3. User Authentication and Network Access Control

IT6406 - Network Security and Audit

Level III - Semester 6

© 2022 e-Learning Centre, UCSC

Overview

This section examines some of the authentication functions that have been developed to support network-based user authentication. The section contains an introduction to some of the concepts and key considerations for user authentication over a network or the Internet, user-authentication protocols that rely on symmetric encryption, Kerberos, user-authentication protocols that rely on asymmetric encryption, concept of federated identity, and network access control and authentication mechanisms.

© 2022 e-Learning Centre, UCSC

Overview

At the end of this lesson, you will be able to;

- Explain the distinction between identification and verification
- Describe the techniques for remote user authentication using symmetric encryption.
- Explain Kerberos protocol
- Describe the use of Kerberos in multiple realms
- Describe the techniques for remote user authentication using asymmetric encryption
- Understand the need for a federated identity management system
- Describe the Extensible Authentication Protocol.
- Understand the operation and role of the IEEE 802.1X Port-Based Network Access Control mechanism.
- Discuss the principal network access enforcement methods.

© 2022 e-Learning Centre, UCSC

Overview

3.1 Remote User-Authentication Principles
- 3.1.1 Means of Authentication
- 3.1.2 Mutual Authentication
- 3.1.3 one-Way Authentication

3.2 Remote User-Authentication Using Symmetric Encryption
- 3.2.1 Mutual Authentication
- 3.2.2 One Way Authentication

3.3 Kerberos
- 3.3.1. A simple authentication dialogue
- 3.3.2. A more secure authentication dialogue
- 3.3.3. The version 4 authentication dialogue
- 3.3.4. Kerberos realms and multiple Kerberi
- 3.3.5. Difference between version 4 and 5
- 3.3.6. The version 4 authentication dialogue

© 2022 e-Learning Centre, UCSC

Overview

3.4 Remote User-Authentication Using Asymmetric Encryption
- 3.4.1. Mutual Authentication
- 3.4.2. One Way Authentication

3.5 Federated Identity Management
- 3.5.1. Identity Management
- 3.5.2. Identity Federation

3.6 Network Access Control
- 3.6.1. Elements of a Network Access Control System

3.7 Extensible Authentication Protocol
- 3.7.1. Authentication Methods
- 3.7.2. Network Access Enforcement Methods
- 3.7.3. EAP Protocol Exchanges
- 3.7.4. EAP Message Flow in Pass Through Mode

