19CSE311- Computer Security- Unit II

January 30, 2022

Dr. Remya S/Mr. Sarath R
Assistant Professor
Department of Computer Science and Engineering

Agenda

- Security Services
- Authentication and Key Exchange Protocols
- Access Control Matrix
- User Authentication
- Directory Authentication Services
- Diffie Hellman Key Exchange Protocol
- Kerberos

Security Services

- The OSI Security Architecture is a framework that provides a systematic way of defining the requirements of security and characterizing the approaches to satisfying those requirements.
- The document defines **security attacks**, **mechanisms** and **services**, and the relationship among these categories.
- X.800 defines a security service as a service provided by a protocol layer of communicating open systems, which ensures adequate security of the system or of data transfers.

- A processing or communication service that is provided by a system to give a specific kind of protection to system resources.
- Security services implement security policies and are implemented by security mechanisms.
- The main security service are:
  - Authentication
  - Access Control
  - Data Confidentiality
  - Data Integrity
  - Non Repudiation

Basis for Authentication

- Something you know:
    - PIN
    - Password
- Something you have:
    - An access key or a card
    - A certificate
    - A smart card
    - A RFID tag
- Something you are:
    - Biometric

Weak Authentication

- PINs, Passwords provides weak authentication
  - Security is based upon how hard the pin/password is to guess.
  - Usually the passwords are short and weak.
  - Vulnerable to dictionary attacks.
- Widely used in practice
  - Unix, Web Emails, ...
- Protocol(A authenticates to B using a password P, that A shares with B)
  1. A $\rightarrow$ B: Hi, this is A!
  2. B $\rightarrow$ A: r(random challenge)
  3. A $\rightarrow B$: H(p,r)

Strong Authentication

- An entity authenticates to the other by proving the knowledge of a secret associated with that entity, without revealing anything meaningful about the secret itself.
- Can be achieved through:
  - Private/Public Key Encryption
  - MAC
  - Digital Signatures
- Strong because the security reduces to the security of the underlying cryptographic primitive, which is assumed to be hard to break.
- There exists both private key and public key based authentication.

Symmetric Encryption based Authentication

- Uses encryption to authenticate Alice to Bob (Assuming Alice-Bob have established a secret key K), then

A auth B

1. A $\rightarrow$ B: Hi Bob, this is Alice!
2. B $\rightarrow$ A: r(random challenge)
3. A $\rightarrow$ B: $Enc_k(r, B)$[response]

Security of the Previous Protocol

- An attacker needs to come up with a valid response.
  - Impossible if encryption is secure.
- r must not be re-used by Bob.

Freshness

- Assurance that message has not been used previously and originated within an acceptably recent time frame.
- Two methods:
  - Nonce (**N**umber used **once**)
  - Timestamps
- **Nonces**
  - One-time random number.
  - We depended on B to make sure r is a good nonce.
  - Choose nonces "randomly" from a large space (such as 2^{128}) to avoid reuse and for unpredictability- good RNG.

- **Timestamps**
  - Inclusion of date/time-stamp in the message allows recipient to check it for freshness.
  - This stamps need to be protected with cryptographic means.
  - A $\rightarrow$ B: $Enc_k(T, B)$, results in fewer messages and rounds
  - But, it requires synchronized clocks, which are hard to archieve in practice!

Encryption-based Mutual Authentication

- Run two copies of Uni-lateral authentication protocol $\rightarrow$ 4 rounds.
- We can piggyback common flows

  - **Method. 1**
    1. A $\rightarrow$ B: A,rA
    2. B $\rightarrow$ A: $Enc_k(rB, rA, a)$
    3. A $\rightarrow$ B: $Enc_k(rA, rB)$
  - **Method. 2**
    1. A $\rightarrow$ B: A,$Enc_k(T, B)$
    2. B $\rightarrow$ A: $Enc_k(T+1, A)$

Session Key Exchange with KDC- Needham Schroeder Protocol

- A $\rightarrow$ KDC: $ID_A||ID_B||N_1$
  (Hello, I am alice, I want to talk to Bob, I need a section key and here is a randome nonce idetifying the request)
- KDC $\rightarrow$ A : $E_{K_A}(K||ID_B||N_1||E_{K_B}(K||ID_A))$
  Encrypted(Here is a key, for you to talk to Bob as per your request N_1 ans also an envelope to Bob containing the same key)
- A $\rightarrow$ B: $E_{K_B}(K||ID_A)$
  (I would like to talk using key in envelope sent by KDC)
- B$\rightarrow$A :$E_K(N_2)$
  (OK Alice, But can you prove to me that you are indeed Alice and know the key?)
- A$\rightarrow$B : $E_K(f(N_2))$ (Sure, I can!)

