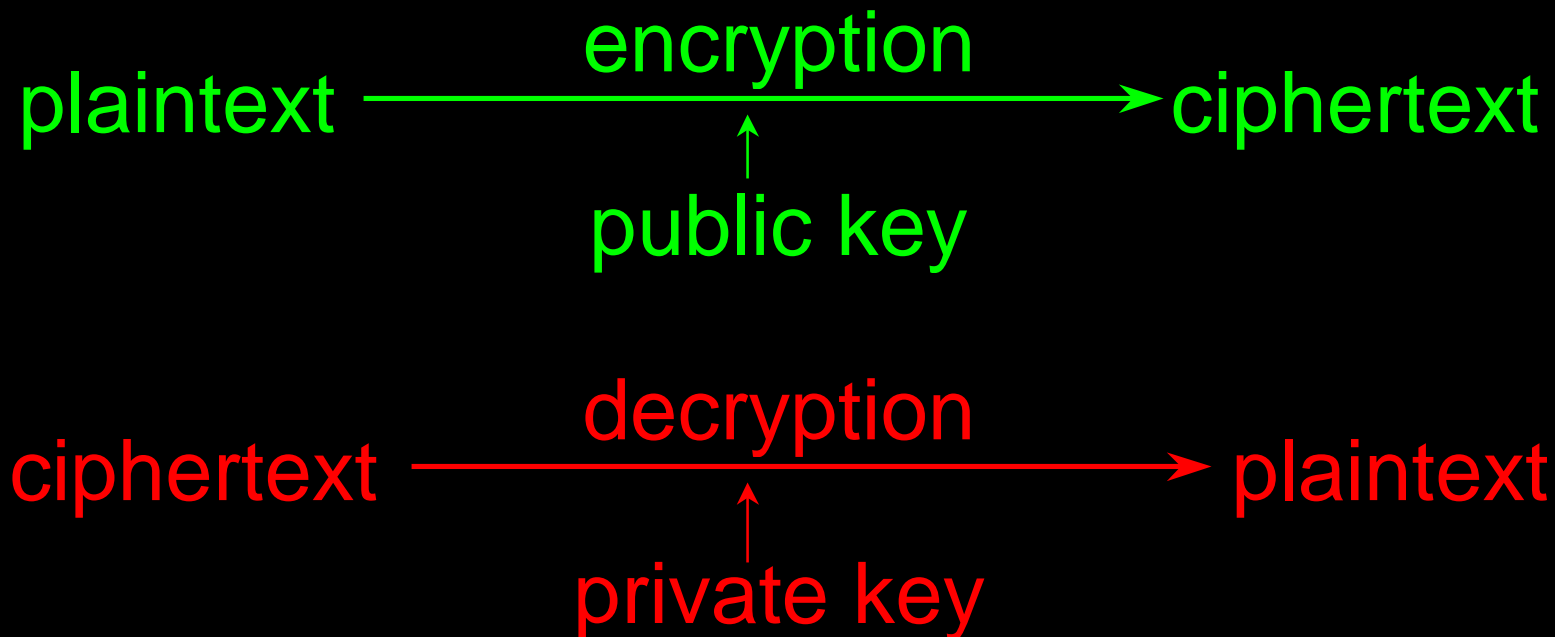
- ◆ Single key used to encrypt and decrypt.
- ◆ Key must be known by both parties.
- ◆ Assuming we live in a hostile environment (otherwise - why the need for cryptography?), it may be hard to share a secret key.

Public Key Cryptography

(a.k.a. asymmetric cryptography)