3.8 IEEE 802.1X Port-Based Network Access Control

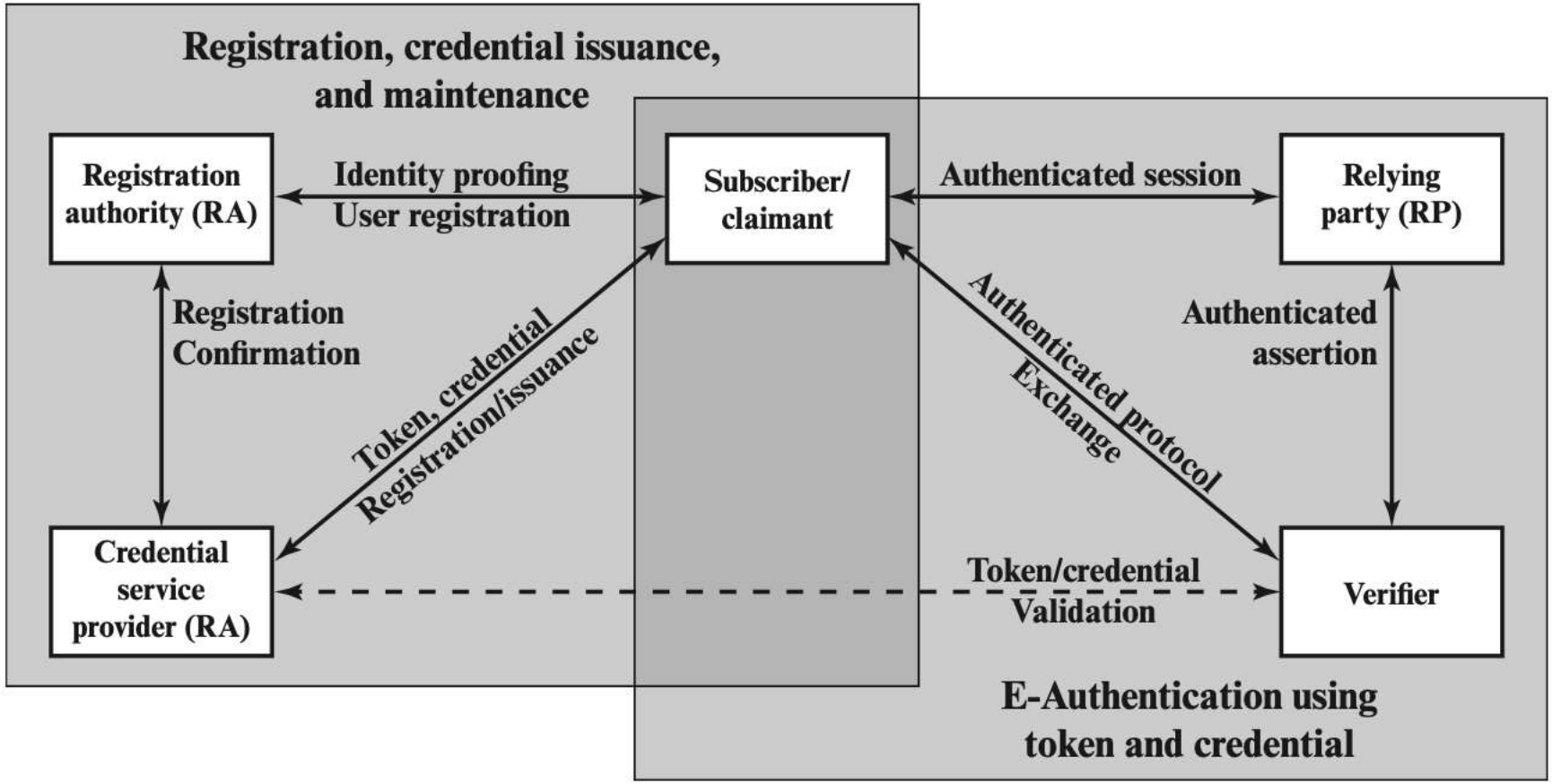
© 2022 e-Learning Centre, UCSC

3.1 Remote User-Authentication Principles

- User authentication is the basis for most types of access control and for user accountability
- This process consists of two steps:
  - Identification step : Presenting an identifier to the security system
  - Verification step : Presenting or generating authentication information that corroborates the binding between the entity and the identifier
  - Use a user login scenario to explain the two steps.

© 2022 e-Learning Centre, UCSC

3.1 Remote User-Authentication Principles



The NIST SP 800-63-2 E-Authentication Architectural Model : Reference 01

© 2022 e-Learning Centre, UCSC

3.1 Remote User-Authentication Principles

3.1.1 Means of Authentication

- **Something the individual knows**: Examples include a password, a personal identification number (PIN), or answers to a prearranged set of question
- **Something the individual possesses**: Examples include cryptographic keys, electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token.
- **Something the individual is (static biometrics)**: Examples include recognition by fingerprint, retina, and face.
- **Something the individual does (dynamic biometrics)**: Something the individual does (dynamic biometrics):

© 2022 e-Learning Centre, UCSC

3.1 Remote User-Authentication Principles

3.1.2 Mutual Authentication

- Mutual authentication protocols enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys.
- Central to the problem of authenticated key exchange are two issues:
  - Confidentiality
  - Timeliness

© 2022 e-Learning Centre, UCSC

3.1 Remote User-Authentication Principles

3.1.2 Mutual Authentication

- Examples for replay attacks
  - The simplest replay attack is one in which the opponent simply copies a message and replays it later
  - An opponent can replay a timestamped message within the valid time window. If both the original and the replay arrive within then time window, this inci- dent can be logged.
  - As with above example, an opponent can replay a timestamped message within the valid time window, but in addition, the opponent suppresses the original message. Thus, the repetition cannot be detected.
  - Another attack involves a backward replay without modification. This is a re- play back to the message sender. This attack is possible if symmetric encryption is used and the sender cannot easily recognize the difference between messages sent and messages received on the basis of content.

© 2022 e-Learning Centre, UCSC

3.1 Remote User-Authentication Principles

3.1.2 Mutual Authentication

- Preventing replay attacks
  - Timestamps
  - Challenge/response

© 2022 e-Learning Centre, UCSC

3.1 Remote User-Authentication Principles

3.1.3 One-Way Authentication

- it is not necessary for the sender and receiver to be online at the same time
- The "envelope" or header of the email message must be in the clear, so that the message can be handled by the store-and-forward email protocol
- the mail-handling protocol not require access to the plaintext form of the message
- Therefore, using encrypted emails we can do one way authentication

© 2022 e-Learning Centre, UCSC

3.2 Remote User-Authentication Using Symmetric Encryption

3.2.1 Mutual Authentication

- Two-level hierarchy of symmetric encryption keys can be used to provide confidentiality for communication in a distributed environment
- A proposal by Needham and Schroeder on secret key (K) distribution using a trusted Key Distribution Center (KDC).
- Despite the handshake of steps 4 and 5, the protocol is still vulnerable to a form of replay attack. Identify the attack.

1. $A \rightarrow KDC: \quad ID_A \| ID_B \| N_1$
2. $KDC \rightarrow A: \quad E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$
3. $A \rightarrow B: \quad E(K_b, [K_s \| ID_A])$
4. $B \rightarrow A: \quad E(K_s, N_2)$
5. $A \rightarrow B: \quad E(K_s, f(N_2))$ where f() is a generic function that modifies the value of the nonce.

© 2022 e-Learning Centre, UCSC

3.2 Remote User-Authentication Using Symmetric Encryption

3.2.1 Mutual Authentication

- Denning proposes to overcome this weakness by a modification to the Needham/Schroeder protocol that includes the addition of a timestamps
- T is a timestamp that assures A and B that the session key has only just been generated.
- The protocol is still vulnerable to a form of suppress-replay attack if the clocks are not synchronised and A's clock is in future than B's which would give attacker more time to try and compromise K_s and launch a replay.

