User Authentication Protocols

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

Audio/Video recordings of this lecture are available at:

http://www.cse.wustl.edu/~jain/cse571-11/

1. Remote User Authentication Using Secret Keys
2. Kerberos V4
3. Kerberos V5
4. Remote User Authentication Using Public Keys
5. Federated Identity Management

These slides are based partly on Lawrie Brown's slides supplied with William Stallings's book "Cryptography and Network Security: Principles and Practice," 5th Ed, 2011.

User Authentication

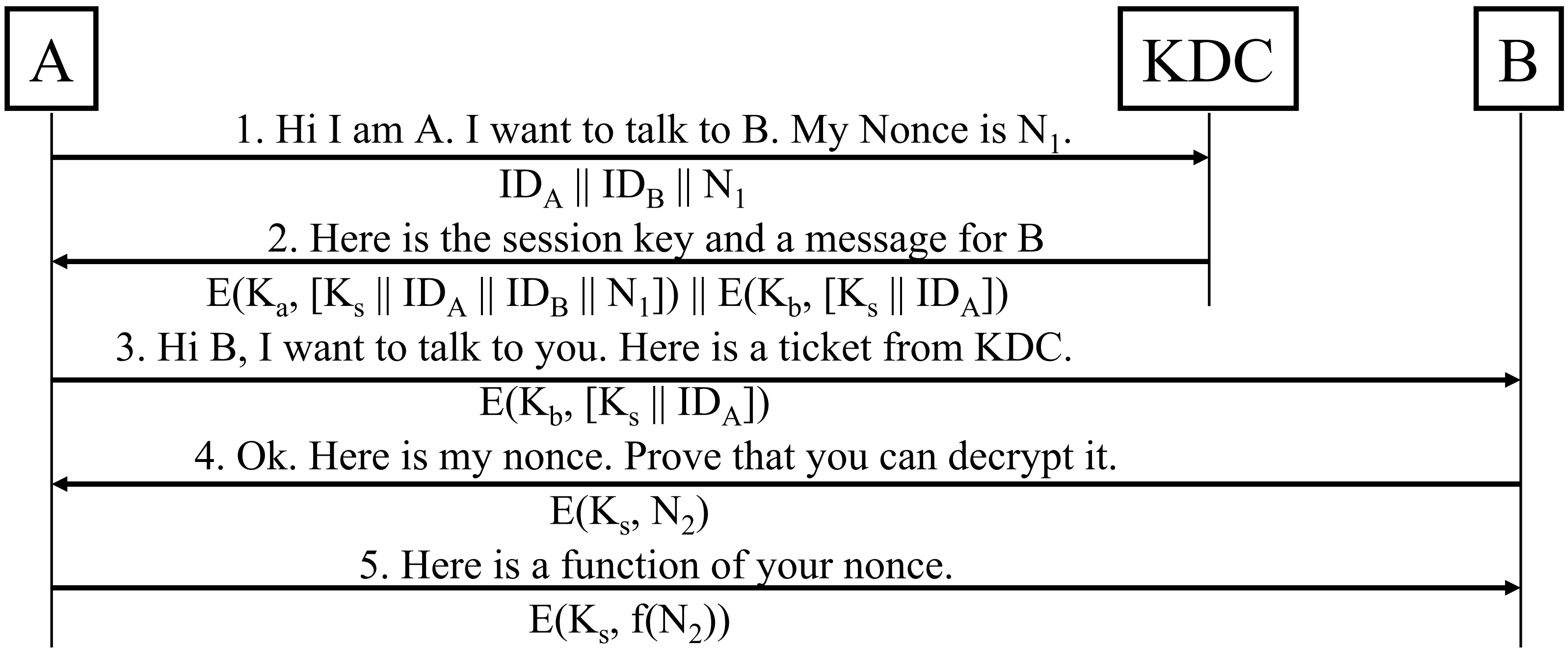
- Four means of authenticating user's identity: Based on something the individual
  1. Knows - e.g., password, PIN
  2. Possesses - e.g., key, token, smartcard
  3. Is (static biometrics) - e.g., fingerprint, retina
  4. Does (dynamic biometrics) - e.g., voice, sign
- Can use alone or combined. All have issues
- May be one-way or mutual
- Key issues are
  - Confidentiality – to protect session keys
  - Timeliness – to prevent replay attacks

Replay Attacks

- A valid signed message is copied and later resent. Examples:
  - Simple replay: No timestamp
  - Repetition that can be logged: time stamped message within valid time
  - Repetition that cannot be detected: Original message replaced with a new message
  - Backward replay without modification: Source's message back to the source
- Countermeasures include
  - Use of sequence numbers (generally impractical)
  - Timestamps (needs synchronized clocks)
  - Challenge/response (using unique nonce)

