Encryption

Todays topics

Hey, check out these slides...

- What is ***encryption***
- What is a ***cryptosystem***
- Types of encryption
  - Symmetric encryption
  - Asymmetric encryption
- Public and Private keys
- Key Exchanges

What is Encryption

Encryption is the process of *encoding* information with a purpose (typically for security security). This process converts the original representation of the information, known as *plaintext*, into an alternative form known as *ciphertext*.

This process, combined with its reverse *decryption*, and all of the data/algorithms necessary to encrypt and decrypt messages form a *cryptosystem*.

-*Wikipedia*

Encoding and Encrpytion

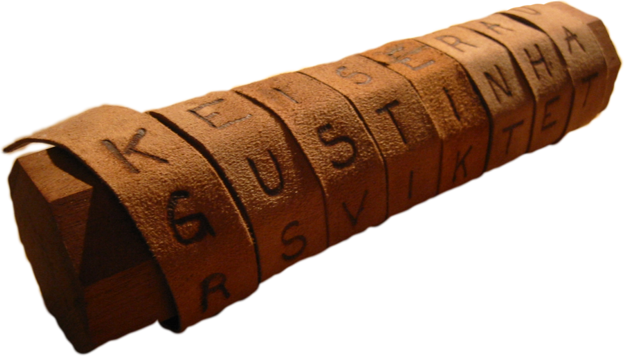


Figure 1: Scytale

What is Encryption (continued)

Some points:

- Encryption is reversible (with a key)
- Encryption is everywhere
- Encryption does NOT need a computer (but good encryption probably does. . .)

Cryptosystem

Five-tuple (sequence) of the following elements:

Element	description
E	the set of ***Encryption*** algorithms
D	the set of ***Decryption*** algorithms
M	the set of plaintext ***Messages***
K	the set of ***Keys***
C	the set of encrypted messages or ***Ciphertexts***

$\boldsymbol{E} = f(\ \boldsymbol{K},\boldsymbol{M}\ )$ -> $\boldsymbol{C}$

$\boldsymbol{D} = f(\ \boldsymbol{K},\boldsymbol{C}\ )$ -> $\boldsymbol{M}$

Scytale Cryptosystem

Element	description
E	wrap around scytale & write *message*
D	wrap around scytale & read *message*
M	the *message*
K	???
C	strip or paper with letters

Caesar Cipher

AKA rotational cipher, shift cipher, probably more. Simple encryption via substitution.

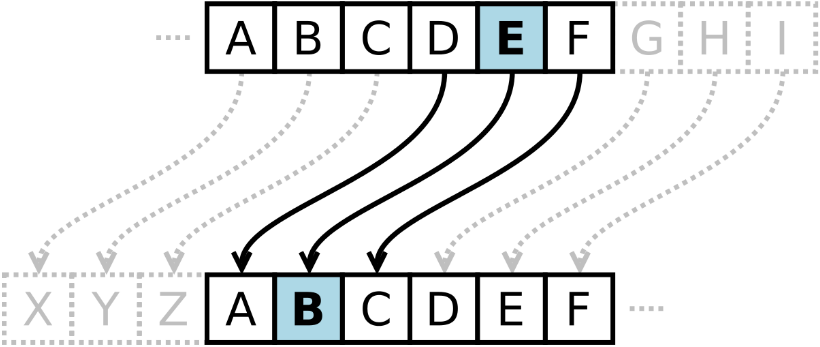


Figure 2: caesar cipher

Lets get Math-y

Element	description
M	***? ?? ????***
C	***F XJ EBOB***
E	***C***[n]=(***M***[n] - 3) Mod 26
D	***M***[n]=(***C***[n] + 3) Mod 26
K	What is the key?

```
M[n] = ABCDEFGHIJKLMNOPQRSTUVWXYZ
C[n] = XYZABCDEFGHIJKLMNOPQRSTUVW
```