Session Key Exchange with KDC- Needham Schroeder Protocol (Corrected version with mutual authentication)

- A $\rightarrow$ KDC : $ID_A||ID_B||N_1$
  (Hello, I am alice, I want to talk to Bob, I need a section key and here is a randome nonce idetifying the request)
- KDC $\rightarrow$ A: $E_{K_A}(K||ID_B||N_1||E_{K_B}(TS1, K||ID_A))$
  Encrypted(Here is a key, for you to talk to Bob as per your request N_1 ans also an envelope to Bob containing the same key)
- A $\rightarrow$B : $E_K(TS2, B), E_{K_B}(TS1, K||ID_A)$
  I would like to talk using key in envelope sent by KDC, here is an authenticator
- B $\rightarrow$A : $E_K(TS2 + 1, A)$
  OK Alice, here is a proof that i am really Bob

Version 4 Summary

(a) Authentication Service Exchange: to obtain ticket-granting ticket

(1) **C → AS:** $ID_c \parallel ID_{tgs} \parallel TS_1$

(2) **AS → C:** $E_{K_c}[K_{c,tgs} \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_{tgs}]$

$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$

(b) Ticket-Granting Service Exchange: to obtain service-granting ticket

(3) **C → TGS:** $ID_v \parallel Ticket_{tgs} \parallel Authenticator_c$

(4) **TGS → C:** $E_{K_{c,tgs}}[K_{c,v} \parallel ID_v \parallel TS_4 \parallel Ticket_v]$

$Ticket_{tgs} = E_{K_{tgs}}[K_{c,tgs} \parallel ID_c \parallel AD_c \parallel ID_{tgs} \parallel TS_2 \parallel Lifetime_2]$

$Ticket_v = E_{K_v}[K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$

$Authenticator_c = E_{K_{c,tgs}}[ID_c \parallel AD_c \parallel TS_3]$

(c) Client/Server Authentication Exchange: to obtain service

(5) **C → K:** $Ticket_v \parallel Authenticator_c$

(6) **K → C:** $E_{K_{c,v}}[TS_5 + 1]$ (for mutual authentication)

$Ticket_v = E_{K_v}[K_{c,v} \parallel ID_c \parallel AD_c \parallel ID_v \parallel TS_4 \parallel Lifetime_4]$

$Authenticator_c = E_{K_{c,v}}[ID_c \parallel AD_c \parallel TS_5]$

MAC Based Authentication

- Message Authentication Code(MAC) algorithm is a symmetric key cryptographic technique to provide message authentication.
- For establishing MAC process, the sender and receiver share a secret key K.
- Essentially, a MAC is am encrypted checksum generated on the underlying message that is sent along with a message to ensure message authentication.
- But in entity authentication, the procedure will be like as follows:
  1. A→B A, rA
  2. B→A rB, $HMAC_K(rB, rA, A)$
  3. A→B $HMAC_K(rA, rB, B)$
- Faster than encryption based protocols(computationally)

Public Key based Authentication(Needham-Shroeder(NS) PK-based)

- Assuming public keys are distributed between entities,
  1. A→B $Enc_{pkb}(rA, A)$
  2. B→A $Enc_{pka}(rA, rB)$
  3. A→B $Enc_{pkb}(rB)$
- Since the public key is available to everyone, there is chances of attack, can be solved by:
  1. A→B $Enc_{pkb}(rA, A)$
  2. B→A $Enc_{pka}(rA, rB, B)$
  3. A→B $Enc_{pkb}(rB)$

Signature Based Authentication(Assuming public keys are distributed between entities)

- **A auth B**
  1. A→B: Hi Bob, this is Alice!
  2. B→A: r(a challenge)
  3. A→B: $Sig_{SK_a}(r, B)$(response)
- **A auth B, B auth A**(run two copies; piggyback common flows)
  1. A→B: A,rA(could sign this too)
  2. B→A: rB,$Sig_{SK_b}(rB, rA, A)$
  3. A→B: $Sig_{SK_a}(rA, rB, B)$

Authenticated Key Exchange(AKE)

- Public-key operations are costly.
- Why not
  1. Use public key mutual authentication protocols to exchange a symmetric key.
  2. Use this symmetric key with a symmetric encryption to secure subsequent communication.
- Authenticated key exchange or Authenticated key agreement is the exchange of session key in a key exchange protocol which also authenticates the identities of parties involved in key exchange.

AKE Protocol

1. A→B: A,rA,$Enc_{PK_b}(K)$
2. B→A: rB,$Sig_{SK_b}(rB, rA, A)$
3. A→B: $Sig_{SK_a}(rA, rB, B)$
4. A and B output K as the authenticated key.

- Such a protocol can be instantiated using RSA encryption/signing.
- Generally only the server authenticates to the client, not vice versa.

X.509: One Way Authentication

- 1 message(A→B) used to establish
  - the identity of A and that message is from A
  - message was intended for B
  - integrity originality of message

X.509: Two Way Authentication

- 2 messages(A→B,B→A)which also establishes in addition:
  - the identity of B and that reply is from B
  - that reply is intended for A
  - integrity and originality of reply

X.509: Three Way Authentication

- 3 messages(A→B,B→A,A→B) which enables above authentication without the need for synchronised clocks.