Needham Schroeder Protocol

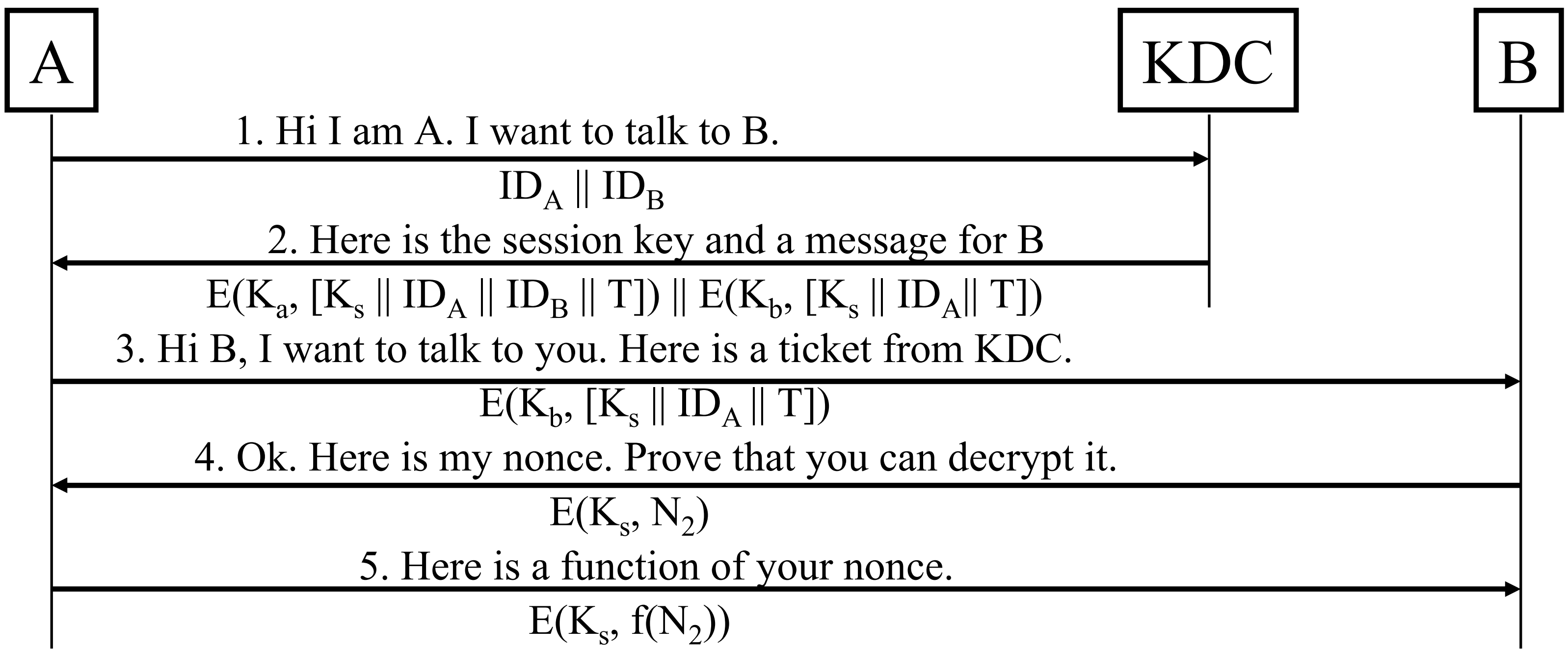
- Everyone has a shared secret key with KDC
- KDC generates session keys



- If someone can crack one K_S, he can replay 3, block 4. Then masquerade as A.