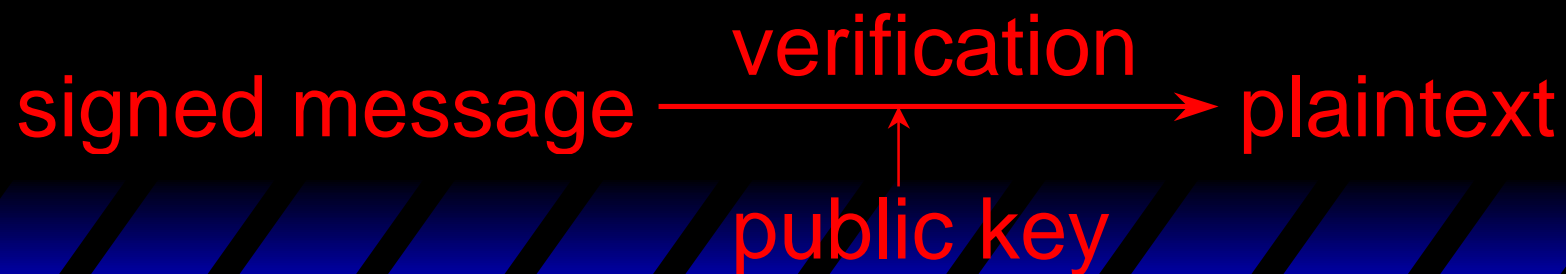
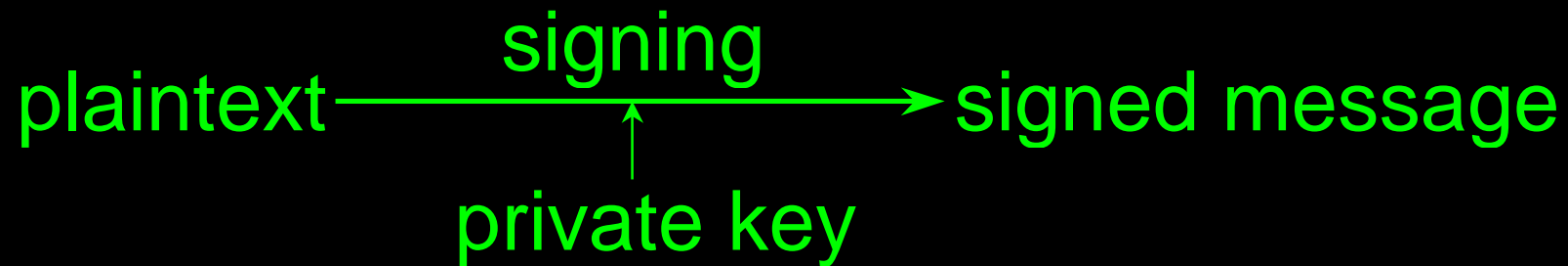
- ◆ Relatively new field - 1975 (as far as we know, the NSA is not talking).
- ◆ Each entity has 2 keys:
 - private key (a secret)
 - public key (well known).

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



Digital Signature

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



Transmitting over an insecure channel.

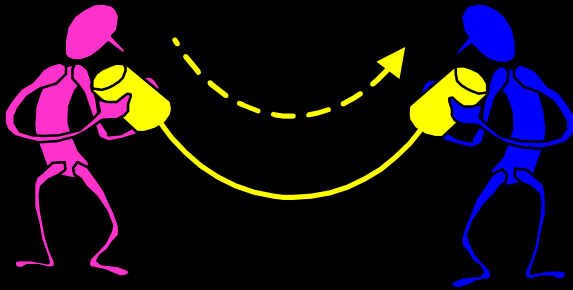
Alice wants to send Bob a private message.

A_{public} is Alice's public key.

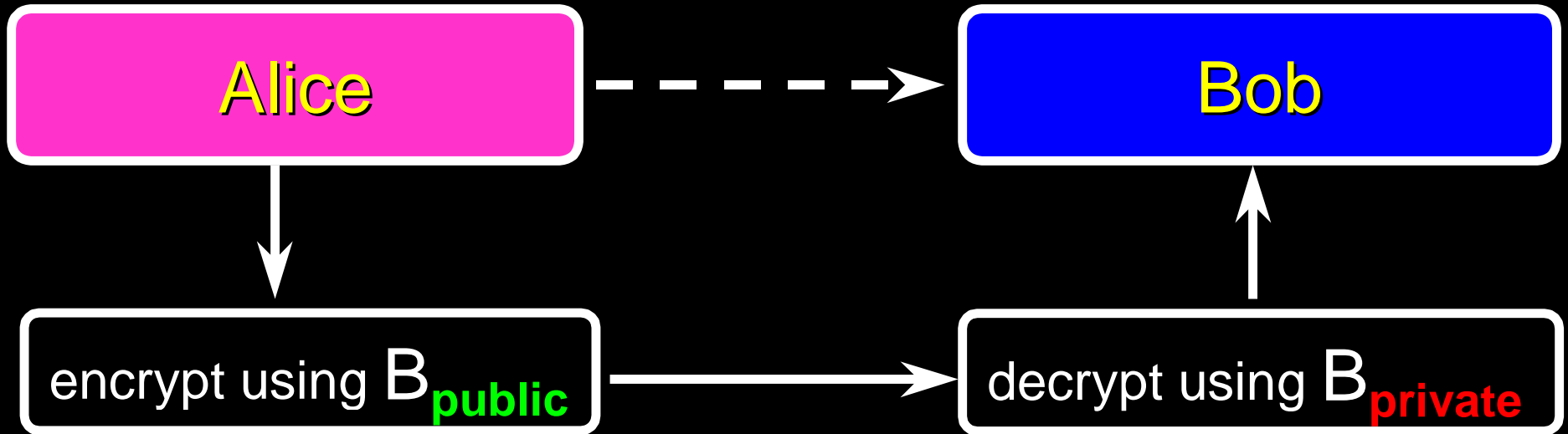
A_{private} is Alice's private key.

