Distributed Systems

COMP90015 2018 Semester 1
Tutorial 10

Things to cover today

Part 1: Security Questions

Part 2: Code Demonstration -- encrypted client and server communication

Security Questions

1. List and briefly explain some worst case assumptions when designing a secure system.
2. Define encryption and describe the two main types of keys used by encryption algorithms.
3. Discuss the three major roles that encryption plays in the implementation of secure systems.
4. Explain how digital signatures work.

Security Questions

5. How does Alice send a secret message to Bob if they both share a secret key?
6. How can Alice authenticate and communicate secretly with Bob assuming there is an authentication server that knows Alice’s and Bob’s secret keys?
7. Assuming that Bob has a public/private key pair, how can Alice and Bob establish a shared key to communicate secretly using a Key Distribution Service?

1. List and briefly explain some worst case assumptions when designing a secure system.

- **Networks are insecure.**
  - Messages can be looked at, copied, modified and retransmitted.
  - Attackers can obtain information that they should not and can pretend to be a legitimate party.
- **The source code is known to the attacker.**
  - Knowing the source code can help the attacker discover vulnerabilities.
- **Interfaces are exposed**
  - Communication interfaces are necessarily open to allow clients to access them.
  - Attackers can send messages to any interface.
- **The attacker has unlimited computing resources.**
  - Assume that attackers will have access to the largest and most powerful computers projected in the lifetime of a system.

2. Define encryption and describe the two main types of keys used by encryption algorithms.

- ***Encryption***
  - process of encoding a message in such a way as to hide its contents.
- ***Shared secret keys*** (symmetric)
  - Sender and recipient share knowledge of the key and it must not be revealed to anyone else.
- ***Public/private key pairs*** (asymmetric)
  - The sender uses a public key to encrypt the message.
  - The recipient uses a corresponding private key to decrypt the message.
  - Only the recipient can decrypt the message, because they have the private key.
  - Typically require 100 to 1000 times as much processing power as secret-key algorithms.

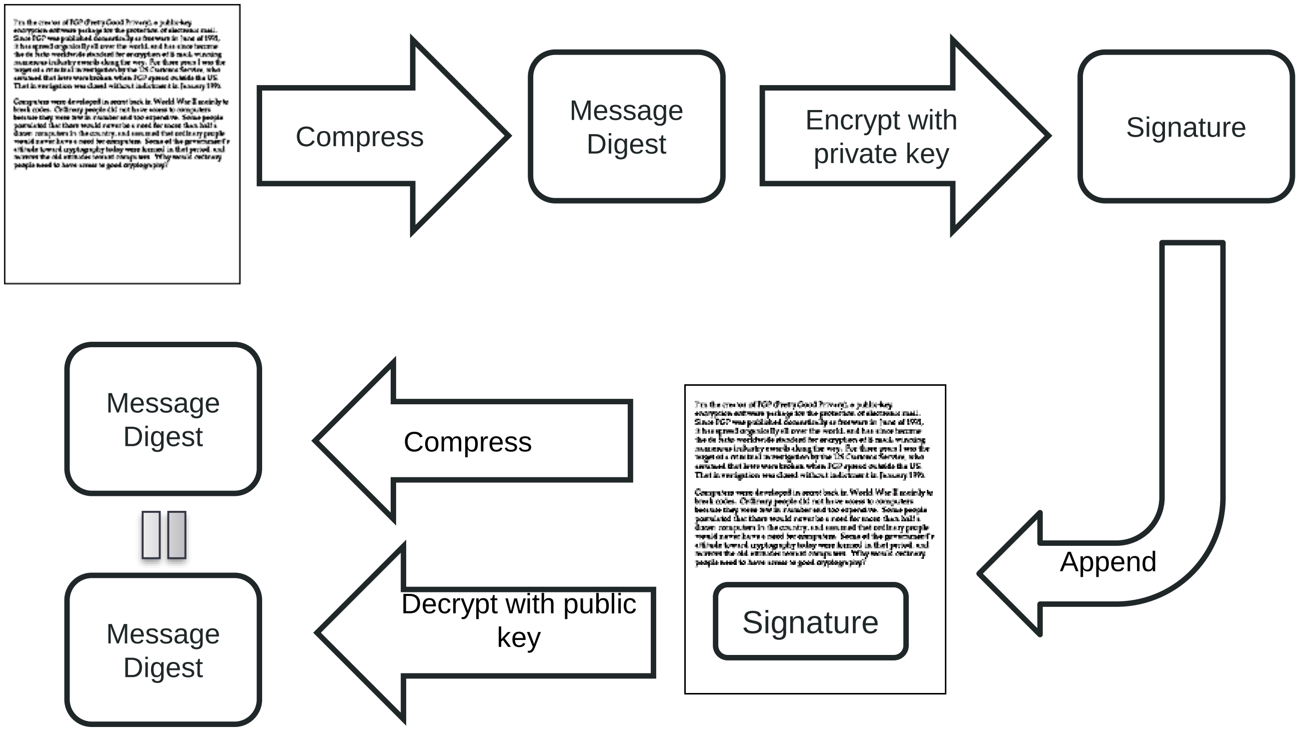
3. Discuss the three major roles that encryption plays in the implementation of secure systems.

