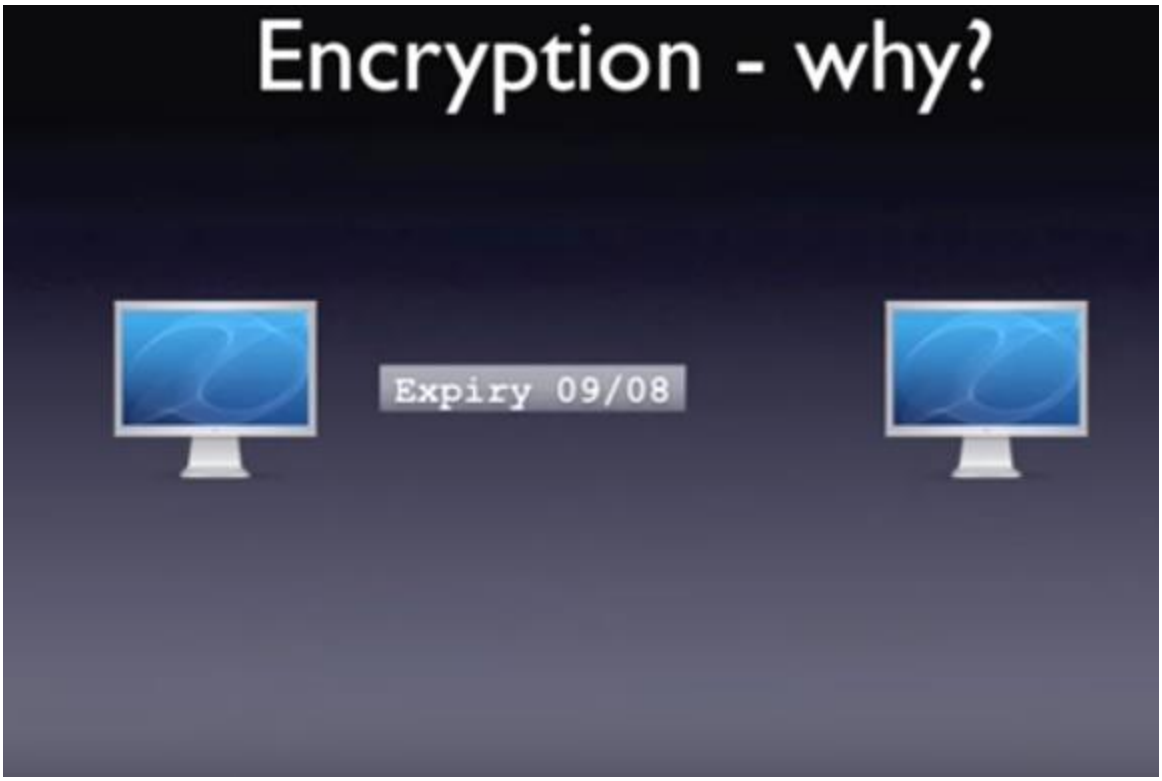
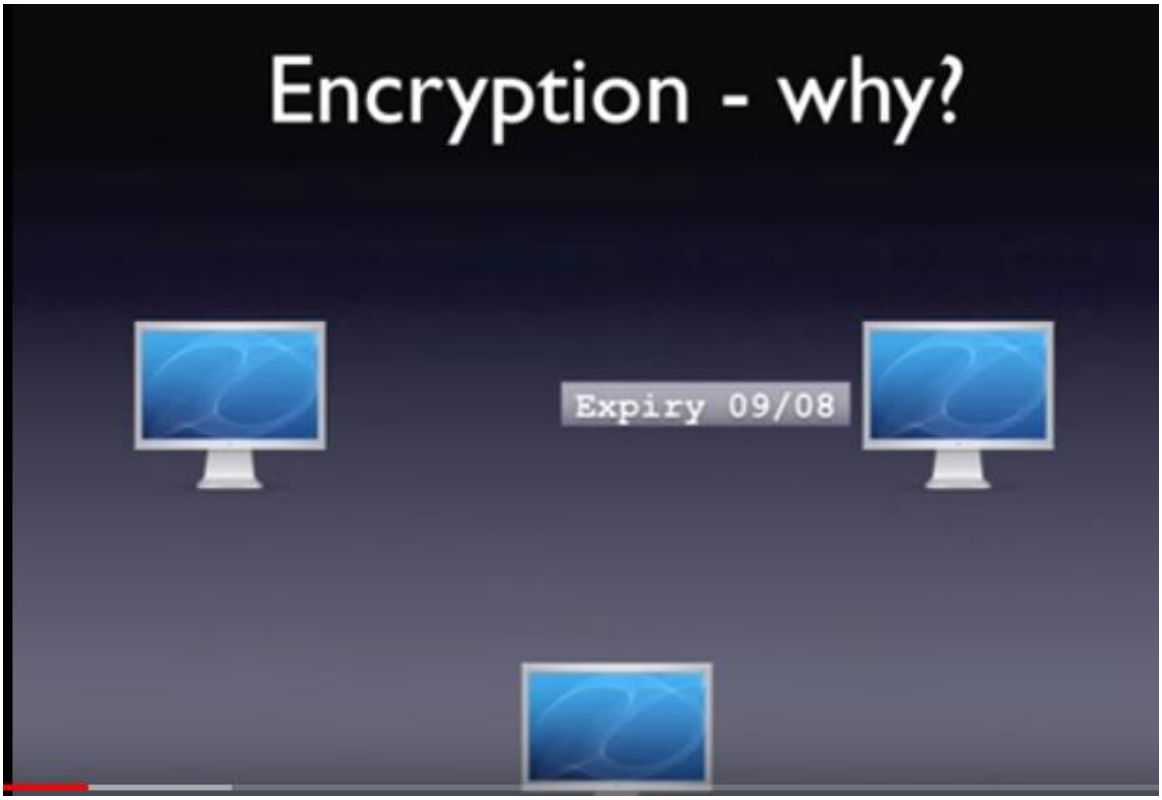
What we will cover

