Public Key Infrastructure

Panko Chapter 3 p129-131 &137-139
Also these slides (adapted from 2015 slides) and the standards

Panko and Panko
Business Data Networks and Security, 10th Edition, Global Edition

Review

- What is **symmetric** cryptography?
- What are the secret elements?
- What is **asymmetric** cryptography?
- What are the secret elements?

Public Key Infrastructure

- Secret key distribution.
- Public key distribution.
- Public key certificate.
- Public key infrastructure.

Secret Key Distribution

- For secure communications based on symmetric key systems (e.g. AES), the secret keys have to be distributed before any decryption can be done.
- Distribution of secret keys can be very costly.
  - By post? insecure, unreliable, time delay.
  - Use courier? unreliable, time delay, expensive.
  - Use secure underground cable? Too costly, hard to change.
  - Use public channels? Virtually everyone can get access.

Secret Key Distribution

- Management of secret key is a big problem.
  - The same key cannot be used too many times.
  - Keys need to be changed from time to time.
  - Cipher code book (a collection of many secret keys): once used in military but no longer.
- To enable secure communication between any 2 out of n parties, how many secret keys are needed?
- There are n(n-1)/2 different secret keys to be distributed!
  - For this class, that's about 10,000
  - Need to update? all the new keys need to be distributed.

Secret Key Distribution: D-H Key Exchange

- **Common knowledge to** A **and** B, **not secret:** a prime p, and a number g (ideally, a primitive root mod p) so $g < p$.
- **Private knowledge**: random numbers a (to A) and b (to B)
- A ⇒ B: $\alpha = g^a \bmod p$
- B ⇒ A: $\beta = g^b \bmod p$
- A **compute:** $K1 = \beta^a \bmod p$.
- B **compute:** $K2 = \alpha^b \bmod p$.
- **Common key:** Then K1 = K2 is the common key

Diffie-Hellman key exchange protocol 1976

- E pretends to be B and establishes a common key with A.
- A thinks it is B and communicates with E.
- E gets all the (confidential) information that A is supposed to share with B, without B being aware of it.
- Why is the attack successful?
- No authentication, i.e. there is no way for A and B to check each other's identity.

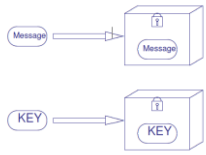
Attack to D-H protocol

RSA encryption and distribution is similar

- But more complex
- Ron Rivest, Adi Shamir, and Leonard Adleman, 1977
- Commercialised with PKI Certificates through the Certificate Authority Verisign 1995 (now Symantec)
- Dominant

Secret Key Distribution

- Use public key system? Yes.
  - Convenient, cheap, and easy to update.
- But whose public key am I supposed to use?! And how do I trust it?



Public Key Distribution

- **Q: But how to distribute public keys?**
  - A sends her public key to B.
  - To how many people does A have to send her public key?
  - How does B trust that it is A's public key?
- An important issue in public key distribution is to make sure the key belongs to the right person.
- **A: Public key certificate**
- A certificate issued by a mutually Trusted Third Party (TTP).
- Third party is trusted by supplicant (key owner) and verifier (who uses the certificate).
- It contains public identification information (not the sensitive information such as credit card number) of the key owner, the **public key**, issuing authority, expiration date, etc.

Public Key Certificate

- **Who issues the public key certificate for you?**
- A public key certificate issuing authority: "CA".
- The authority's public key is publicly available (and trusted).
- The validity of individual's public key can be verified using the authority's public key.
- The individual's public key can be retrieved from the public key certificate.
- You have to trust the authority, if you want to use the public key certificate services.

e.g. Commonwealth government trusted certificate issuing authorities

Policy requirement: the Gatekeeper PKI Framework states Australian Government agencies must only use digital keys and certificates issued by a Gatekeeper-accredited organisation for PKI authentication.

Gatekeeper accredited service providers

The following services have been granted accreditation by the Gatekeeper Competent Authority.

Provider	Service type	Accreditation date
Symantec (eSign)	Certification Authority	31 March 2000
Australia Post (KeyPost)	Registration Authority	December 2001
Department of Defence	Certification and Registration Authority	17 May 2007
Department of Industry and Science	Validation Authority	6 January 2011
Medicare Australia	Certification Authority	29 June 2011
Verizon Australia	Certification Authority	16 February 2012
Australian Taxation Office	Certification Authority	30 April 2013
Property Exchange Australia Limited	Certification Authority	1 October 2014

From https://www.dta.gov.au/standard-design-guides/authentication-frameworks/gatekeeper-public-key-infrastructure-framework

Public Key Certificate

How to use a public key certificate?

- Whoever wants to use a public key has to get the corresponding public key certificate from a directory, check it's validity and extract the relevant information.
- Public key certificates are updated periodically or on request by the owner.
- Always check the public key certificate before using the public key.

Public Key Infrastructure: Roles

To enable public key systems to work properly, a single TTP to manage the public key certificates is not enough. **How many TTPs?**

- **CA:** Certificate authority (may be more than one), who issues public key certificates.
- **RA:** Registration authority, who authenticates the entity making the request on behalf of the CA and requests certificate. May use physical credentials.

Store the public key certificates: International standard is ITU-T/ ISO authentication framework, known as X.500 protocols.