Denning's Modification

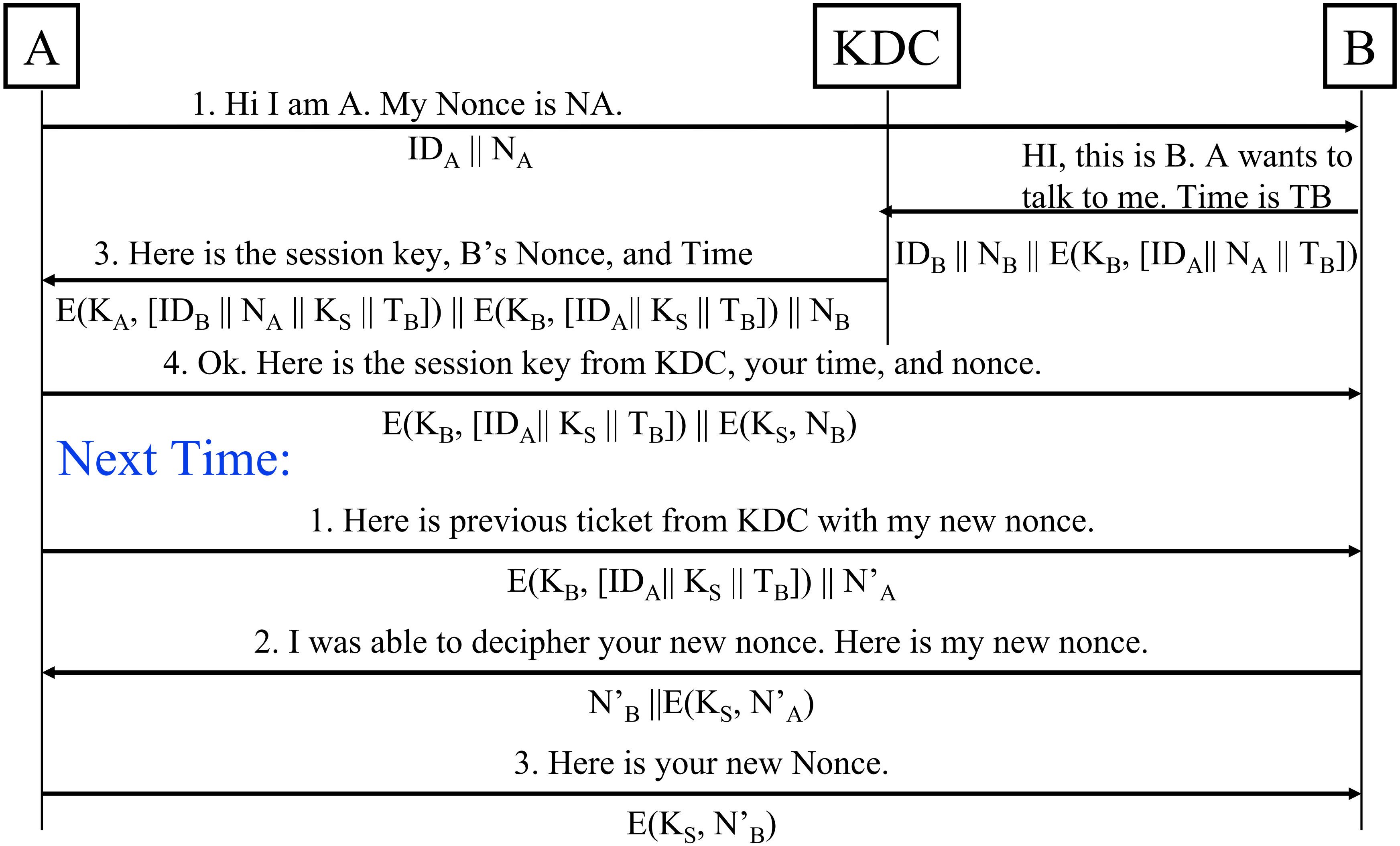
- Include timestamps



- Needs synchronized clocks.

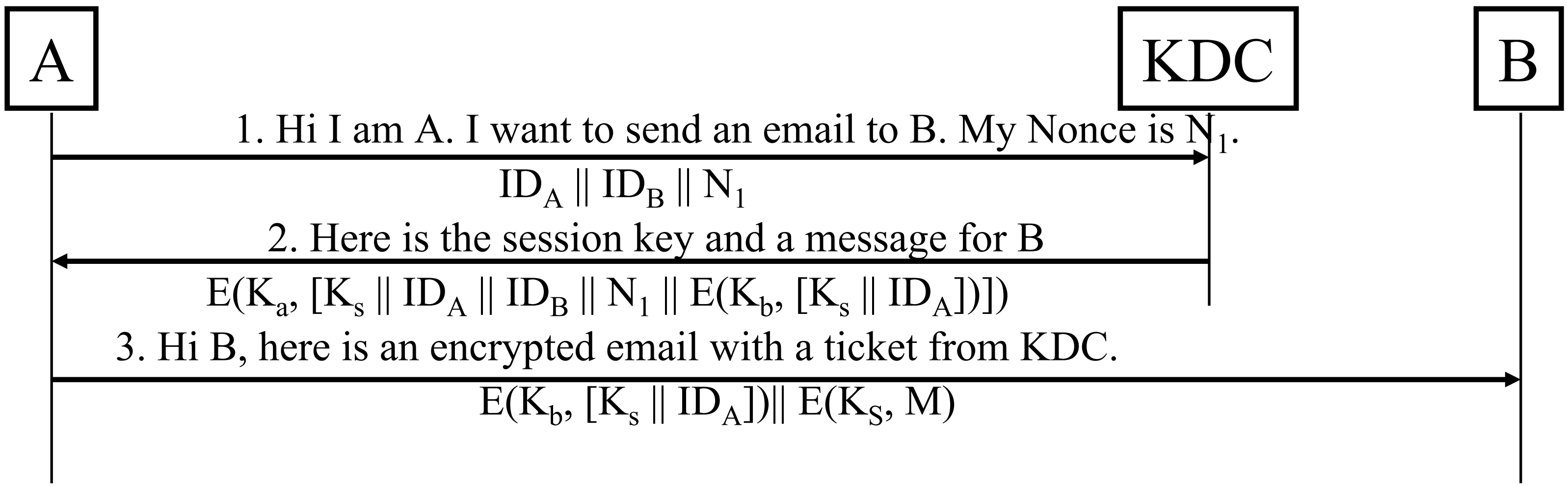
Corrected Protocol

- Include timestamps and nonce. Can set up next conn w/o KDC



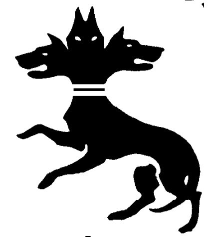
One-Way Authentication for Email

- B is not up, when A wants to send email



- Only B can read the message.
- Authenticates that A sent the message.

Overview of Kerberos



- Allows two users (or client and server) to authenticate each other over an insecure network
- Named after the Greek mythological character *Kerberos* (or *Cerberus*), known in Greek mythology as being the *monstrous three-headed guard dog of Hades*
- Designed originally for Project Athena at M.I.T.
- Implementation freely available from M.I.T.
- V5 is proposed as an Internet Standard (RFC 4120)
- Windows 2000/XP/Server 2003/Vista use Kerberos as their default authentication mechanism
- Apple's Mac OS X clients and servers also use Kerberos
- Apache HTTP Server, Eudora, NFS, OpenSSH, rcp (remote copy), rsh, X window system allow using Kerberos for authentication.