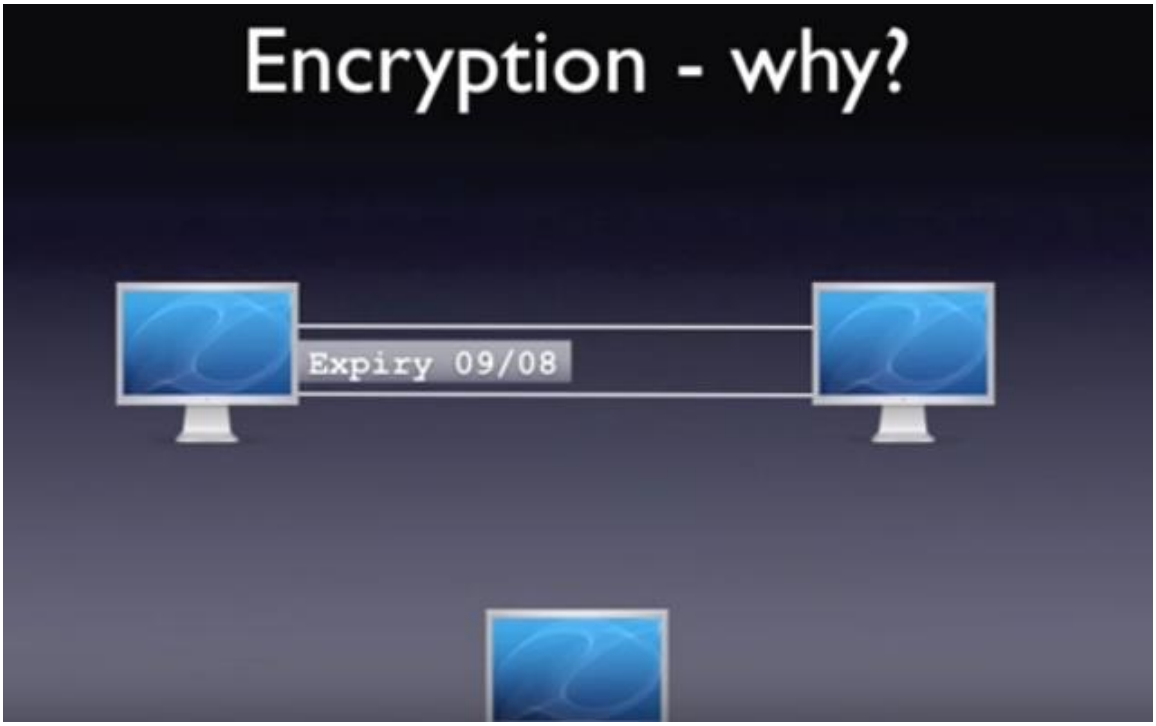
- Why SSL and certificates exist
- Encryption
  - Why encrypt?
  - How encryption works
- Identification
  - How your computer can decide who to trust