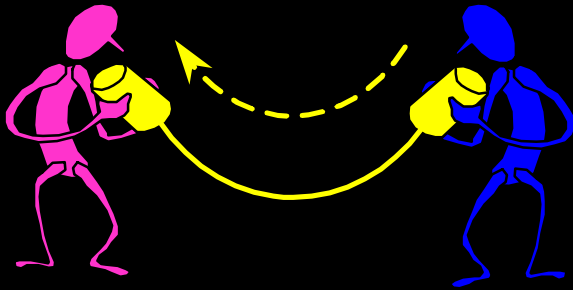
B_{public} is Bob's public key.

B_{private} is Bob's private key.

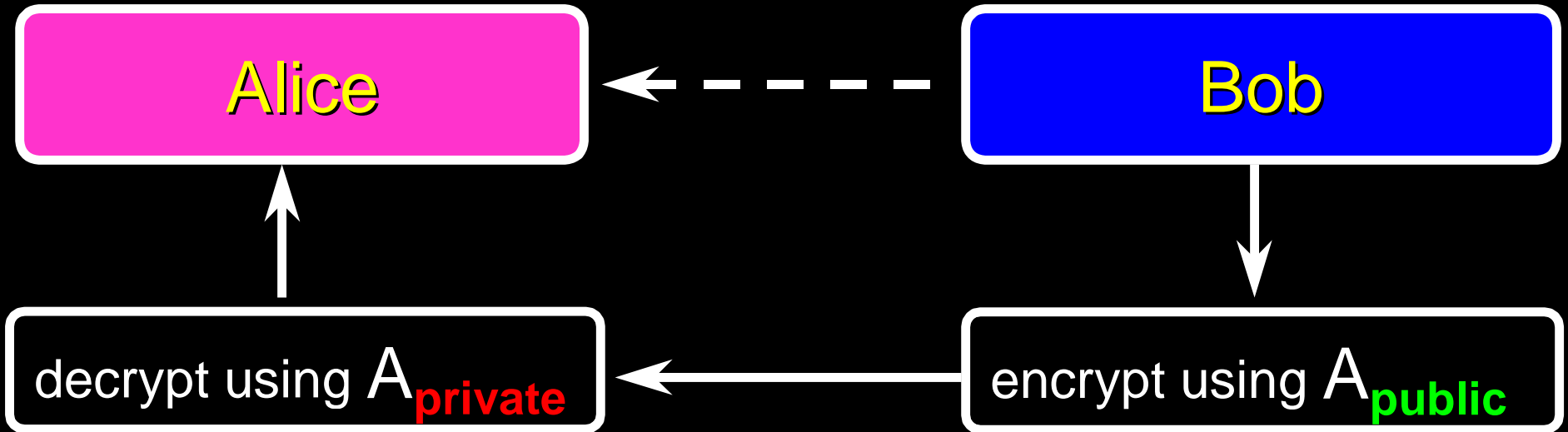


Hello Bob,
Wanna get together?





OK Alice,
Your place or mine?



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.

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_{private} .