Overview (Cont)

- Protects against eavesdropping and replay attacks
- Uses a trusted third party (Authentication Server) and symmetric key cryptography
- First 3 versions are no longer in use.
- V5 is a generalization of V4 with several problems fixed and additional features.
- It is easier to understand V5 if you know V4
- Learn V4's features and mistakes

Sample Kerberos Exchange

Client

Authentication Server (AS)

Ticket Granting Server (TGS)

Service Server

1. Hi! Jain@cse would like to use the network today → (to AS)
2. Here is a day pass for jain@cse ← (from AS)
3. Hi! Jain@cse would like to communicate with PrintServer. Attached is his day pass. → (to TGS)
4. Here is the ticket for jain@cse to communicate with PrintServer. It includes a session key. ← (from TGS)
5. Hi jain@cse wants to communicate with you.. Here is his ticket. → (to Service Server)
6. Perfect. Let us use the session key that was in your ticket. ← (from Service Server)

Kerberos V4 Concepts

- **Authentication Server (AS):** Physically secure node with complete authentication database
- **Principal**: Authentication Server A, Ticket Granting Server G, Client (Computer) C, User (Human) U, Server V
- **Ticket Granting Server (TGS)**
- **Keys**: K_{cg}, K_{cv}, K_{ag}, K_{u}, K_{gv}
- **Ticket**: Encrypted information. All current V4 implementations use DES.
- **Ticket Granting Ticket (TGT):** Allows user to get tickets from TGS

Concepts (Cont)

- **Authenticator**: Name and time encrypted with a session key. Sent from client to server with the ticket and from server to client.
- **Credentials**: Session key + Ticket
- User enters a name and password. Client converts the password to a key K_u.
- TGT and the session key are good for a limited time (21 hours).

Key Design Principles

1. The network is open ⇒ Need a proper secret key to understand the messages received (except message 1, which is in clear)
2. Every client and server has a pre-shared secret with the AS.
3. AS and Ticket Granting Server (TGS) are logically separate but share a secret key
4. Both AS and TGS are stateless and do not need to remember the permissions granted. All the state is in the tickets. (Day pass is just a longer term ticket)
5. Longer term secrets are used less frequently. Short term secrets are created and destroyed after a limited use.

Kerberos V4 Message Exchange

Client

AS A

TGS G

Server V

1. $ID_A \parallel ID_G \parallel TS_1$ (Client → AS A)

2. (AS A → Client)
$E(K_C, [K_{CG} \parallel ID_G \parallel TS_2 \parallel Lifetime_2 \parallel Ticket_G])$
$Ticket_G = E(K_G, [K_{CG} \parallel ID_C \parallel AD_C \parallel ID_G \parallel TS_2 \parallel Lifetime_2])\}$
User name (ID_C) — *Network Address of Computer* (AD_C)

3. (Client → TGS G)
$ID_V \parallel Ticket_G \parallel Authenticator_C$
$Ticket_G = E(K_G, [K_{CG} \parallel ID_C \parallel AD_C \parallel ID_G \parallel TS_2 \parallel Lifetime_2])$
$Authenticator_C = E(K_{CG}, [ID_C \parallel AD_C \parallel TS_3])$

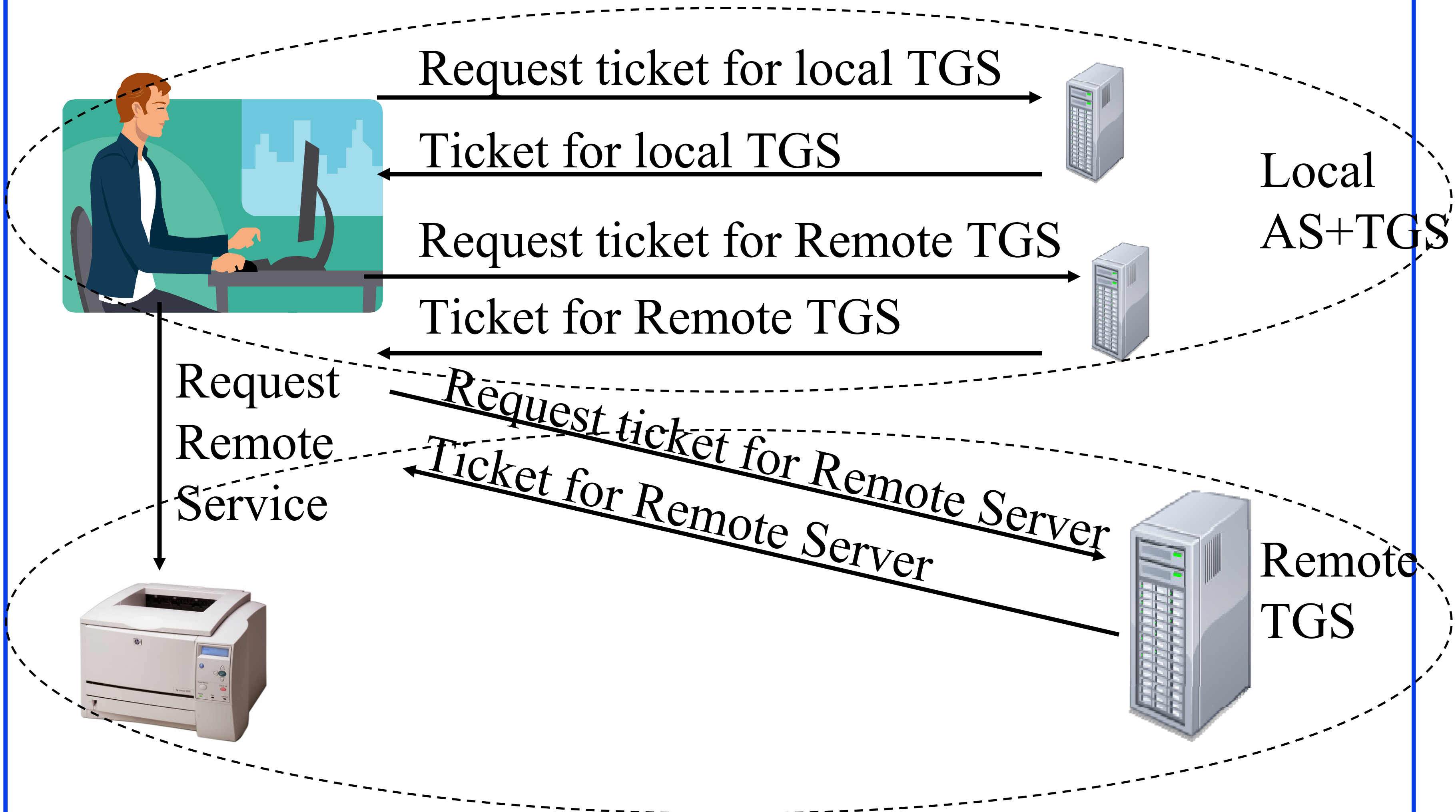
4. (TGS G → Client)
$E(K_{CG}, [K_{CV} \parallel ID_V \parallel TS_4 \parallel Ticket_V])$
$Ticket_V = E(K_V, [K_{CV} \parallel ID_C \parallel AD_C \parallel ID_V \parallel TS_4 \parallel Lifetime_4])$