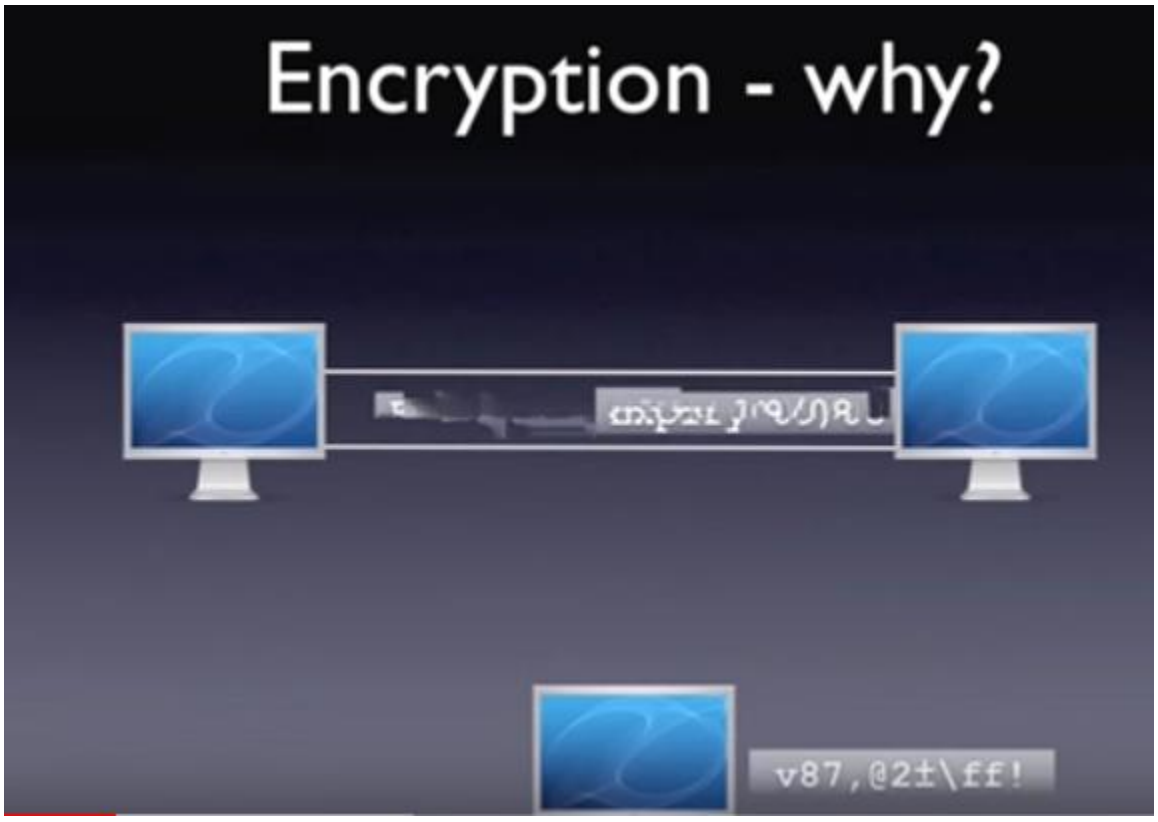
Why SSL exists

Encryption	Identification
Hiding what is sent from one computer to another	Making sure the computer you are speaking to is the one you trust


Encryption - why?
Expiry 09/08


Encryption - why?
Expiry 09/08


Encryption - why?
Expiry 09/08


Encryption - why?
v87,@2±\ff!

Encryption - how?

1. Computers agree on how to encrypt
2. Server sends certificate
3. Your computer says 'start encrypting'
4. The server says 'start encrypting'
5. All messages are now encrypted

Encryption - how?