- **Secrecy and integrity**
    - Messages encrypted with a particular key can only be decrypted by a recipient who knows the corresponding decryption key => Secrecy.
    - Integrity can be maintained if some redundant information such as a checksum is included and checked in the encrypted message.
- **Authentication**
    - If keys are held in private, a successful decryption authenticates the decrypted message as coming from a particular sender.
- **Digital signatures**
    - Verify to a third party that a message or a document is an unaltered copy of one produced by the signer.
    - It is a "stamp" Bob places on the data which is unique to Bob, and is very difficult to forge.
    - It also assures that any changes made to the data that has been signed cannot go undetected.

4. Explain how digital signatures work.

1. To sign a document, Bob first ‘compresses’ the message into just a few lines. This is called a *digest*.
2. Bob then encrypts the message digest with his private key. The result is the digital signature.
3. Bob appends the digital signature to the document. All of the data that was ‘compressed’ into the digest has been signed.
4. Bob sends the document to Alice.
5. Alice decrypts the signature (using Bob's public key) changing it back into a message digest.
6. Alice ‘compresses’ the document data into a message digest. If the message digest is the same as the message digest created when the signature was decrypted, then Alice knows that the signed data has not been changed and that Bob signed the document (because only Bob has his private key)

4. Explain how digital signatures work.



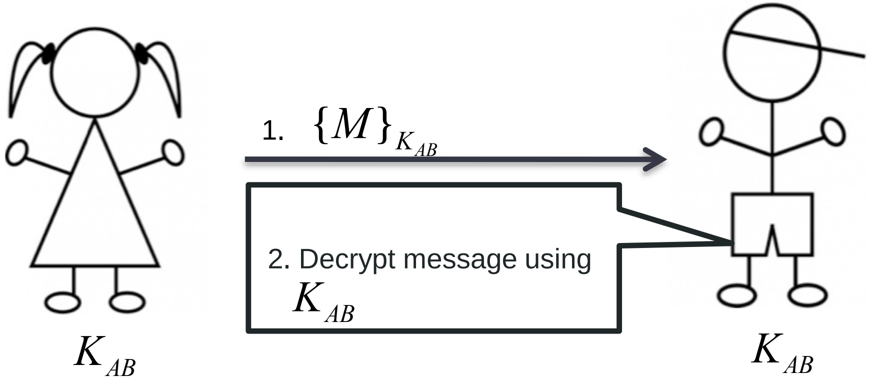
I'm the creator of PGP (Pretty Good Privacy), a public-key encryption software package for the protection of electronic mail. Since PGP was published domestically as freeware in June of 1991, it has spread organically all over the world, and has since become the de facto worldwide standard for encryption of E-mail, winning numerous industry awards along the way. For three years I was the target of a criminal investigation by the US Customs Service, who assumed that laws were broken when PGP spread outside the US. That investigation was closed without indictment in January 1996.

Computers were developed in secret back in World War II mainly to break codes. Ordinary people did not have access to computers because they were few in number and too expensive. Some people postulated that there would never be a need for more than half a dozen computers in the country, and assumed that ordinary people would never have a need for computers. Some of the government's attitude toward cryptography today were formed in that period, and mirrors the old attitudes toward computers. Why would ordinary people need to have access to good cryptography?