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(K_a, [K_s \| ID_B \| T \| E(K_b, [K_s \| ID_A \| T])])$
3. $A \rightarrow B: \quad E(K_b, [K_s \| ID_A \| T])$
4. $B \rightarrow A: \quad E(K_s, N_1)$
5. $A \rightarrow B: \quad E(K_s, f(N_1))$

© 2022 e-Learning Centre, UCSC

3.2 Remote User-Authentication Using Symmetric Encryption

3.2.1 Mutual Authentication

- A more robust solution.

1. $A \rightarrow B: \quad ID_A \| N_a$
2. $B \rightarrow KDC: \quad ID_B \| N_b \| E(K_b, [ID_A \| N_a \| T_b])$
3. $KDC \rightarrow A: \quad E(K_a, [ID_B \| N_a \| K_s \| T_b]) \| E(K_b, [ID_A \| K_s \| T_b]) \| N_b$
4. $A \rightarrow B: \quad E(K_b, [ID_A \| K_s \| T_b]) \| E(K_s, N_b)$

© 2022 e-Learning Centre, UCSC

3.2 Remote User-Authentication Using Symmetric Encryption

3.2.2 One Way Authentication

- One of the limitations of this is that the sender cannot expect the recipient to be online.
- The protocol does not protect against replays. Some measure of defence could be provided by including a timestamp with the message.

1. $A \rightarrow KDC: \quad ID_A \| ID_B \| N_1$
2. $KDC \rightarrow A: \quad E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$
3. $A \rightarrow B: \quad E(K_b, [K_s \| ID_A]) \| E(K_s, M)$

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- The problem that Kerberos addresses is:
  - Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restrict access to authorized users and to be able to authenticate requests for service. In this environment, a workstation cannot be trusted to identify its users correctly to network services.

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- Requirements for Kerberos
  - **Secure**: A network eavesdropper should not be able to obtain the necessary information to impersonate a user. More generally, Kerberos should be strong enough that a potential opponent does not find it to be the weak link.
  - **Reliable**: For all services that rely on Kerberos for access control, lack of availability of the Kerberos service means lack of availability of the supported services. Hence, Kerberos should be highly reliable and should employ a distributed server architecture with one system able to back up another.
  - **Transparent**: Ideally, the user should not be aware that authentication is taking place beyond the requirement to enter a password.
  - **Scalable**: The system should be capable of supporting large numbers of clients and servers. This suggests a modular, distributed architecture.

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.1. A simple authentication dialogue
  - In an unprotected network environment, any client can apply to any server for service
  - The obvious security risk is that of impersonation.
  - To counter this threat, servers must be able to confirm the identities of clients who request service.
  - Each server can be required to undertake this task for each client/server interaction which is not scalable solution.

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.1. A simple authentication dialogue
  - An alternative is to use an authentication server (AS) that knows the passwords of all users/services and stores these in a centralized database.

$$
\begin{aligned}
&\textbf{(1)}\ \mathrm{C} \rightarrow \mathrm{AS}: && ID_C \| P_C \| ID_V \\
&\textbf{(2)}\ \mathrm{AS} \rightarrow \mathrm{C}: && Ticket \\
&\textbf{(3)}\ \mathrm{C} \rightarrow \mathrm{V}: && ID_C \| Ticket \\
&Ticket = E(K_v, [ID_C \| AD_C \| ID_V])
\end{aligned}
$$

where

C = client
AS = authentication server
V = server
ID_C = identifier of user on C
ID_V = identifier of V
P_C = password of user on C
AD_C = network address of C
K_v = secret encryption key shared by AS and V

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.1. A simple authentication dialogue
  - Issues of the proposed solution:
    - we would like to minimize the number of times that a user has to enter a password
    - An eavesdropper could capture the password and use any service accessible to the victim
    - Using the ticket to conduct replay attacks.

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.2. A more secure authentication dialogue

Once per user logon session:

(1) $C \rightarrow AS: \quad ID_C \| ID_{tgs}$

(2) $AS \rightarrow C: \quad E(K_c, Ticket_{tgs})$

Once per type of service:

(3) $C \rightarrow TGS: \quad ID_C \| ID_V \| Ticket_{tgs}$

(4) $TGS \rightarrow C: \quad Ticket_v$

Once per service session:

(5) $C \rightarrow V: \quad ID_C \| Ticket_v$

$$Ticket_{tgs} = E(K_{tgs}, [ID_C \| AD_C \| ID_{tgs} \| TS_1 \| Lifetime_1])$$

$$Ticket_v = E(K_v, [ID_C \| AD_C \| ID_v \| TS_2 \| Lifetime_2])$$

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.2. A more secure authentication dialogue
- Issues of the proposed solution:
    - Replay attacks using tickets
    - Lifetime related issues
- Thus, we arrive at an additional requirement. A network service (the TGS or an application service) must be able to prove that the person using a ticket is the same person to whom that ticket was issued.
- The second problem is that there may be a requirement for servers to authenticate themselves to users.

