# Department of Electronics

## Cryptography

Fall 2019

Hasan Mahmood

hasan@qau.edu.pk

Week 10 (5, 6 November 2019)

- Suppose Eve discovers the value of a
- Alice signature can be produced on any document

# Security Protocols

- Basic cryptographic tools: encryption, hash and digital signature
- How to make computer communications secure

- Public key algorithms
  - Parties who have never met to exchange messages securely
  - Authenticate the origin of t message
  - Hash functions, signature operations can be made efficient

- Still, there are many problems…..

# Problems with Public Keys

- How public keys are distributed?
- People cannot believe public keys
- Imposters can distribute public keys
- Website fake or real, transactions, false organizations
- Authentication issues. How do you confirm identity
- Confirmation required!
- Security protocols are required by combining different cryptographic tools to prevent clever attacks

# Intruder-in-the-Middle and Impostors

- You receive email that asks account information. Legitimacy?
- Imposter can setup a webpage that looks similar to the original
- Certificates and trusted authority (discussed later in detail)
- Public channels, results in intruder-in-the-middle attach
- Real world implementations and applications

# Intruder-in-the-Middle Attack

- Playing game of chess simultaneously, claiming either win one game or draw both
- Intruder-in-the-Middle attack can be used against many cryptographic protocols
- How do we eliminate Intruder-in-the-Middle attack is the challenge
- One example, how to attack Diffie-Hellman scheme

1. Either Alice or Bob selects a large, secure prime number $p$ and a primitive root $\alpha \pmod p$. Both $p$ and $\alpha$ can be made public.

2. Alice chooses a secret random $x$ with $1 \le x \le p - 2$, and Bob selects a secret random $y$ with $1 \le y \le p - 2$.

# Attack on Diffie-Hellman scheme

3. Alice sends $\alpha^x \pmod{p}$ to Bob, and Bob sends $\alpha^y \pmod{p}$ to Alice.
4. Using the messages that they each have received, they can each calculate the session key $K$. Alice calculates $K$ by $K \equiv (\alpha^y)^x \pmod{p}$, and Bob calculates $K$ by $K \equiv (\alpha^x)^y \pmod{p}$.

# Diffie-Hellman scheme, i-i-t-m

1. Eve chooses an exponent $z$.
2. Eve intercepts $\alpha^x$ and $\alpha^y$.
3. Eve sends $\alpha^z$ to Alice and to Bob (Alice believes she is receiving $\alpha^y$ and Bob believes he is receiving $\alpha^x$).
4. Eve computes $K_{AE} \equiv (\alpha^x)^z \pmod{p}$ and $K_{EB} \equiv (\alpha^y)^z \pmod{p}$. Alice, not realizing that Eve is in the middle, also computes $K_{AE}$, and Bob computes $K_{EB}$.
5. When Alice sends a message to Bob, encrypted with $K_{AE}$, Eve intercepts it, deciphers it, encrypts it with $K_{EB}$, and sends it to Bob. Bob decrypts with $K_{EB}$ and obtains the message. Bob has no reason to believe the communication was insecure. Meanwhile, Eve is reading the juicy gossip that she has obtained.

# How to avoid intruder-in-the-middle attack

- Procedure must be in place to authenticate Alice’s and Bob’s identity
- Authenticated key agreement protocol
- Standard way to stop the attacker: station-to-station (STS) protocol
- Uses digital signatures
- Each user U has a digital signature function $sig_U$
- Verification algorithm is $ver_U$
- For example, $sig_U$ could produce RSA or ElGamal signature
- $ver_U$ checks that it is a valid signature for $U$
- Trent, the trusted authority who certifies that $ver_U$ is actually the verification algorithm of $U$ and not Eve

# Diffie-Hellman Key Exchange Revised, I

1. They choose a large prime $p$ and a primitive root $\alpha$.
2. Alice chooses a random $x$ and Bob chooses a random $y$.
3. Alice computes $\alpha^x \pmod{p}$, and Bob computes $\alpha^y \pmod{p}$.
4. Alice sends $\alpha^x$ to Bob.

# Diffie-Hellman Key Exchange Revised, II

5. Bob computes $K \equiv (\alpha^x)^y \pmod{p}$.
6. Bob sends $\alpha^y$ and $E_K(sig_B(\alpha^y, \alpha^x))$ to Alice.
7. Alice computes $K \equiv (\alpha^y)^x \pmod{p}$.
8. Alice decrypts $E_K(sig_B(\alpha^y, \alpha^x))$ to obtain $sig_B(\alpha^y, \alpha^x)$.