I'm the creator of PGP (Pretty Good Privacy), a public-key encryption software package for the protection of electronic mail. Since PGP was published domestically as freeware in June of 1991, it has spread organically all over the world, and has since become the de facto worldwide standard for encryption of E-mail, winning numerous industry awards along the way. For three years I was the target of a criminal investigation by the US Customs Service, who assumed that laws were broken when PGP spread outside the US. That investigation was closed without indictment in January 1996.

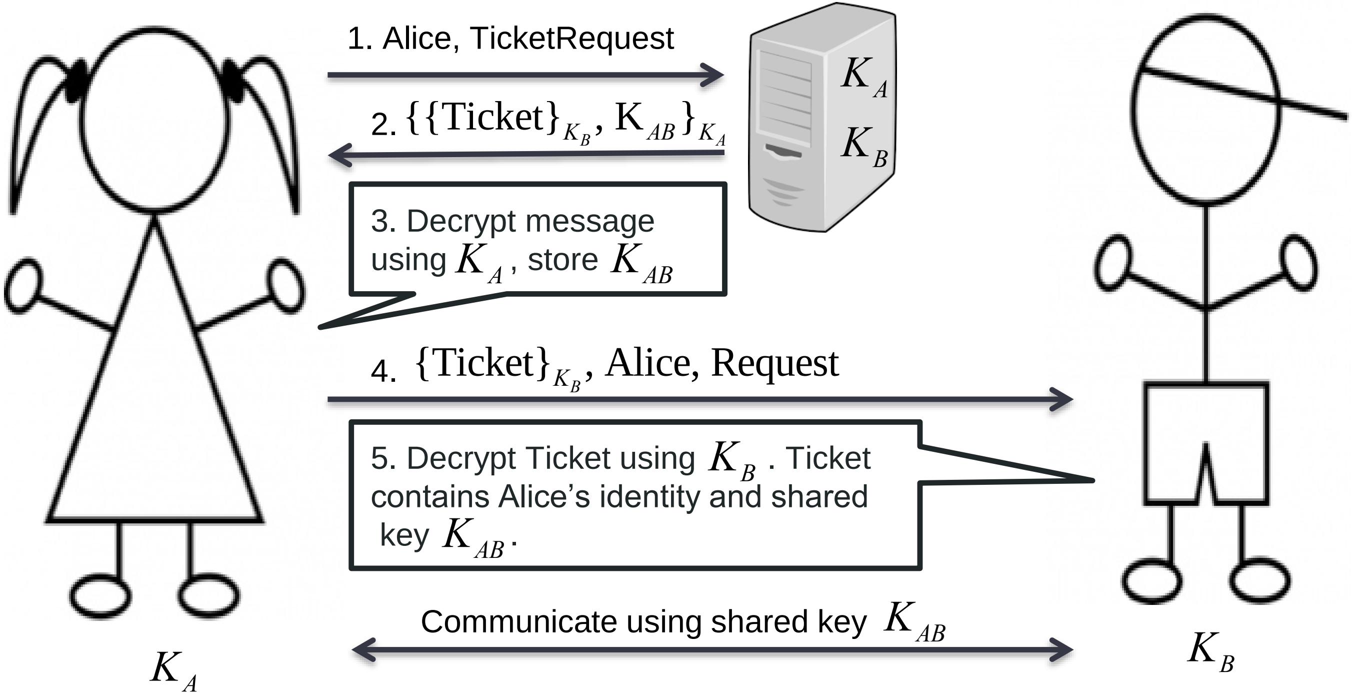
Computers were developed in secret back in World War II mainly to break codes. Ordinary people did not have access to computers because they were few in number and too expensive. Some people postulated that there would never be a need for more than half a dozen computers in the country, and assumed that ordinary people would never have a need for computers. Some of the government's attitude toward cryptography today were formed in that period, and mirrors the old attitudes toward computers. Why would ordinary people need to have access to good cryptography?

