Introduction to Modern Cryptography

11th lecture:

Digital Signatures

Public-Key Infrastructures

last time:

- RSA encryption
- CCA security

11th lecture (today):

- Digital Signatures
- Public-Key Infrastructures

	secret key	public key
confidentiality	private-key encryption	public-key encryption
authentication	message authentication codes (MAC)	digital signatures

Certificates & Public-Key Infrastructures (PKI)

- use digital signatures to securely distribute public keys!
- a digital certificate is a signature, binding some entity to some public key
- For instance:

$$cert_{C \to B} = Sign_{sk\_C}(\text{“Bob's key is } pk_B\text{”})$$

- Standard used on the internet: X.509

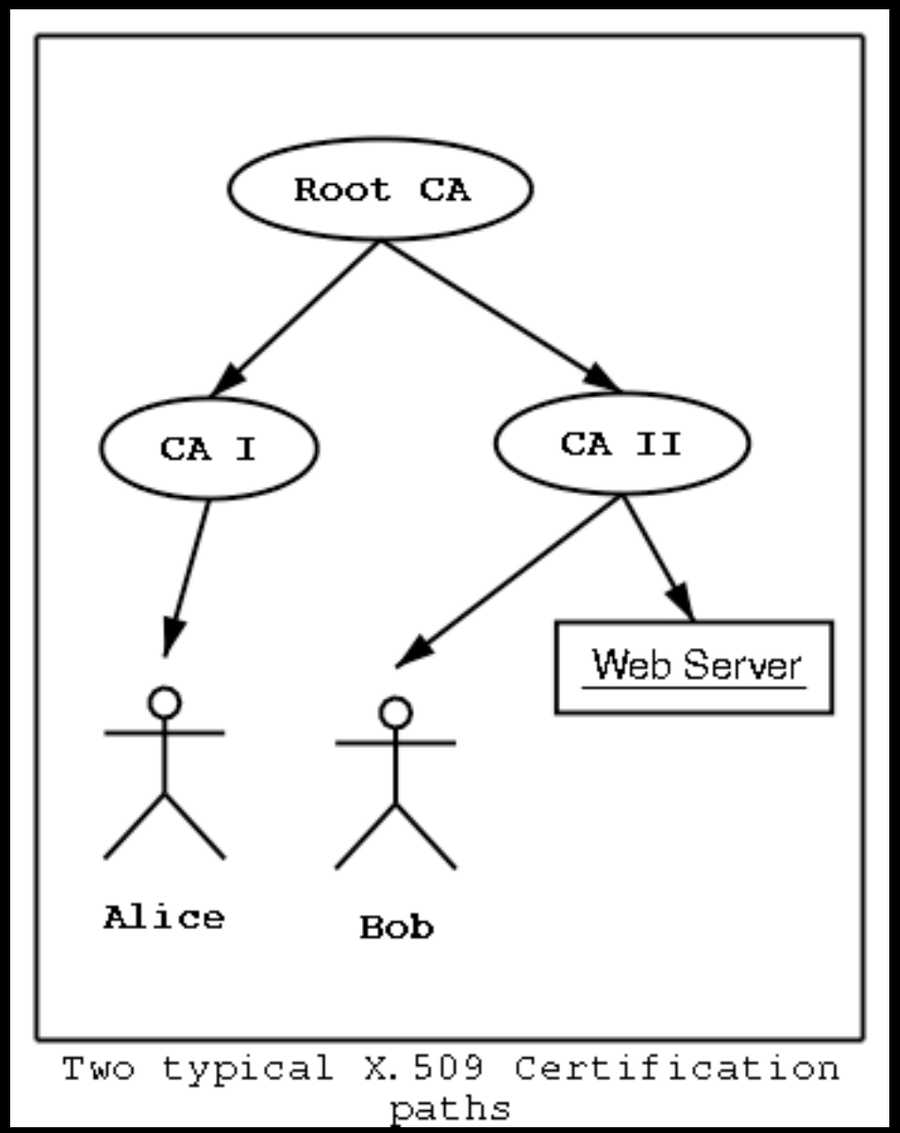
Example of PKI: Certificate Authority (CA)

- completely trusted by everybody
- every user needs to know the CA's public key pk_{CA}
- ship it bundled with software (e.g. in browsers)