5. (Client → Server V)
$Ticket_V \parallel Authenticator_C$
$Ticket_V = E(K_V, [K_{CV} \parallel ID_C \parallel AD_C \parallel ID_V \parallel TS_4 \parallel Lifetime_4])$
$Authenticator_C = E(K_{CV}, [ID_C \parallel AD_C \parallel TS_5])$

6. (Server V → Client)
$E(K_{CV}, [TS_5+1])$ (for mutual authentication

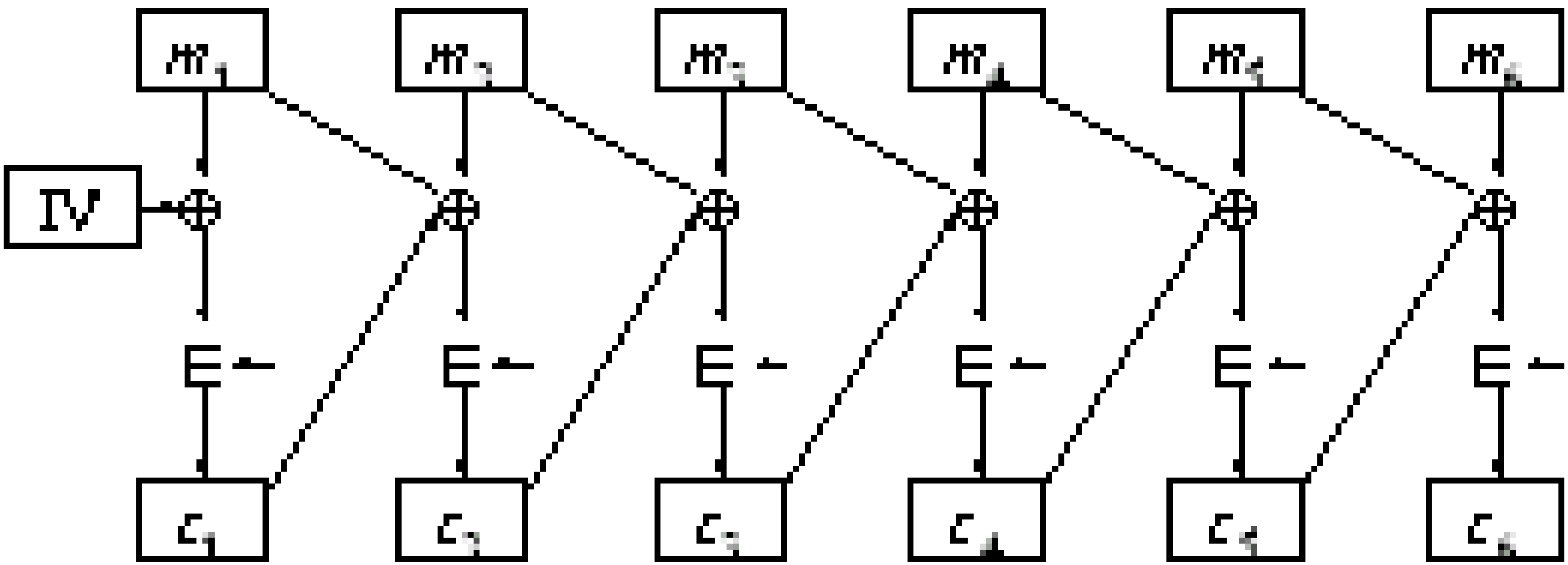
Inter-Realm Authentication

- Realm: One AS and its clients and servers



Privacy and Integrity

- Kerberos V4 uses and extension to CBC
- With CBC, only two blocks are affected by a change.
- **Plaintext Cipher Block Chaining** (PCBC) causes all blocks to change.