1. Computers agree on how to encrypt

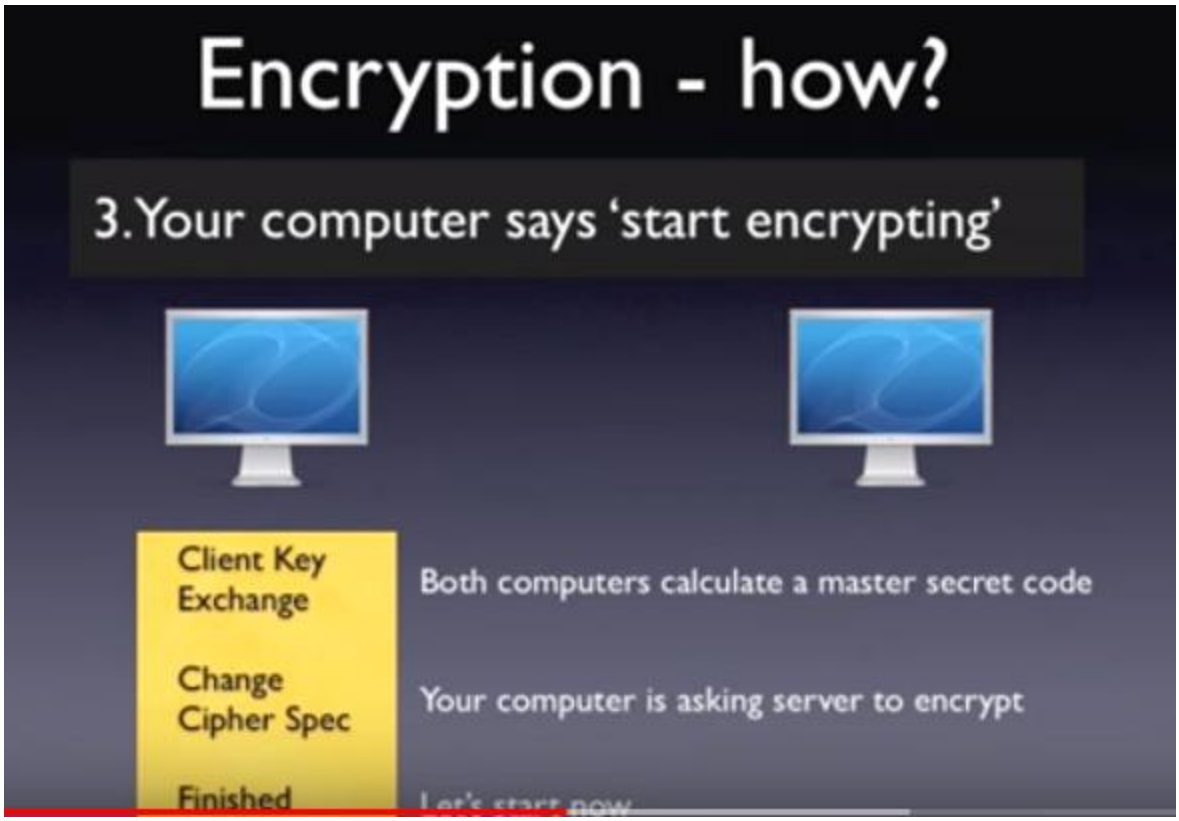
Key	Cipher	Hash
RSA	RC4	HMAC-MD5
Diffie-Hellman	Triple DES	HMAC-SHA
DSA	AES	