Two types of Encryption

Symmetric

Same key for both encrypting and decrypting

Asymmetric

Different keys for encrypting and decrypting

Symmetric Encryption

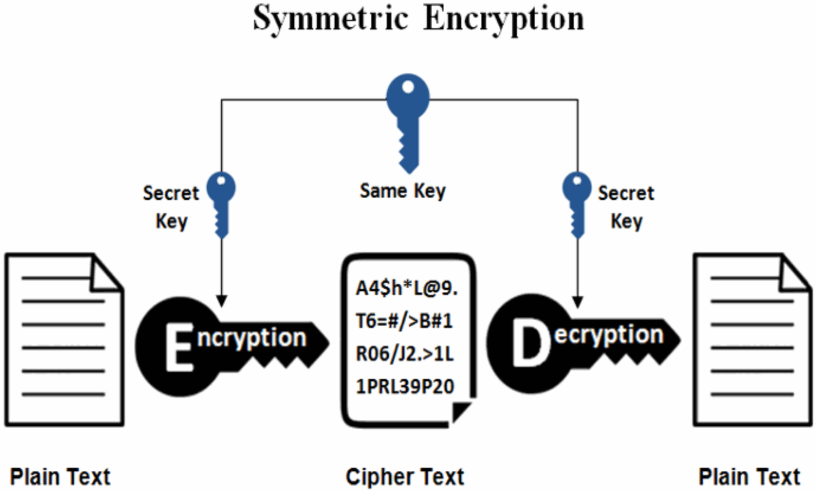


Figure 3: symmetric key encryption

Keys are identical for encryption and decryption.

Asymmetric Encryption

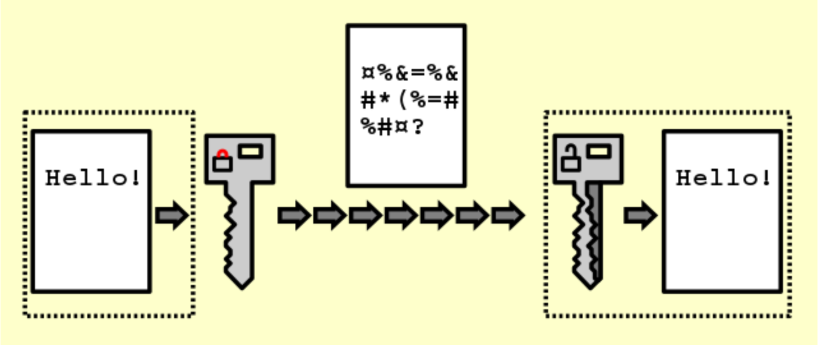


Figure 4: public key encryption

Keys are different but mathematically related.

Differences

Symmetric	Asymmetric
* One key * Fast * Used to encrypt various amounts of data * Key must be secured * Needs more keys to establish secure communications within a group	* Multiple keys * Slow * Used to authenticate and initiate symmetric encryption * only private key needs securing * Needs fewer keys to establish secure communication in a group

Public and Private Key pairs

- Public key is public information, can be shared with anyone.
- Assume **everyone** has your public key
- Private key must be kept private
- **Public** and **Private** key pairs share a unique mathematical relation:
  - Anything **encrypted** with the **public key** can only be *decrypted* with the *private key*
  - Anything **encrypted** with the **private key** can only be *decrypted* with the *public key*

Key Pairs continued

If we trust this relationship, and we trust that recorded knowledge of who has what keys:

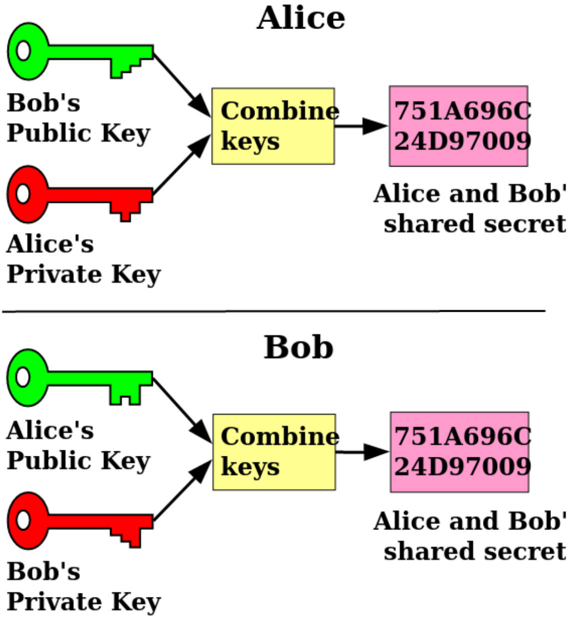
- We can check *Authenticity* by asking them to encrypt something with their private key
- We can recieve (one way) secure communications (ask someone else to first encrypt with your public key)
- If we send something encrypted with our private key who can read it???