- Single point of failure / trust
- Use multiple CAs instead

Certificate Chains and Delegation

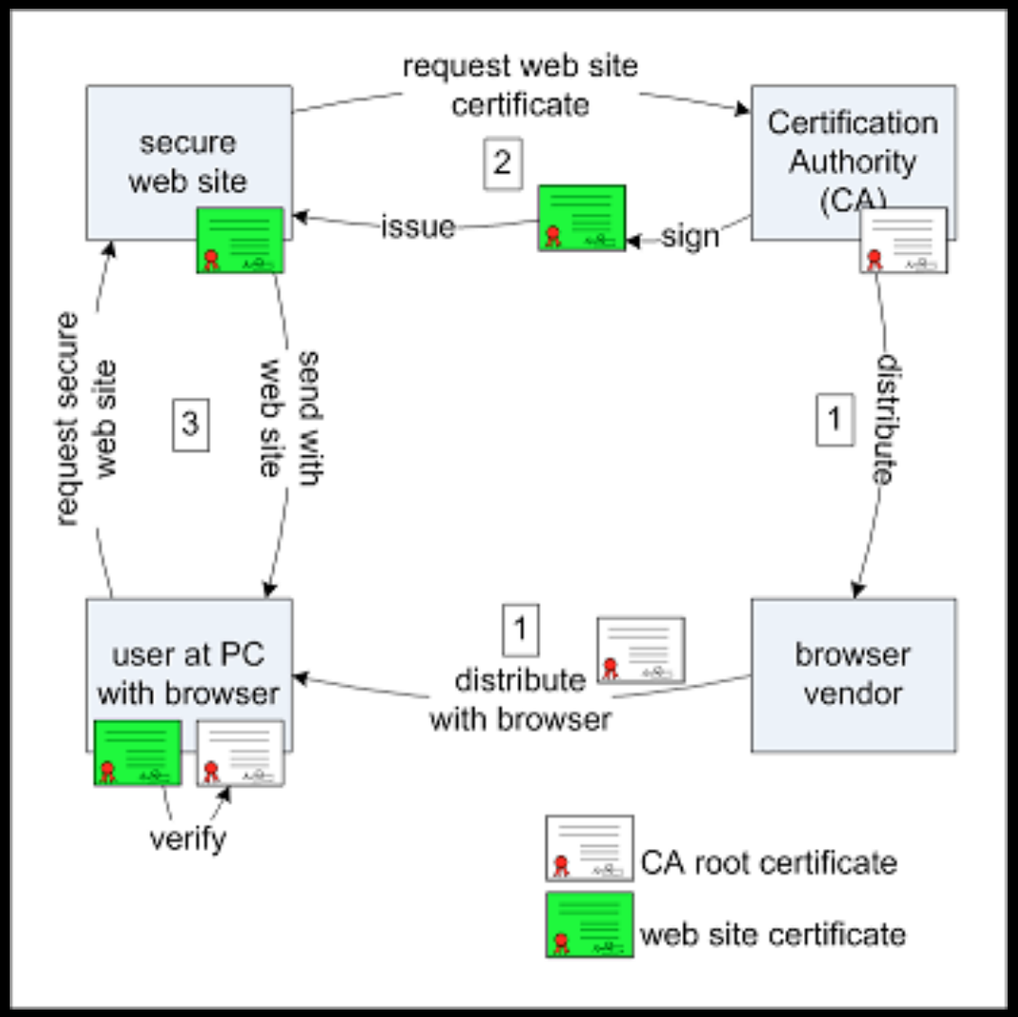
- $cert_{B \to A}$ = Sign $_{sk\_B}$ ("Alice's key is pk_A")
- If Alice wants to communicate to Dave who knows and trusts Charlie, she sends
  pk_A, $cert_{B \to A}$, pk_B , $cert_{C \to B}$
- "stronger trust": Dave learns pk_B and needs to trust Bob to issue other certificates



Two typical X.509 Certification paths

image credit: http://gnutls.org/openpgp.html

Web Certificates



1. A Certification Authority distributes its CA root certificate via browser vendors to browsers. These root certificates reside in a "trust list" on the user's PC.