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.3. The version 4 authentication dialogue

(1) $\mathbf{C} \rightarrow \mathbf{AS}$ $ID_c \| ID_{tgs} \| TS_1$

(2) $\mathbf{AS} \rightarrow \mathbf{C}$ $\mathrm{E}(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$$Ticket_{tgs} = \mathrm{E}(\mathrm{K}_{tgs}, [\mathrm{K}_{c,tgs} \| \mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{ID}_{tgs} \| \mathrm{TS}_2 \| \mathrm{Lifetime}_2])$$

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

(3) $\mathbf{C} \rightarrow \mathbf{TGS}$ $ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) $\mathbf{TGS} \rightarrow \mathbf{C}$ $\mathrm{E}(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$$Ticket_{tgs} = \mathrm{E}(\mathrm{K}_{tgs}, [\mathrm{K}_{c,tgs} \| \mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{ID}_{tgs} \| \mathrm{TS}_2 \| \mathrm{Lifetime}_2])$$

$$Ticket_v = \mathrm{E}(\mathrm{K}_v, [\mathrm{K}_{c,v} \| \mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{ID}_v \| \mathrm{TS}_4 \| \mathrm{Lifetime}_4])$$

$$Authenticator_c = \mathrm{E}(\mathrm{K}_{c,tgs}, [\mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{TS}_3])$$

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

(5) $\mathbf{C} \rightarrow \mathbf{V}$ $Ticket_v \| Authenticator_c$

(6) $\mathbf{V} \rightarrow \mathbf{C}$ $\mathrm{E}(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$$Ticket_v = \mathrm{E}(\mathrm{K}_v, [\mathrm{K}_{c,v} \| \mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{ID}_v \| \mathrm{TS}_4 \| \mathrm{Lifetime}_4])$$

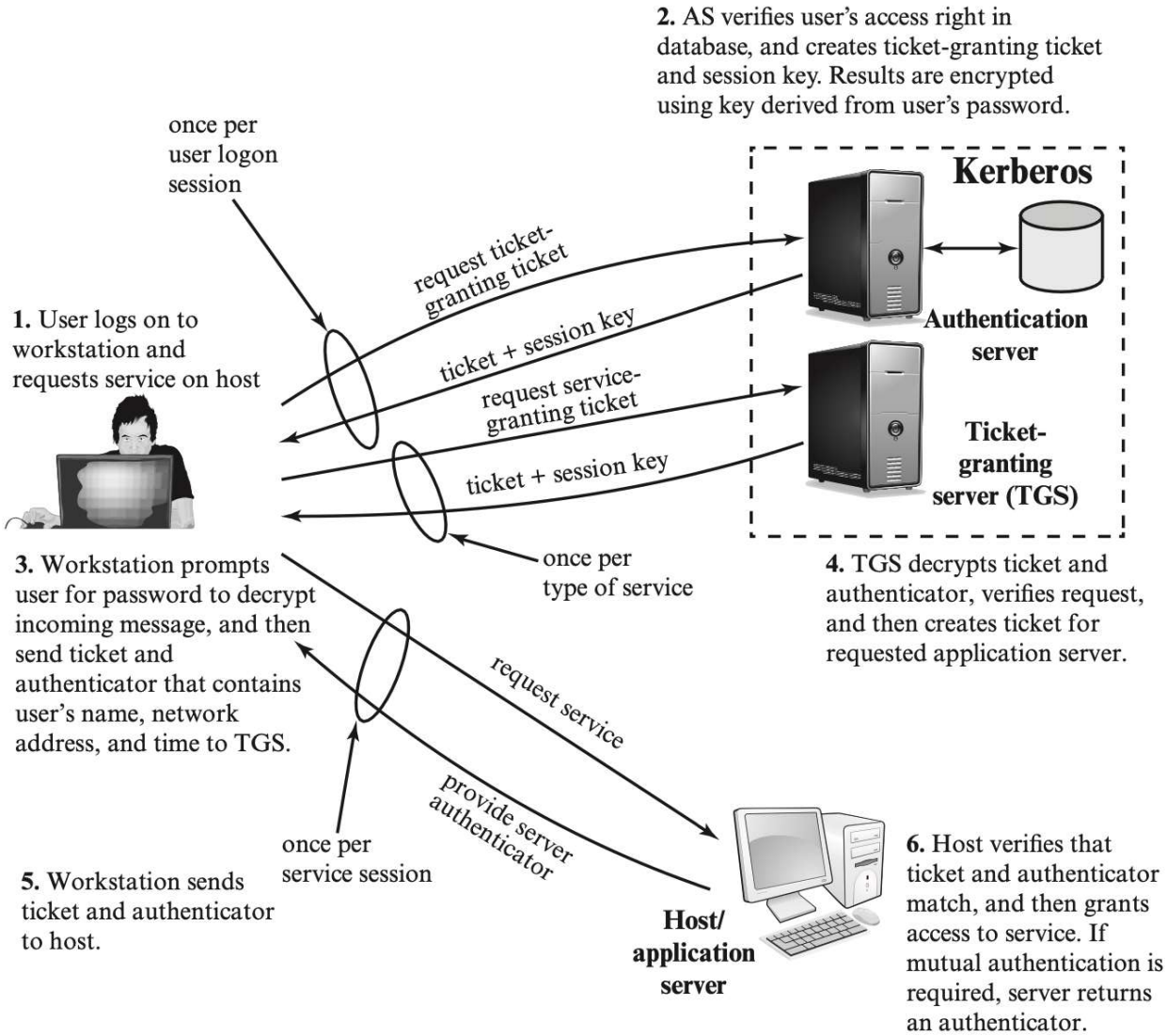
$$Authenticator_c = \mathrm{E}(\mathrm{K}_{c,v}, [\mathrm{ID}_C \| \mathrm{AD}_C \| \mathrm{TS}_5])$$