Kerberos V4 Issues

1. 1. Names, Instance, Realm (non standard). Limited to 40 Char.
2. 1. Only DES encryption. Not strong.
3. 2. Only IPv4 addresses. No IPv6 or ISO CLNP addresses.
4. 3. Byte ordering indicated in the message (ASN.1 better)
5. 4. Maximum life time limited to 21 hours: 8 bit life time in units of 5 minutes
6. 5. No delegation. A server cannot access another server on behalf of the client.
7. 6. Inter-realm authentication limited to pairs $\Rightarrow$ N^2 pairs
8. 7. Double encryption of the ticket: $K_{client}[K_{server}[...]]$
9. 8. Propagating Cipher Block Chaining (PCBC) does not detect interchange of cipher blocks
10. 9. No subsession keys for long sessions
11. 10. Brute force password attack

Kerberos V5

1. Names, Instance, Realm have ASN.1 names. Can be any length.
2. Any encryption. Encryption scheme coded.
3. Any type of addresses. Address type specified.
4. ASN.1 Byte ordering
5. Explicit Start time and End time. Can have arbitrary life times.
6. Delegation possible by requesting proxy able tickets.
7. Inter-realm authentication hierarchy
8. No Double encryption of the ticket
9. Explicit integrity mechanism detects block interchange
10. Subsession keys for long sessions
11. Password attack made difficult by a pre-authentication mechanism

Kerberos V5 Messages

Client

AS

TGS

Service Server

1. Options $\|$ ID_C $\|$ $Realm_C$ $\|$ ID_G $\|$ Times $\|$ $Nonce_1$

2. $Realm_C$ $\|$ ID_C $\|$ $Ticket_G$ $\|$ $E(K_C, [K_{CG}$ $\|$ Times $\|$ $Nonce_1$ $\|$ $Realm_G$ $\|$ $ID_G])$
   $Ticket_G = E(K_{AG}, [Flags \| K_{CG} \| Realm_C \| ID_C \| AD_C \| Times])$

3. Options $\|$ ID_V $\|$ Times $\|$ $Nonce_2$ $\|$ $Ticket_G$ $\|$ $Authenticator_C$

4. $Realm_C$ $\|$ ID_C $\|$ $Ticket_V$ $\|$ $E(K_{CG}$ $[K_{CV}$ $\|$Times $\|$$Nonce_2$ $\|$ $Realm_V$ $\|$ $ID_V])$
   $Ticket_G = E(K_G, [Flags \| K_{CG} \| Realm_C \| ID_C \| AD_C \| Times])$
   $Ticket_V = E(K_V, [Flags \| K_{CV} \| Realm_C \| ID_C \| AD_C \| Times])$
   $Authenticator_C = E(K_{CG}, [ID_C \| Realm_C \| TS_1])$

5. Options $\|$ $Ticket_V$ $\|$ $Authenticator_V$

6. $E(K_{CV}, [TS_2 \| SUbkey \| Seq \#])$
   $Ticket_V = E(K_V, [Flags \| K_{CV} \| Realm_C \| ID_C \| AD_C \| Times])$
   $Authenticaltr_C = E(K_{CV}, [ID_C \| Realm_C \| TS_2 \| Subkey \| Seq\#])$

Kerberos V5 Messages

Client C

AS

TGS G

Server V

Start
Expiration
Renew till

(1) Options || ID_C || $Realm_C$ || ID_G || Times || $Nonce_1$

(2) $Realm_C$ || ID_C || $Ticket_G$ || $E(K_C, [K_{CG}$ || Times || $Nonce_1$ || $Realm_G$ || $ID_G])$

$Ticket_G = E(K_G, [Flags || K_{CG} || Realm_C || ID_C || AD_C || Times])$

(3) Options || ID_V || Times || $Nonce_2$ || $Ticket_G$ || $Authenticator_C$

(4) $Realm_C$ || ID_C || $Ticket_V$ || $E(K_{CG} [K_{CV}$ ||Times ||$Nonce_2$ || $Realm_V$ || $ID_V])$

$Ticket_G = E(K_G, [Flags || K_{CG} || Realm_C || ID_C || AD_C || Times])$

$Ticket_V = E(K_V, [Flags || K_{CV} || Realm_C || ID_C || AD_C || Times])$

$Authenticator_C = E(K_{CG}, [ID_C || Realm_C || TS_1])$

(5) Options || $Ticket_V$ || $Authenticator_V$

(6) $E(K_{CV}, [TS_2 || Subkey || Seq\ \#])$

$Ticket_V = E(K_V, [Flags || K_{CV} || Realm_C || ID_C || AD_C || Times])$

$Authenticator_C = E(K_{CV}, [ID_C || Realm_C || TS_2 || Subkey || Seq\#])$

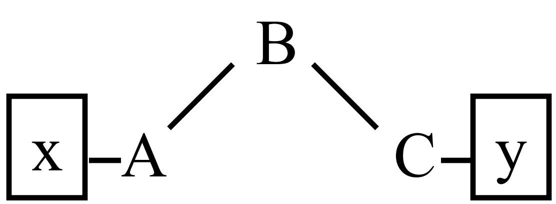
Session Key   Starting Seq #

Kerberos V5 Flags

- **Initial**: Ticket issued by AS (not by TGT)
- **Pre-Authent**: The client was pre-authenticated by AS before a ticket was issued
- **HW-Authent**: Pre-authenticated using hardware (e.g., smart card) possessed solely by name client
- **Renewable**: TGS can issue a new ticket that expires at a later date. Allows long life time.
- **May-Postdate**: TGS can issue a post-dated ticket
- **Postdated**: This ticket is postdated. Check authentication time field for original authentication time