- **X.500** Hierarchical directories, to hold the certificates- largely outdated by LDAP (as a directory) and Browser+SSL (for public key certificates)
- **X.509** Protocol to manage the certificates (how to put, get, update, revoke, plus certificate format) remains critical for Web security

Public Key Infrastructure: Certification phase

- **Initiate:** Requester (subject) applies for a public key certificate from CA.
  - Requester presents his/her public key to be certified.
  - Requester shows his/her ID (e.g. off-line).
  - RA checks the validity of the requester, and requests public key certificate from CA.
  - CA generates a public key certificate.
  - RA (probably) puts it in a directory to be discovered.

Public Key Infrastructure: Certification phase

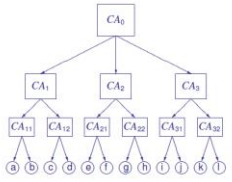
- **Update:** Requester (subject) applies for his/her public key certificate to be updated.
  - Requester sends his/her new public key to be certified.
  - Requester signs request message, new public key, ID information (old public key serial number, etc.), using his/her old private key.
  - CA verifies the request information using requester's old public key.
  - CA generates a new public key certificate.
  - RA checks the validity of the new certificate, and (probably) updates the old one from the directory.

Public Key Infrastructure: Verification phase

- What happens if Alice and Bob do not have the same CA?
- User Bob wants to get Alice's public key.
  - Bob finds the closest directory to Alice and gets her public key certificate.
  - Bob checks CA's ID (information is associated with Alice's public key certificate).
  - Bob gets and verifies CA's public key certificate (using another CA's public key).
  - Bob verifies the validity of Alice's public key certificate.
  - Bob extracts Alice's public key from the certificate.

Public Key Infrastructure

Hierarchical structure of certificates and issuing authorities
□: Certificate authority.
○: End user.



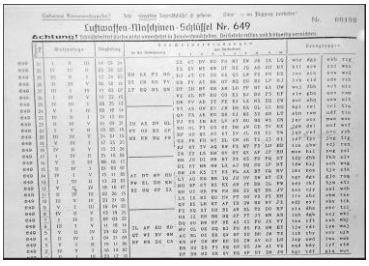
Public Key Infrastructure

- Each certification authority has issued a public key certificate for all its associated CAs.
- When a CA changes its public key, all of its associated CAs have to re-issue a public key certificate for the CA.
- If user **A** want to get user **D**'s public key, she has to

1. get $\mathbf{CA}_1$'s public key certificate issued by $\mathbf{CA}_{11}$; then
2. get $\mathbf{CA}_{12}$'s public key certificate issued by $\mathbf{CA}_1$; then
3. get **D**'s public key certificate issued by $\mathbf{CA}_{12}$; then
4. retrieve and verify **D**'s public key.

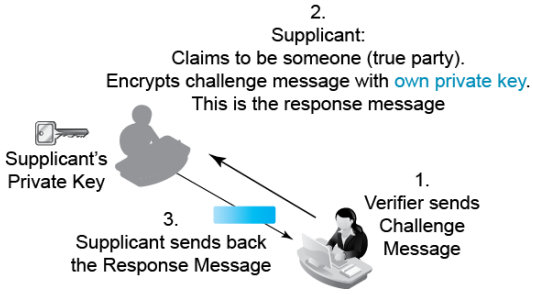
- This process is written as
  - $CA_{11} << CA_1 >> CA_1 << CA_{12} >> CA_{12} << D >>$
- The procedure that **D** can get **A**'s public key is the reverse.
- How can **E** get **A**'s public key (from the above diagram)?

Main Point Summary

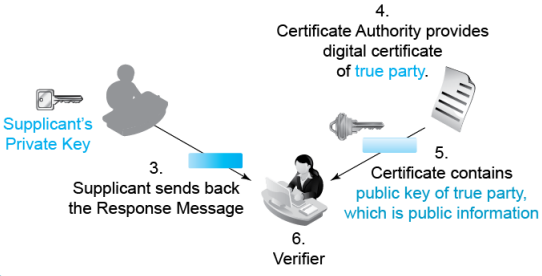


World War II list of keys for the German Enigma cipher machine From https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

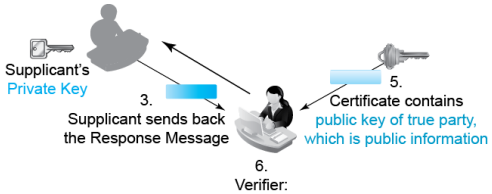
3.18 Digital Certificate Authentication



3.18 Digital Certificate Authentication



3.18 Digital Certificate Authentication



3.18 Digital Certificate Authentication

True Party	Has a Public and Private Key
Supplicant	Encrypts challenge message with supplicant's private key
Verifier	Decrypts response message with true party's public key
Choice	If decryption works, supplicant knows the true party's private key so must be the true party

About Pretty Good Privacy (PGP)

- **What is PGP?** A free software package for email security. We will use the GPG software for PGP.
- PGP has its own means of key management. It does not use X.509 certificates. It does not rely on a hierarchy of trusted authorities—instead it relies on a "web of trust".
- PGP uses public key ring, a database on the end user's server. If the public key cannot be retrieved from the public key ring, the public key is not available.
- PGP uses private key ring, a database on the user's local server.
- Retrieval of private key needs a pass phrase (like a password). The private key is encrypted using the pass phase and stored on the computer.
- It uses public key as well as secret key systems. Public key algorithms are used for session key encryption, while session key is used for real message (body of email) encryption using symmetric key encryption algorithm (triple-DES, or IDEA).
- User can choose encryption only, signature only, or neither or both.