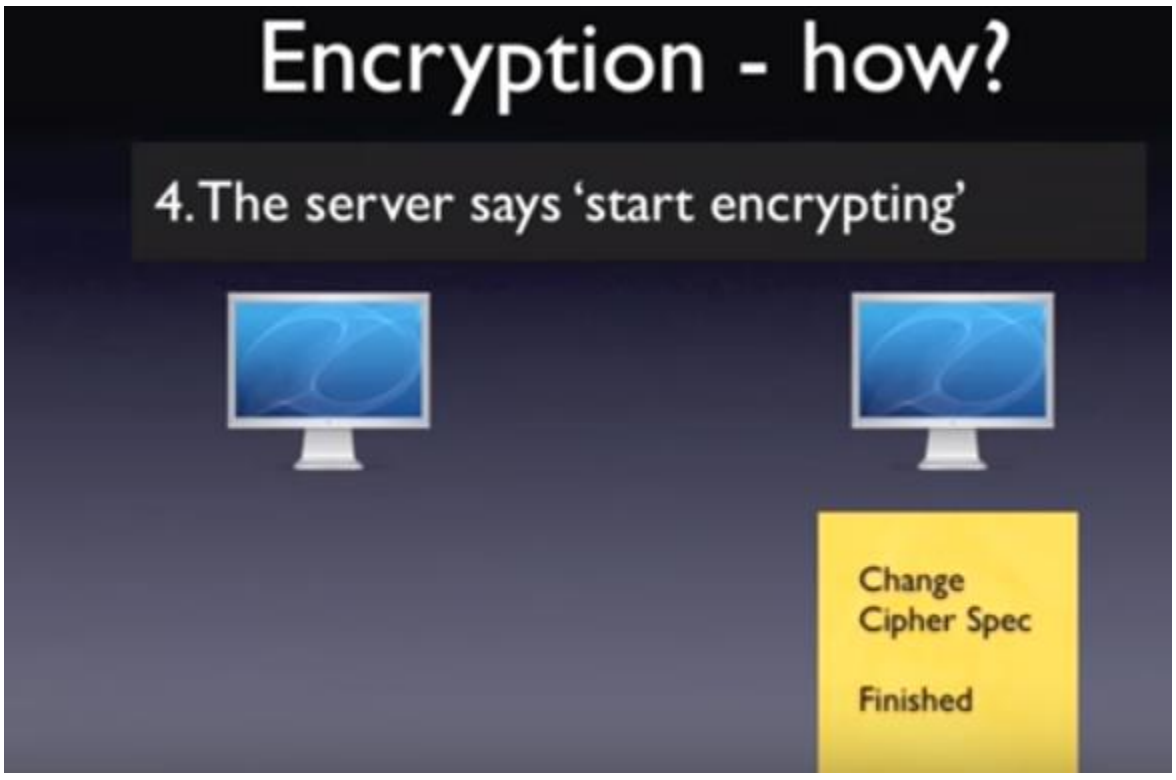
Version 3.3

Key	Cipher	Hash
RSA	RC4	HMAC-MD5
Diffie-Hellman	Triple DES	HMAC-SHA
DSA	AES	

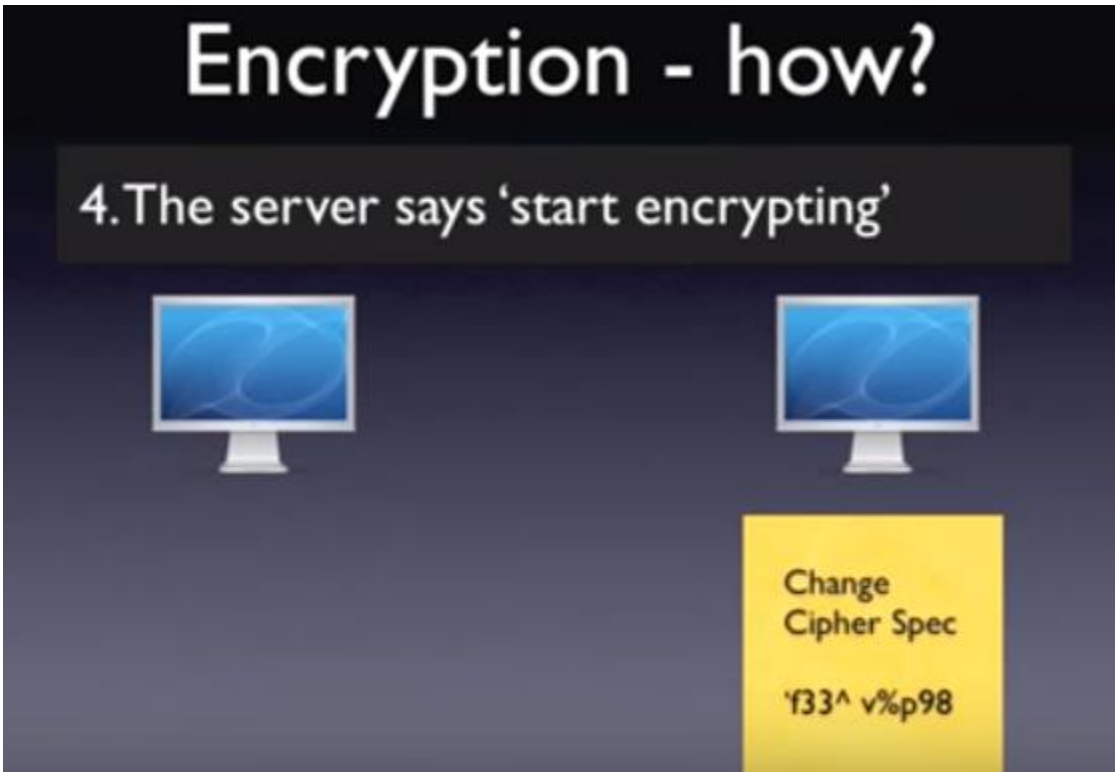

Encryption - how?
2. Server sends a certificate
Serial: 123123
Issuer: Verisign
Valid: from-to
Public key
Subject:
Site
Company