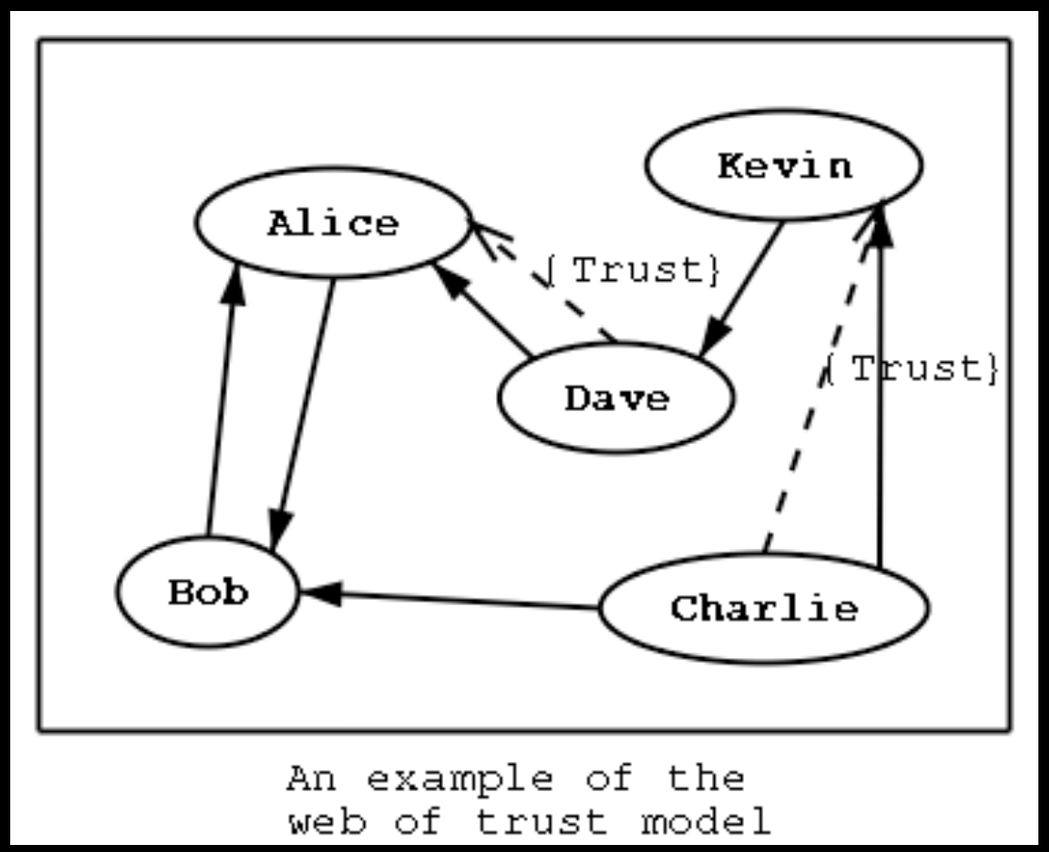
2. A company that wants its website to be secured, purchases a website certificate at the CA. This certificate is signed by the CA and guarantees the identity of the website to the users.

3. When a user wants to visit the secure website, the web browser will first ask the web server for the certificate. If its signature can be verified with the certificate of a CA in the trust list, the website certificate will be accepted. Then the website will be loaded into the browser, and all traffic between the browser and the website will be secured by using encryption.

credit: http://www.win.tue.nl/hashclash/rogue-ca/

Web of Trust (as in OpenPGP)

- every users decides individually whom to trust
- public keys can be signed by different people, e.g. at key-signing parties



An example of the
web of trust model

- Dave implicitly trusts Bob's pk
- Charlie signed Bob's key but does not trust him

image credit: http://gnutls.org/openpgp.html

Invalidating Certificates

- insert expiry dates:
  $cert_{C \to B}$ = Sign $_{sk\_C}$ ("Bob's key is pk_B" , date)
- when date has passed, get a new certificate

- revocation: (include a serial number with all certs)
  $cert_{C \to B}$ = Sign $_{sk\_C}$ ("Bob's key is pk_B" , serial-#)
- CA stores a list of (Bob, pk_B , serial-#)
- If skB is stolen, Bob alerts the CA
- CA creates certificate revocation list (CRL) with all serial-#s of revoked certificates, signs the list with date and publishes it
- verifying the certificate now requires checking if the serial-# has not been revoked