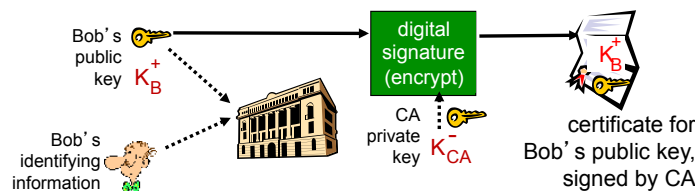


Network Security (III)

Certificates

- Solving the public key distribution problem
- Trust (having somebody's public key) is transitive
 - A trusts C and B trusts C $\Rightarrow$ A can establish trust with B
- Where to start?
 - who to trust?
 - how is the initial trust established?
- Solution: **Certificate Authority (CA)**
- **Goal:** A wants to prove its identity to B
- **Given:** A and B trust CA and both have CA's public key
- **Use:** A presents its certificate when initiating communication with B
- So, what is a certificate and how to use it?
- A certificate (aka public key certificate) is A's public key plus additional information (e.g., A's identification) encrypted with CA's private key.



Certificates - Issuance

- If A gets a certificate from a CA:
 - A generates public/private key pair
 - A generates Certificate Signing Request (CSR) and sends it to CA
 - CA (hopefully) makes sure that it interacts with A and not an impostor
 - CA encrypts the certificate (public key + A's identification + ...) with its own private key
 - The certificate is delivered to A (can be done securely since A has CA's public key and CA is trusted by A)

Certificates - Use

- If B needs to authenticate A
 - B sends a request to A for A's certificate coupled with a nonce
 - A sends back its certificate and the encrypted nonce using A's private key

- B decrypts the certificate with CA's public key (as the certificate is encrypted using CA's private key, and B trusts CA) and verifies that it was issued to A
- B decrypts the encrypted nonce using A's public key and verifies that the nonce matches the original nonce sent to A.
- If the decrypted nonce matches the original value, B can tell A is the real A and can use A's public key to establish a secure communication (e.g., exchange the symmetric key).

Inspect Certificate Information

- We can use a web browser to inspect the information of a certificate of a website running over HTTPS. We can also use `openssl` to inspect the information too.
- `openssl s_client -connect www.google.com:443 | openssl x509 -text -noout`
 - `openssl s_client -connect <hostname>:<port>` test connectivity to a server.
 - `openssl x509` certificate display and signing utility
 - `-text` outputs the certificate's in the text format.
 - `-noout` prevents output the encoded version

Disclaimer: Notes adapted from the textbook and online resources.