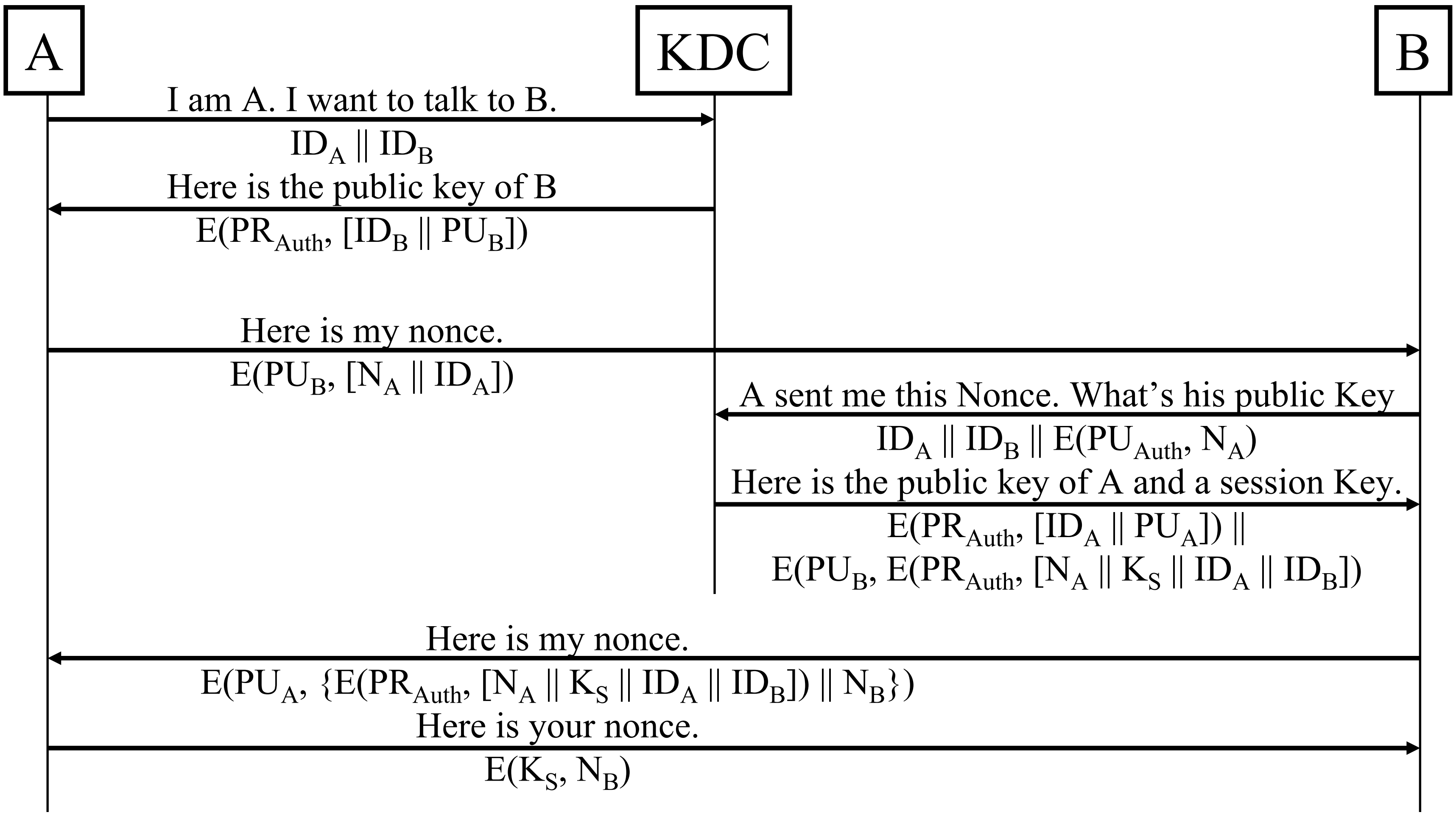
Kerberos V5 Flags (Cont)

- **Invalid**: This ticket is invalid and must be validated by TGS before use. Used with postdated tickets.
- **Proxiable**: TGS can issue a new service granting ticket with a different network address
- **Proxy**: Indicates that this ticket is a proxy
- **Forwardable**: TGS can issue a ticket with a different address for use in a different realm.
- **Forwarded**: This ticket has been forwarded or was issued based on a forwardable TGT. x@A can get to y@C via B. List of all transited realms is put in the ticket.