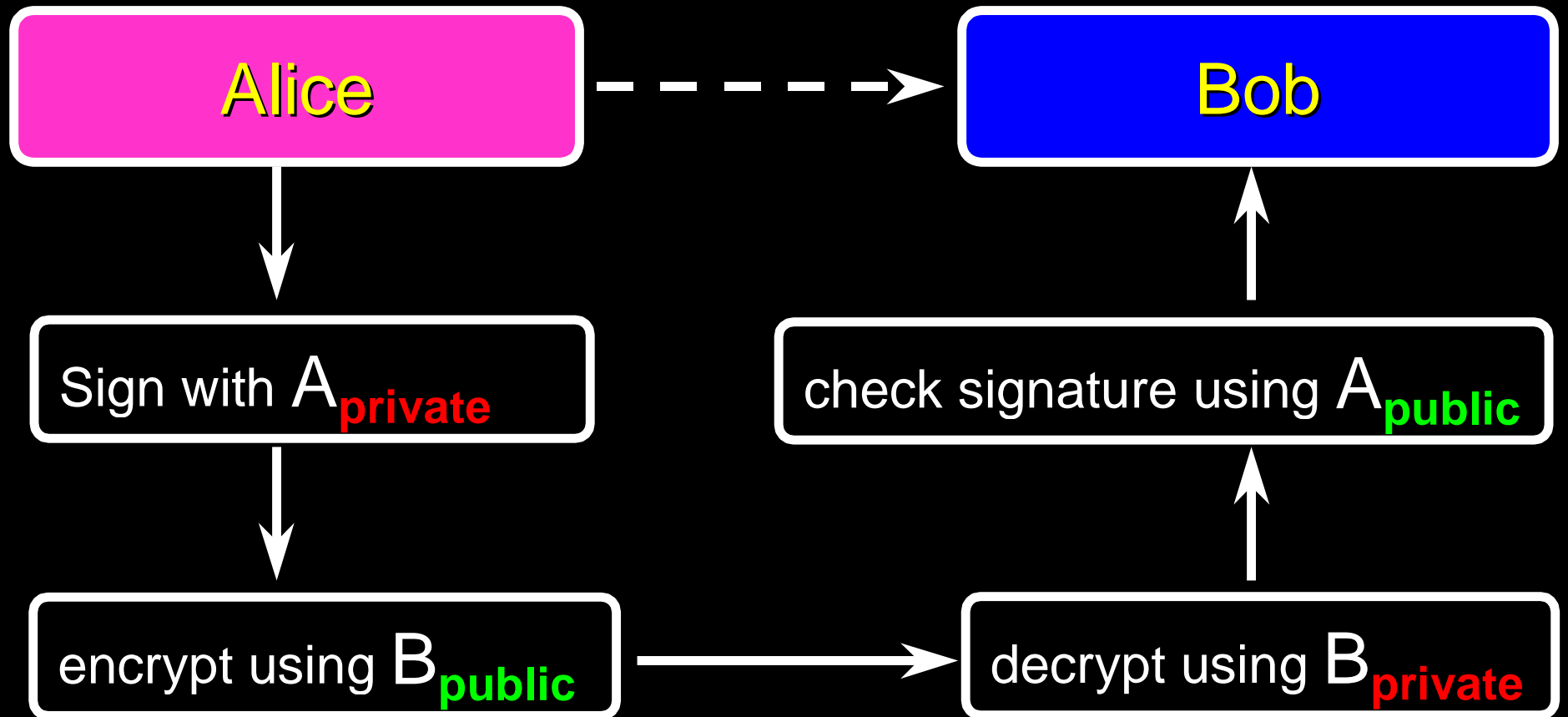
Message Digest

- ◆ 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_{private} .
- ◆ Bob can decrypt the message digest using A_{public} .
- ◆ Bob can compute the message digest himself.
- ◆ If the 2 message digests are identical Bob knows Alice sent the message.

Revised Scheme



Why the digest?

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

Implications

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

Another possible problem

- ◆ Suppose O.J. receives a message from Alice including a digital signature.
- ◆ O.J. sends the same message to Bob claiming to be Alice, and includes the digital signature from the message Alice sent to him.
- ◆ Bob is convinced Alice sent the message!

Solution?

- ◆ Always start your messages with:
 - Dear O.J.
- ◆ 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.
- ◆ Secure Sockets Layer (SSL) is a different approach - a new layer is added that provides a secure channel (currently TCP only).
- ◆ Most secure WWW communication is SSL based.

Security

- ◆ There is a course on security offered - much more in depth.
- ◆ I have lots of books for anyone interested in secure protocols, encryption standards and algorithms, etc.