Diffie-Hellman Key Exchange

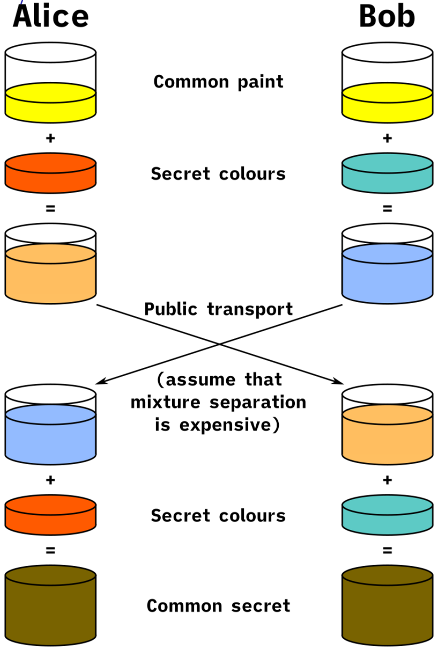
Solves the problem of securely exchanging cryptographic keys over a public channel.

- The internet is public and filled with many potential eavsdroppers (***Eve's***)
-
- How to establish a *secure* communication channel across an insecure one?

Diffie-Hellmen



Diffie-Hellman w/colors



Diffie-Hellman usage

HTTPS://

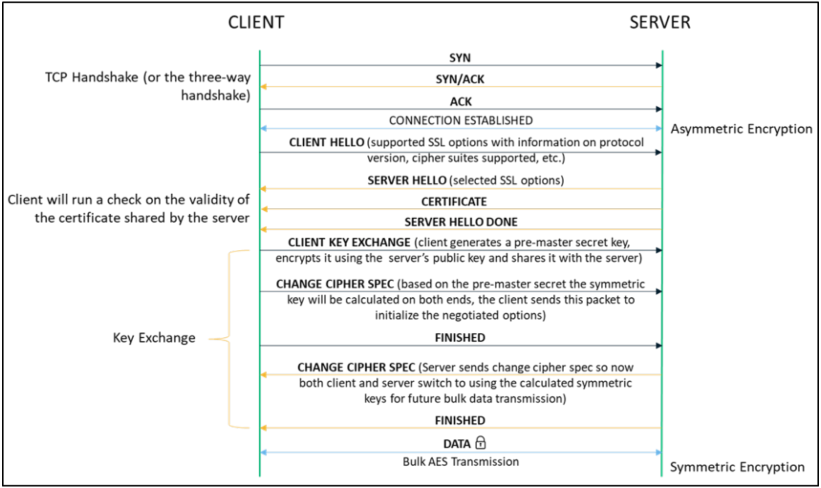

CLIENT
SERVER
TCP Handshake (or the three-way handshake)
SYN
SYN/ACK
ACK
CONNECTION ESTABLISHED
Asymmetric Encryption
CLIENT HELLO (supported SSL options with information on protocol version, cipher suites supported, etc.)
SERVER HELLO (selected SSL options)
Client will run a check on the validity of the certificate shared by the server
CERTIFICATE
SERVER HELLO DONE
CLIENT KEY EXCHANGE (client generates a pre-master secret key, encrypts it using the server's public key and shares it with the server)
CHANGE CIPHER SPEC (based on the pre-master secret the symmetric key will be calculated on both ends, the client sends this packet to initialize the negotiated options)
Key Exchange
FINISHED
CHANGE CIPHER SPEC (Server sends change cipher spec so now both client and server switch to using the calculated symmetric keys for future bulk data transmission)
FINISHED
DATA 🔒
Bulk AES Transmission
Symmetric Encryption

Figure 7: TCP Handshake