Encryption - how?
3. Your computer says 'start encrypting'
Client Key
Exchange
Change
Cipher Spec
Finished


Encryption - how?
3. Your computer says 'start encrypting'
Client Key Exchange
Change Cipher Spec
Finished
Both computers calculate a master secret code
Your computer is asking server to encrypt

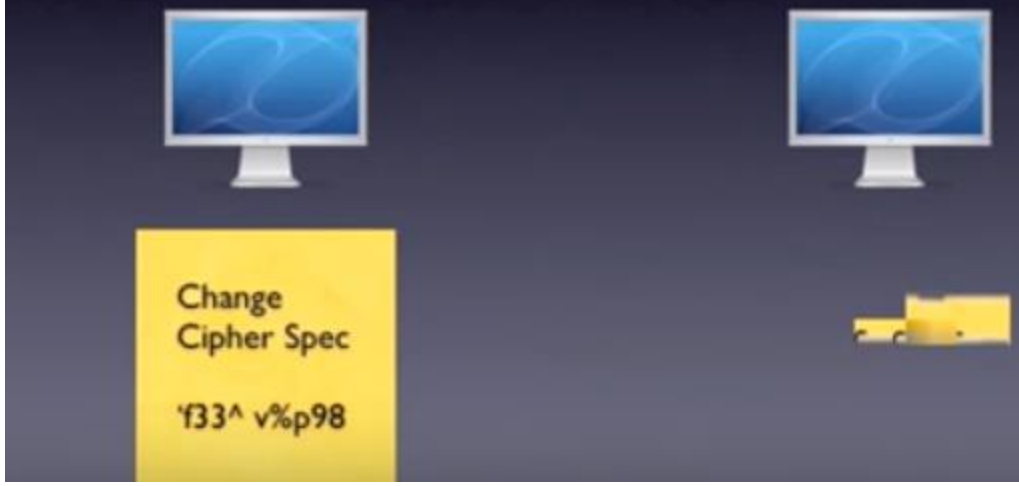

Encryption - how?
4. The server says 'start encrypting'
Change Cipher Spec
Finished


Encryption - how?
4. The server says 'start encrypting'
I'm going to send encrypted messages now
Let's go
Change Cipher Spec
Finished


Encryption - how?
4. The server says 'start encrypting'
Change Cipher Spec
'f33^ v%p98

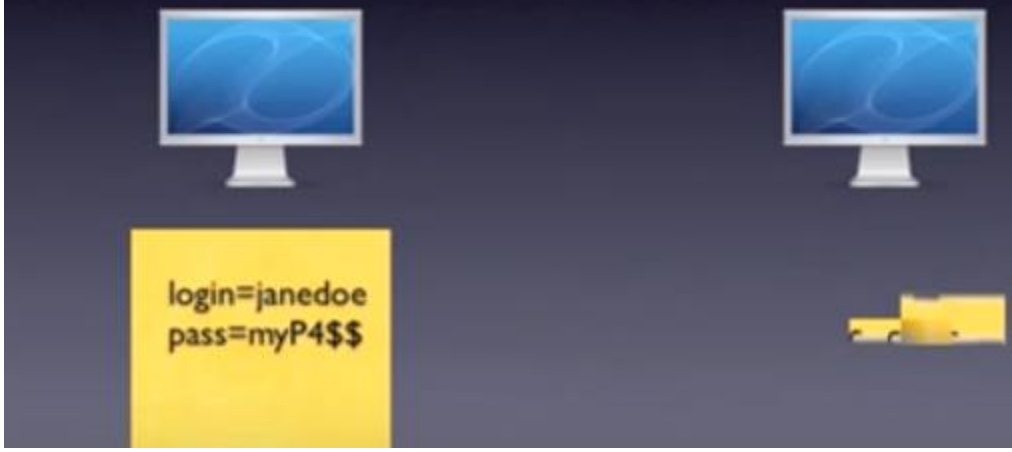
Encryption - how?

4. The server says 'start encrypting'



Encryption - how?

5. All messages are now encrypted




Encryption - how?
5. All messages are now encrypted
login=janedoe
pass=myP4$$


Encryption - how?
5. All messages are now encrypted
hx&@HX373
nwd73*§dh'm
/**yqw

Who to trust

1. Company asks CA for a certificate
2. CA creates certificate and signs it
3. Certificate installed in server
4. Browser issued with root certificates
5. Browser trusts correctly signed certs

Who to trust?