5. How does Alice send a secret message to Bob if they both share a secret key?



1. Alice uses K_AB and an agreed encryption function *E*(*K_AB, M*) to encrypt and send the message to Bob.
2. Bob decrypts the encrypted message using the corresponding decryption function *D*(*K_AB*, *M*).

6. How can Alice authenticate and communicate secretly with Bob assuming there is an authentication server that knows Alice’s and Bob’s secret keys?



1. Alice, TicketRequest

2. $\{\{\text{Ticket}\}_{K_B}, \text{K}_{AB}\}_{K_A}$

3. Decrypt message using K_A, store K_{AB}

4. $\{\text{Ticket}\}_{K_B}$, Alice, Request

5. Decrypt Ticket using K_B . Ticket contains Alice’s identity and shared key K_{AB}.

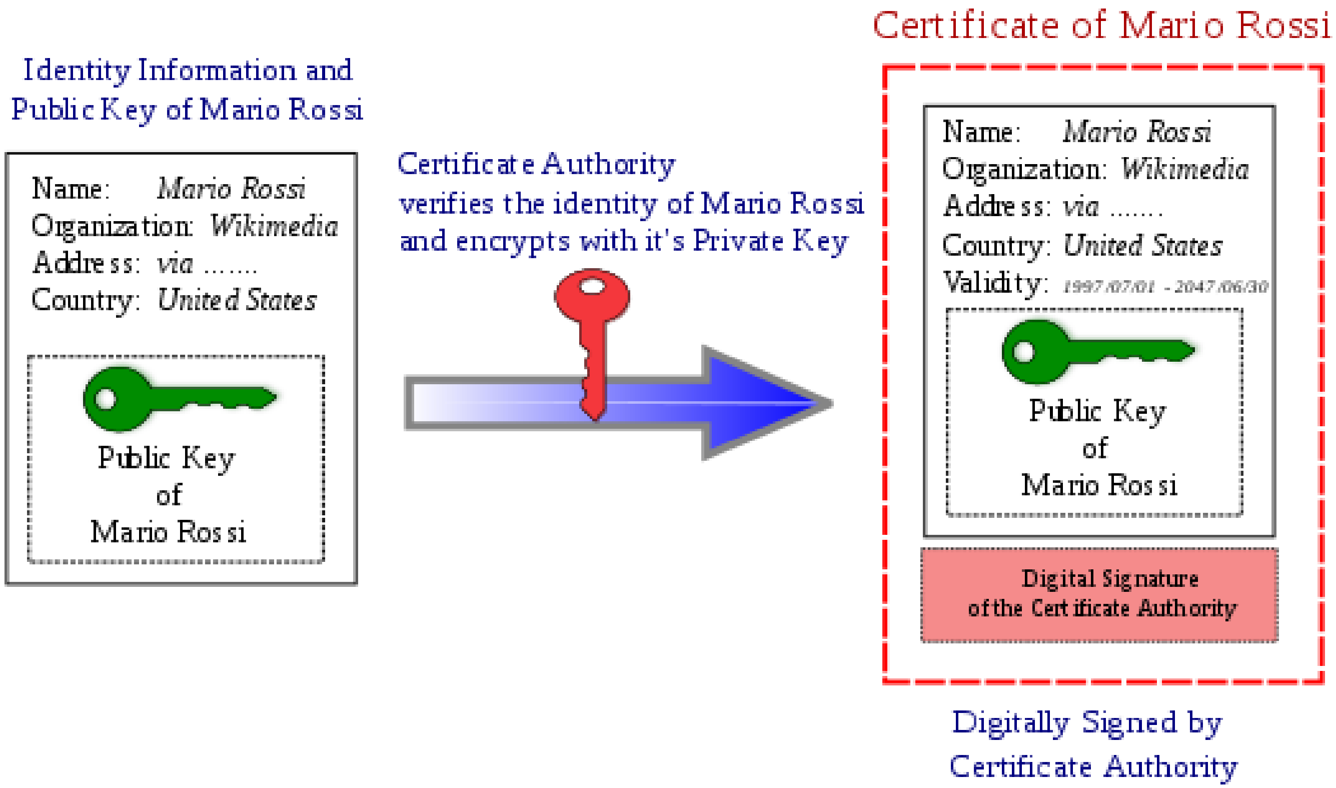
Communicate using shared key K_{AB}

Certificates

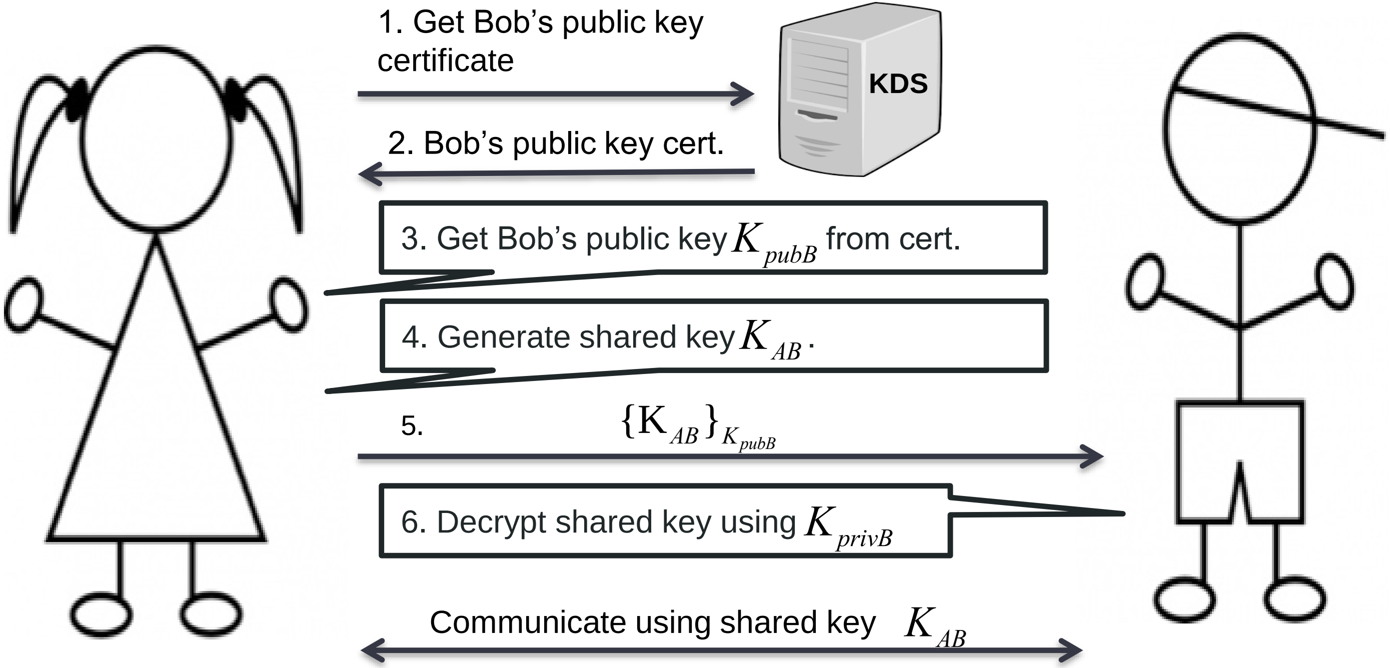
Certificate type: Public key
Name: Bob
Public key: kBpub
Certifying authority: Sara
Signature: {Digest(field 2+field 3)}_{kSpriv}

- In your own words, what is this certificate saying?
  - Sara certifies that Bob’s public key is kBpub
- Why can’t Sara deny that she has attested to this fact?
  - Because if someone can decrypt the signature using kSpub, only someone who had kSpriv could have encrypted it.
- What must be known to anyone who wants to make sure the certificate is authentic?
  - kSpub

Public Key Certificate



7. Assuming that Bob has a public/private key pair, how can Alice and Bob establish a shared key to communicate secretly using a Key Distribution Service?



1. Get Bob's public key certificate

2. Bob's public key cert.

3. Get Bob's public key K_{pubB} from cert.

4. Generate shared key K_{AB}.

5. $\{K_{AB}\}_{K_{pubB}}$

6. Decrypt shared key using K_{privB}

Communicate using shared key K_{AB}

Code Demonstration