(c) Client/Server Authentication Exchange to obtain service

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.3. The version 4 authentication dialogue



© 2022 e-Learning Centre, UCSC

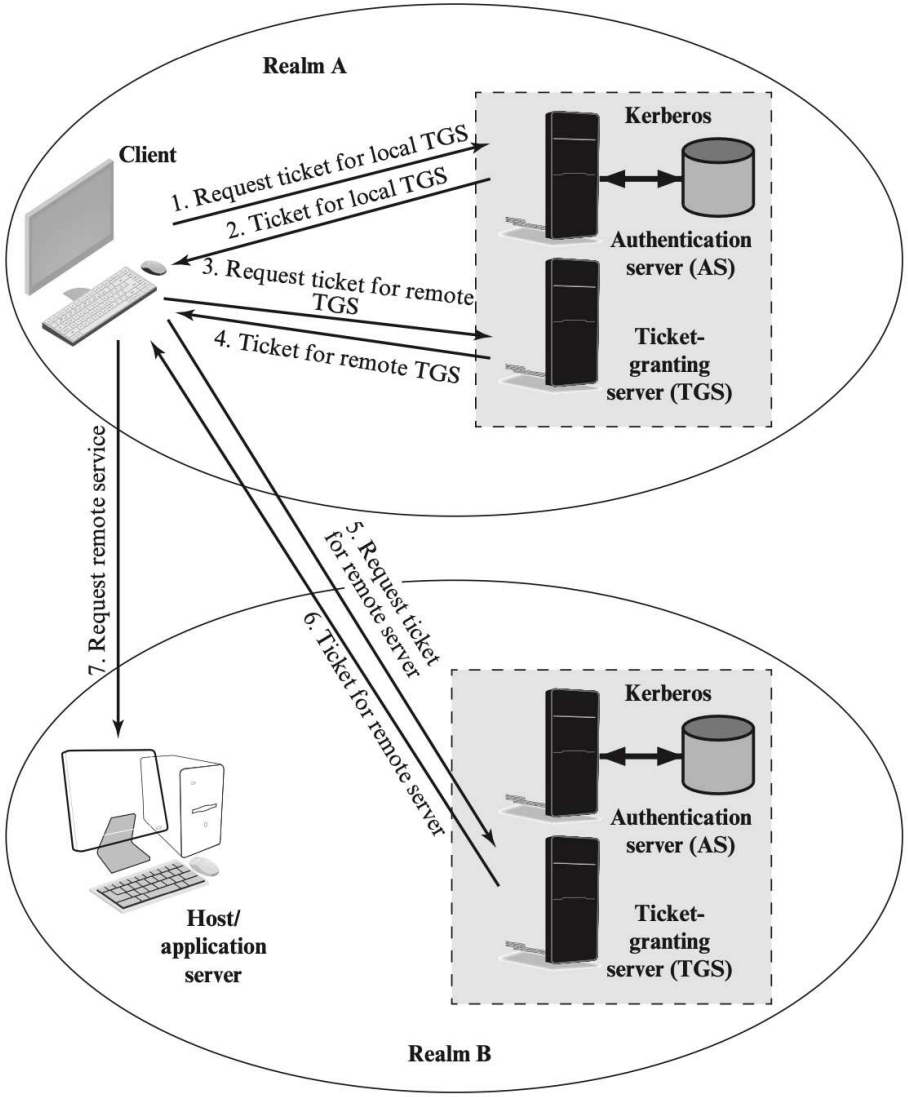
3.3 Kerberos

- 3.3.4. Kerberos realms multiple Kerberi
- The Kerberos server must have the user ID and hashed passwords of all participating users in its database. All users are registered with the Kerberos server.
- The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server.
- The Kerberos server in each interoperating realm shares a secret key with the server in the other realm. The two Kerberos servers are registered with each other.