1. Company asks CA for a certificate

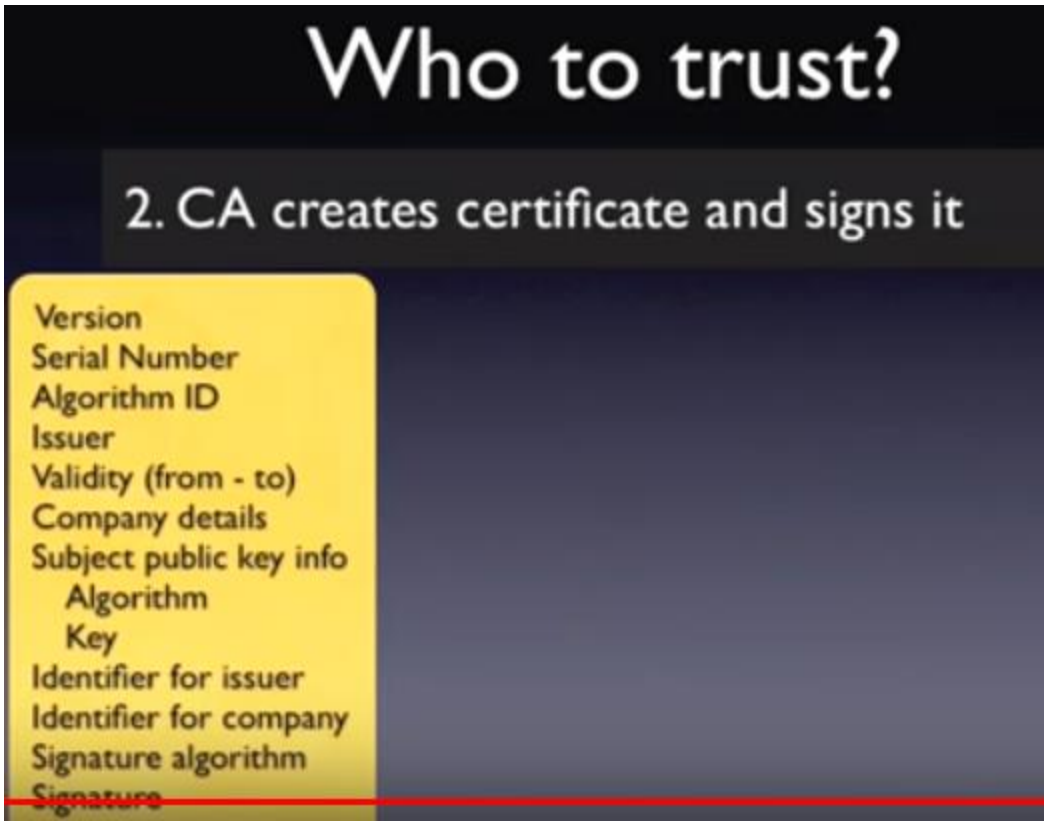
The company has to give information about:

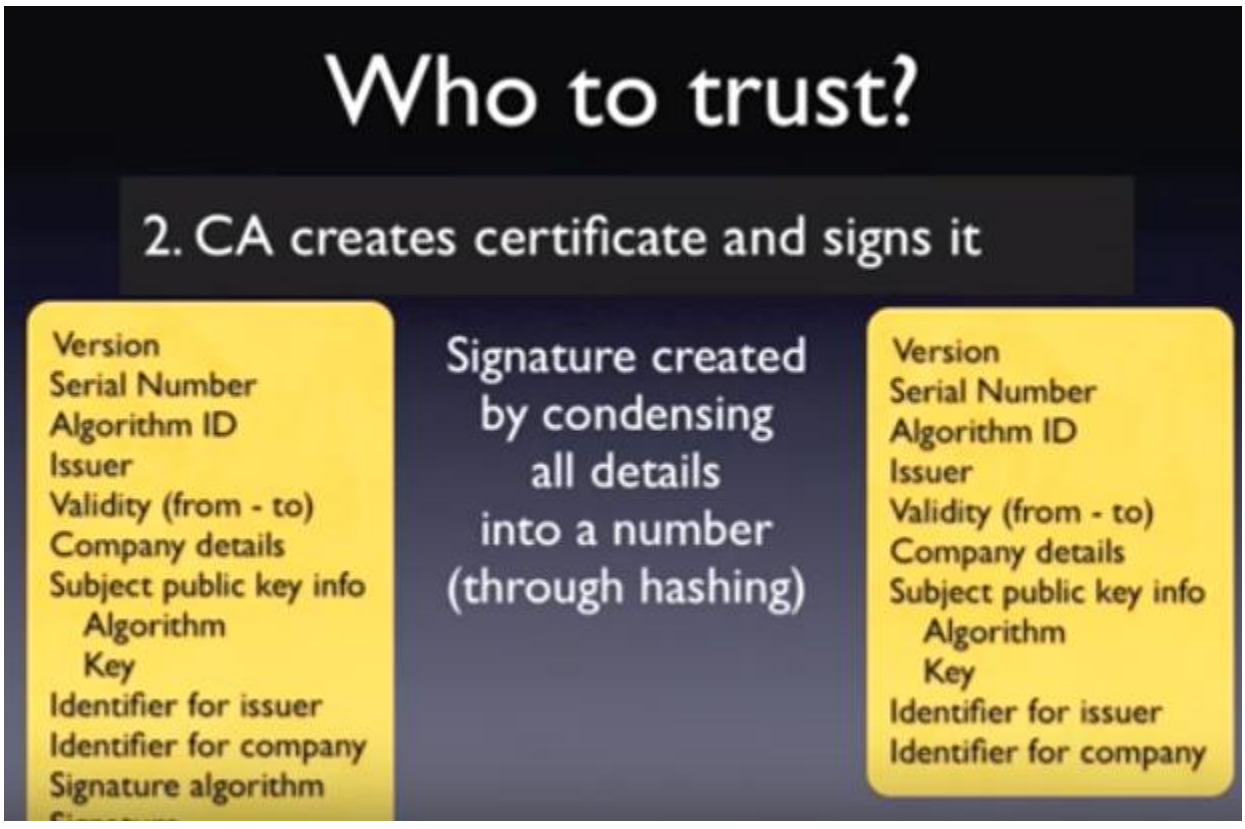
The web server

What the company is

Where it is located

Certificate Authority checks correctness and authenticity of company


Who to trust?
2. CA creates certificate and signs it
Version
Serial Number
Algorithm ID
Issuer
Validity (from - to)
Company details
Subject public key info
Algorithm
Key
Identifier for issuer
Identifier for company
Signature algorithm
Signature


Who to trust?
2. CA creates certificate and signs it
Version
Serial Number
Algorithm ID
Issuer
Validity (from - to)
Company details
Subject public key info
Algorithm
Key
Identifier for issuer
Identifier for company
Signature algorithm
Signature created by condensing all details into a number (through hashing)
Version
Serial Number
Algorithm ID
Issuer
Validity (from - to)
Company details
Subject public key info
Algorithm
Key
Identifier for issuer
Identifier for company

Who to trust?

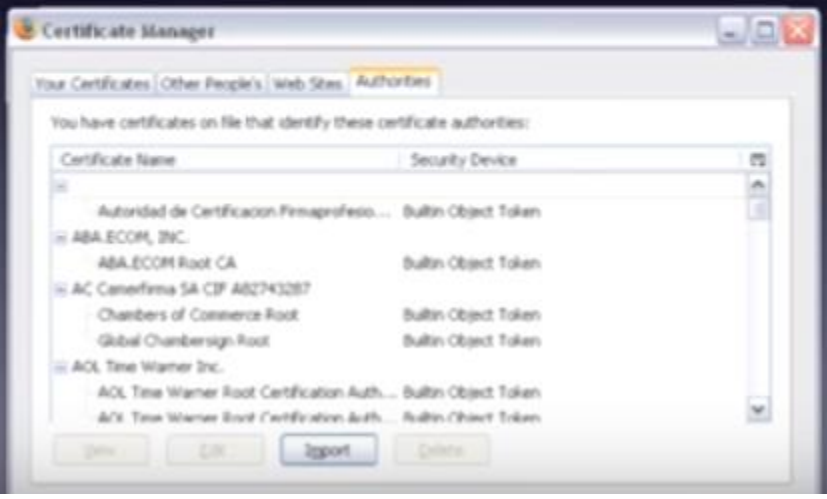
3. Certificate installed in server

The company will run a web server

The certificate is installed into this server

Who to trust?

4. Browser issued with root certificates




Who to trust?
5. Browser trusts correctly signed certs
CA Cert
Verl
Site Cert