# Diffie-Hellman Key Exchange Revised, III

9. Alice asks Trent to verify that $ver_B$ is Bob's verification algorithm.
10. Alice uses $ver_B$ to verify Bob's signature.
11. Alice sends $E_K(sig_A(\alpha^x, \alpha^y))$ to Bob.
12. Bob decrypts, asks Trent to verify that $ver_A$ is Alice's verification algorithm, and then uses $ver_A$ to verify Alice's signature.

# Key Distribution

- Strength of a cryptographic algorithm lies in the security of its keys
- If Alice and Bob are unable to meet in order to exchange their keys, can they still decide on a key without compromising further communications?
- Symmetric key cryptography, both Alice and Bob use same key for encryption and decryption
- Public key methods, RSA, where the sender has one key, and the receiver has another
- Public key encryption, keys are stored in public databases
- Computationally inefficient or slow, e.g., RSA is used to transmit a DES key

# Key Pre-distribution

- Alice wants to communicate with Bob
- Key schedule is decided in a
- Vance and this information is sent securely from one to the other
  - Used by German navy in World War II
  - Codebooks were captured from ships
- Problem with this method, Alice and Bob have to meet again, if keys can are compromised
- We need a trusted authority, Trent

# Key Pre-distribution

- For every pair of users, call them (*A*, *B*), Trent produces a random key $K_{AB}$ (to be used for symmetric key encryption, $K_{AB}=K_{BA}$)
- It is assumed that Trent is powerful and has established a secure channel to every user
- Each user will receive $n$-1 keys. Total key distributed is $n(n-1)/2$
- Blom key pre-distribution scheme is used
- Start with a large prime $p$ (known to all), with $n$ users ($p>=n$)

# Blom Key Pre-distribution

1. Each user $U$ in the network is assigned a distinct public number $r_U \pmod{p}$.

2. Trent chooses three secret random numbers $a$, $b$, and $c$ mod $p$.

3. For each user $U$, Trent calculates the numbers

$$a_U \equiv a + br_U \pmod{p} \quad b_U \equiv b + cr_U \pmod{p}$$

and sends them via his secure channel to $U$.

# Blom Key Pre-distribution

4. Each user $U$ forms the linear polynomial

$$g_U(x) = a_U + b_U x.$$

5. If Alice (A) wants to communicate with Bob (B), then Alice computes $K_{AB} = g_A(r_B)$, while Bob computes $K_{BA} = g_B(r_A)$.

6. It can be shown that $K_{AB} = K_{BA}$ (Exercise 2). Alice and Bob communicate via a symmetric encryption system, for example, DES, using the key (or a key derived from) $K_{AB}$.

# Example

- Network with Alice, Bob and Charlie
- $p$=23
- Let

$$r_A = 11, \quad r_B = 3, \quad r_C = 2.$$

- Suppose Trent chooses the number $a$=8, $b$=3, $c$=1
- Corresponding linear polynomials are

$$g_A(x) = 18 + 14x, \quad g_B(x) = 17 + 6x, \quad g_C(x) = 14 + 5x.$$

# Example

- It is now possible to calculate the keys that this scheme would generate

$$K_{AB} = g_A(r_B) = 14, \quad K_{AC} = g_A(r_C) = 0, \quad K_{BC} = g_B(r_C) = 6.$$

- Can be decoded if Eve and Oscar conspire

$$\begin{pmatrix} 0 & 1 & r_E \\ 1 & r_O & 0 \\ 0 & 1 & r_O \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} \equiv \begin{pmatrix} b_E \\ a_O \\ b_O \end{pmatrix} \pmod{p}.$$