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.4. Kerberos realms multiple Kerberi



© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.5. Differences between version 4 and 5
  - Kerberos V4 environmental shortcomings
    - Encryption system dependence
    - Internet protocol dependence
    - Message byte ordering
    - Ticket lifetime
    - Authentication forwarding
    - Inter-realm authentication

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.5. Differences between version 4 and 5
  - Kerberos V4 technical deficiencies address in V5
    - Double encryption
    - PCBC encryption
    - Session keys
    - Password attacks

© 2022 e-Learning Centre, UCSC

3.3 Kerberos

- 3.3.6. The version 5 authentication dialogue

(1) $\mathbf{C} \rightarrow \mathbf{AS}$ $Options \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$

(2) $\mathbf{AS} \rightarrow \mathbf{C}$ $Realm_C \| ID_C \| Ticket_{tgs} \| \mathrm{E}(K_c, [K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}])$

$$Ticket_{tgs} = \mathrm{E}(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$$

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

(3) $\mathbf{C} \rightarrow \mathbf{TGS}$ $Options \| ID_v \| Times \| Nonce_2 \| Ticket_{tgs} \| Authenticator_c$

(4) $\mathbf{TGS} \rightarrow \mathbf{C}$ $Realm_c \| ID_C \| Ticket_v \| \mathrm{E}(K_{c,tgs}, [K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v])$

$$Ticket_{tgs} = \mathrm{E}(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$$
$$Ticket_v = \mathrm{E}(K_v, [Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$$
$$Authenticator_c = \mathrm{E}(K_{c,tgs}, [ID_C \| Realm_c \| TS_1])$$

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

(5) $\mathbf{C} \rightarrow \mathbf{V}$ $Options \| Ticket_v \| Authenticator_c$

(6) $\mathbf{V} \rightarrow \mathbf{C}$ $\mathrm{E}_{K_{c,v}}[TS_2 \| Subkey \| Seq\ \#]$

$$Ticket_v = \mathrm{E}(K_v, [Flag \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$$
$$Authenticator_c = \mathrm{E}(K_{c,v}, [ID_C \| Relam_c \| TS_2 \| Subkey \| Seq\ \#])$$

(c) Client/Server Authentication Exchange to obtain service

© 2022 e-Learning Centre, UCSC

3.4 Remote User-Authentication Using Asymmetric Encryption

- 3.4.1. Mutual Authentication

1. $A \rightarrow KDC: \quad ID_A \| ID_B$
2. $KDC \rightarrow A: \quad E(PR_{auth}, [ID_B \| PU_b])$
3. $A \rightarrow B: \quad E(PU_b, [N_a \| ID_A])$
4. $B \rightarrow KDC: \quad ID_A \| ID_B \| E(PU_{auth}, N_a)$
5. $KDC \rightarrow B: \quad E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_A \| ID_B]))$
6. $B \rightarrow A: \quad E(PU_a, [N_b \| E(PR_{auth}, [N_a \| K_s \| ID_A \| ID_B])])$
7. $A \rightarrow B: \quad E(K_s, N_b)$

© 2022 e-Learning Centre, UCSC

3.4 Remote User-Authentication Using Asymmetric Encryption

- 3.4.2. One Way Authentication

- If confidentiality is the primary concern, then the following may be more efficient

$$\mathrm{A} \rightarrow \mathrm{B}: \quad \mathrm{E}(PU_b, K_s) \,\|\, \mathrm{E}(K_s, M)$$

- If authentication is the primary concern, then a digital signature may suffice

$$\mathrm{A} \rightarrow \mathrm{B}: \quad M \,\|\, \mathrm{E}(PR_a, \mathrm{H}(M))$$

© 2022 e-Learning Centre, UCSC

3.5 Federated Identity Management

- 3.5.1. Identity Management

- Identity management is a centralized, automated approach to provide enterprise wide access to resources by employees and other authorized individuals
- The central concept of an identity management system is the use of single sign-on (SSO)

© 2022 e-Learning Centre, UCSC

3.5 Federated Identity Management

- 3.5.1. Identity Management
- Typical services provided by a federated identity management system include the following
  - Point of contact
  - SSO protocol services
  - Trust services
  - Key services
  - Identity services
  - Authorization
  - Provisioning
  - Management
- Note that Kerberos contains a number of the elements of an identity management system

© 2022 e-Learning Centre, UCSC

3.5 Federated Identity Management

- 3.5.2. Identity Federation
- Identity federation is, in essence, an extension of identity management to multiple security domains
- Federated identity management refers to the agreements, standards, and technologies that enable the portability of identities, identity attributes, and entitlements across multiple enterprises and numerous applications and supporting many thousands, even millions, of users
- Federated identity management uses a number of standards as the building blocks for secure identity exchange across different domains or heterogeneous systems.

© 2022 e-Learning Centre, UCSC

3.6 Network Access Control

- 3.6.1. Elements of a Network Access Control System
- Access requestor (AR)
- Policy server
- Network access server (NAS)

© 2022 e-Learning Centre, UCSC

3.7 Extensible Authentication Protocol

- 3.7.1. Authentication Methods
- EAP-TLS (EAP Transport Layer Security)
- EAP-TTLS (EAP Tunneled TLS)
- EAP-GPSK (EAP Generalized Pre-Shared Key)
- EAP-IKEv2