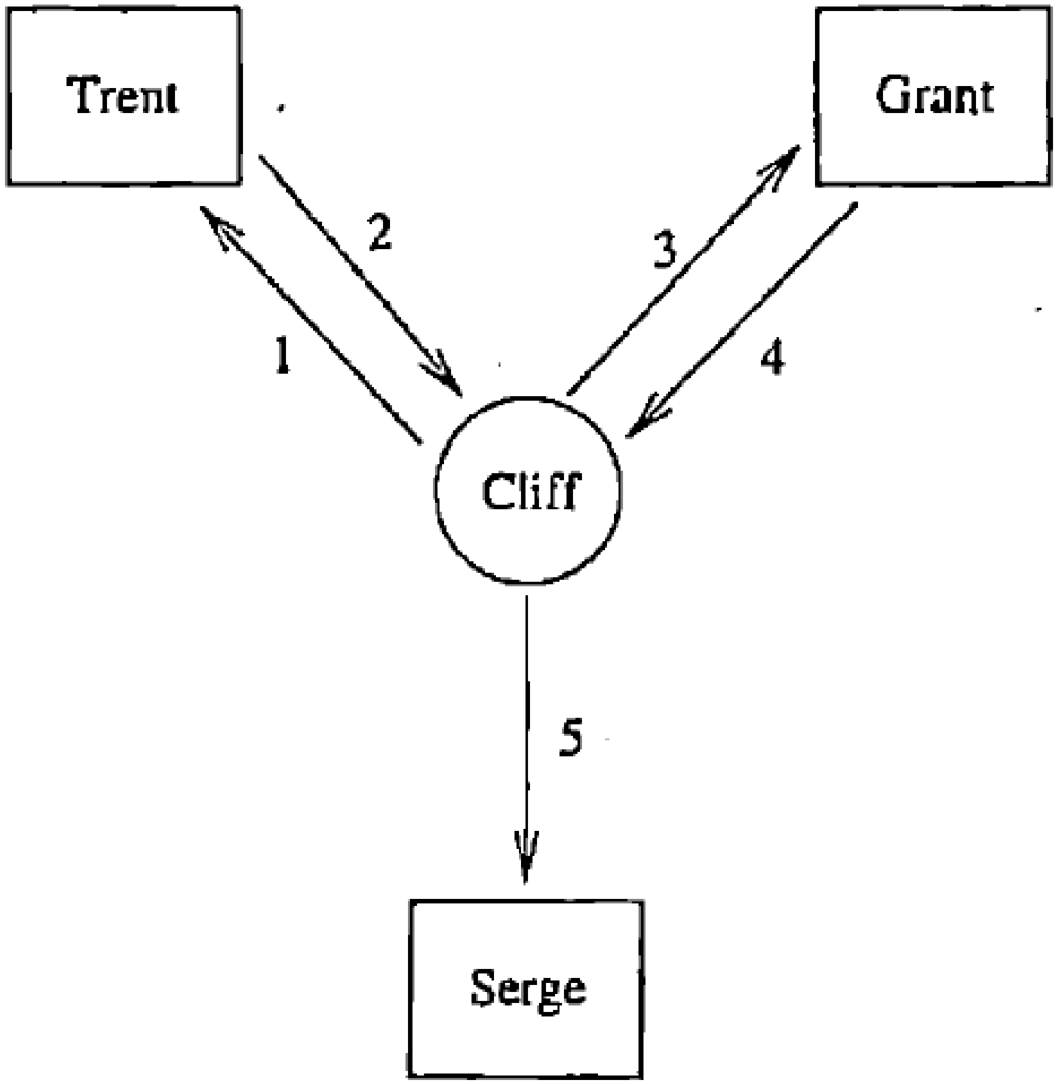
# Kerberos

- Kerberos developed at MIT as project Athena
- Objective is to provide huge network of computer workstations to students
- Insecure communications on public networks
- Client/server authentication model architecture

- Cliff: a client
- Serge: a server
- Trent: a trusted authority
- Grant: a ticket-granting server

# Kerberos

- Cliff and Serge have ne secret key information shared between them
- Serge needs to verify Cliff's identity



# Public Key Infrastructures (PKI)

- Public key cryptography allows for authentication, key distribution and non-repudiation
- Authenticity of published public keys (Eve may substitute her own public key in place of Alice’s key
- The benefits of public key depends on authenticity and validity
- Infrastructure to track public keys is required
- PKI infrastructure to define policies, procedures for publishing keys and certificates
- Certification binds a public key to an entity (user, or a piece of info)

# Certificate

- Certificate is a quantity of information that has been signed by its publisher, who is commonly referred to as the certification authority (CA)
- Identity certificates, and credential certificates (two types)
- Date encrypted through CA’s public key
- CA publishes identity certificates for Alice and Bob
- If Alike knows CA’s public key, she can verify Bob’s information
- Trusted company publishes manages the public key for Bob (govt or phone company)
- Many CA’s are operating