© 2022 e-Learning Centre, UCSC

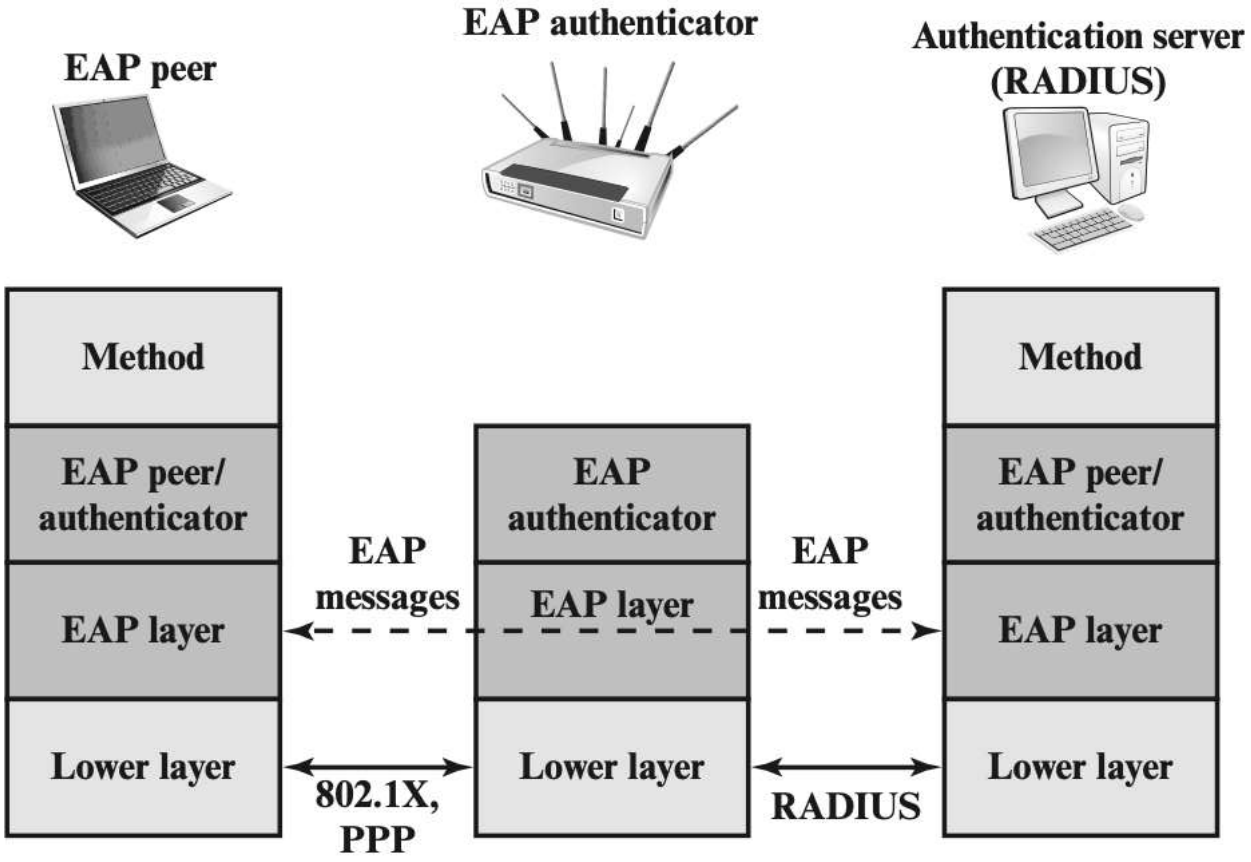
3.7 Extensible Authentication Protocol

- 3.7.2. Network Access Enforcement Methods
- IEEE 802.1X
- Virtual local area networks (VLANs)
- Firewall
- DHCP management

© 2022 e-Learning Centre, UCSC

3.7 Extensible Authentication Protocol

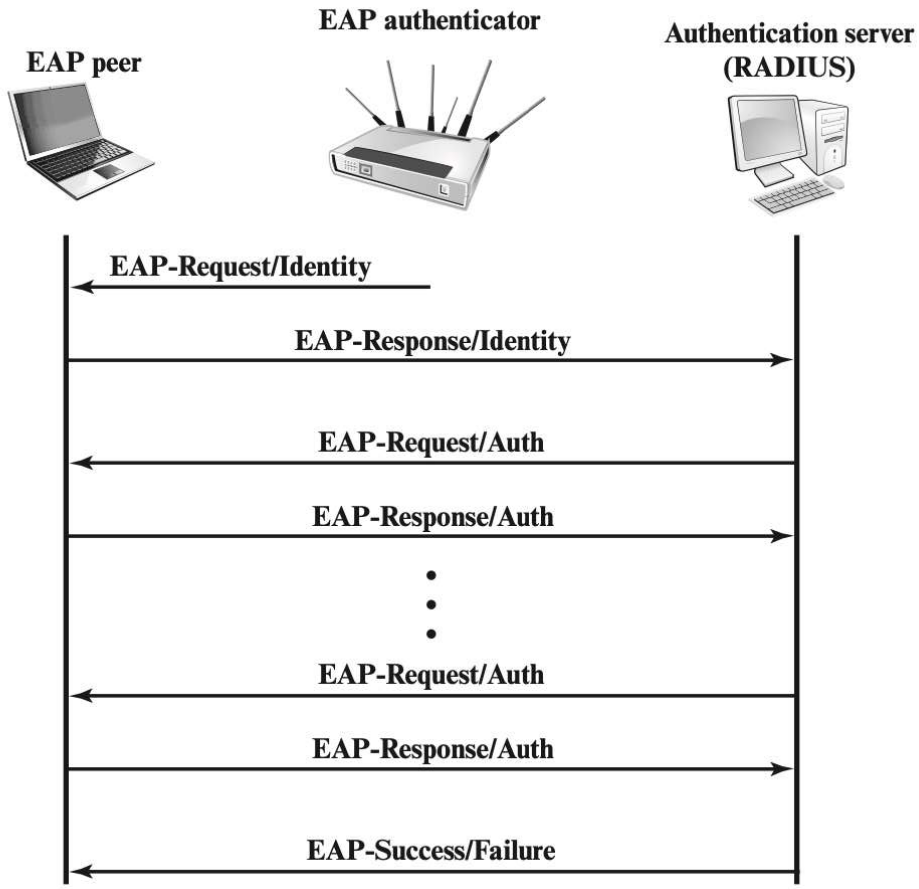
- 3.7.3. EAP Protocol Exchanges



© 2022 e-Learning Centre, UCSC

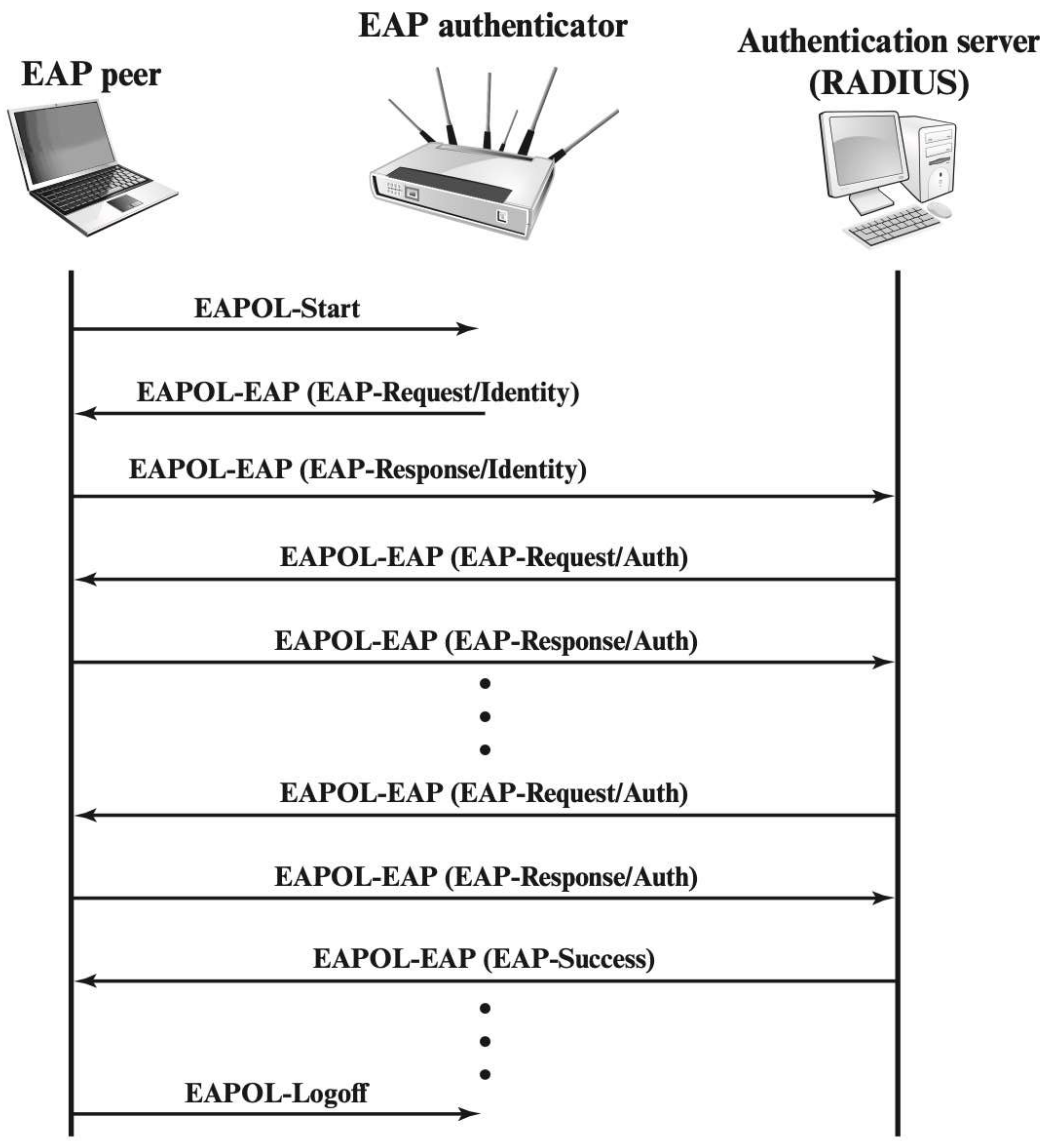
3.7 Extensible Authentication Protocol

- 3.7.4. EAP Message Flow in Pass Through Mode



© 2022 e-Learning Centre, UCSC

3.8 IEEE 802.1X Port-Based Network Access Control



© 2022 e-Learning Centre, UCSC