Remote User Authentication Using Public Keys

- KDC can be used to provide public keys for mutual authentication



Remote User Authentication Using Public Keys (Cont)

One-Way Authentication: Required for Email

- Can use public keys for encryption and authentication
- Long message $\Rightarrow$ Computation complexity
- For encryption, better to use a secret key and send the secret key using public key

$$A \rightarrow B: E(PU_B, K_S) || E(K_S, M)$$

- For authentication, use a digital signature

$$A \rightarrow B: M \,||\, E(PR_A, H(M))$$

Note: Someone else can replace the signature
$\Rightarrow$Encrypt the message and signature:

$$A \rightarrow B: E(PU_B, [M \,||\, E(PR_A, H(M))])$$

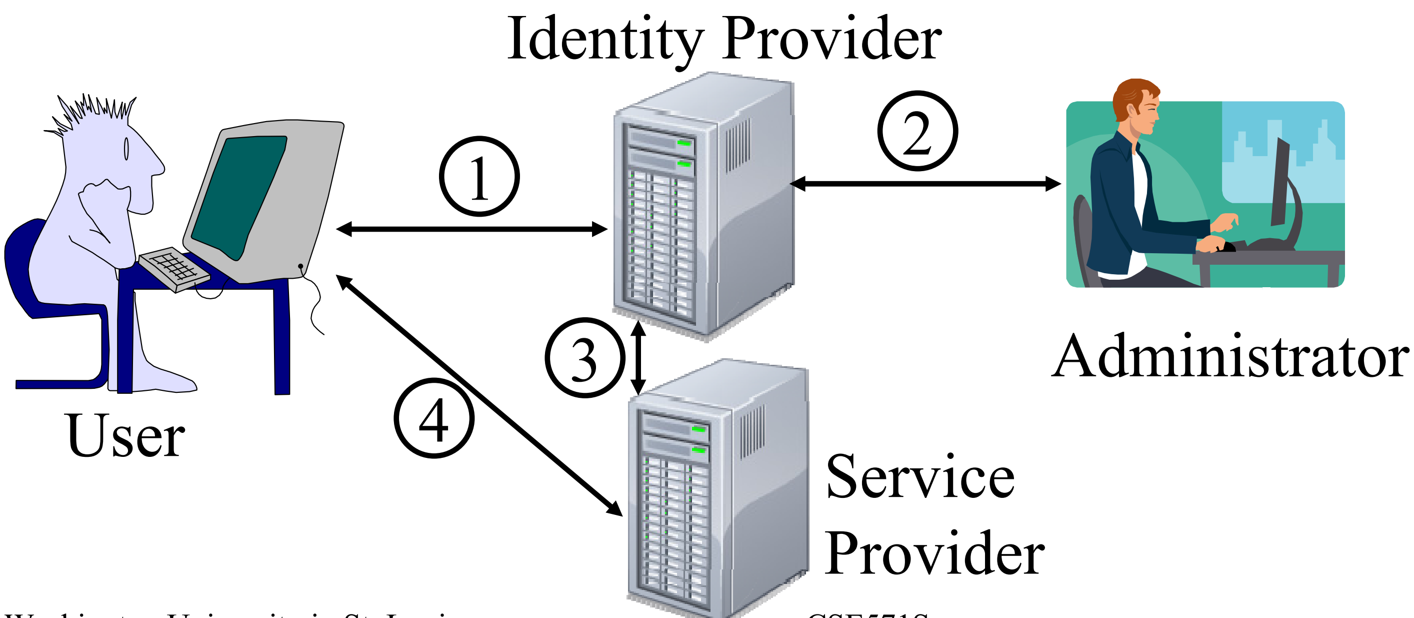
- Recipient B must know A's public key
$\Rightarrow$ A can sent its certificate with the message

Federated Identity Management

- Generalization of Single-Sign on
- User is authenticated once and then can use resources at other partner organizations across multiple security domains
- Examples:
  - Employees accessing purchasing sites
  - Health insurance providers
  - Purchasing sites to shipping sites
- Identity Management is more general than authentication
  - Authentication, authorization, accounting, provisioning, workflow automation, delegated administration, password synchronization, self-service password reset, federation
- Kerberos contains many of these elements

Federated Identity Operation

1. End user authenticates with the identity provider
2. Administrator associates attributes with each user or each role
3. Identity provider passes the id, attributes, and authentication to service provider
4. Service provider opens session with the user

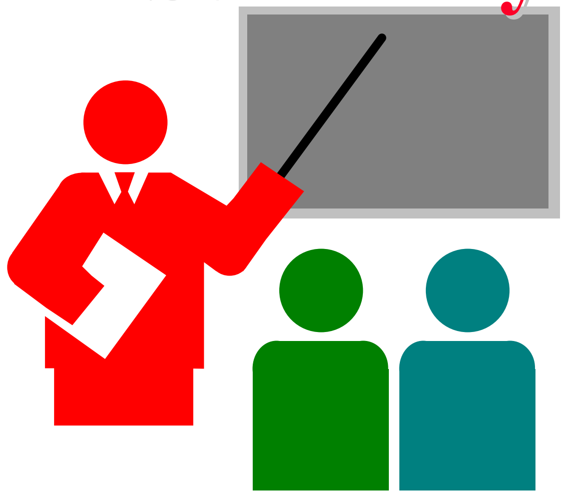


Standards for Federated ID Management

- Security Assertion Markup Language (SAML)
  - XML-based language for exchange of security information between online business partners
- Part of OASIS (Organization for the Advancement of Structured Information Standards) standards for federated identity management

Ref: http://en.wikipedia.org/wiki/SAML

Summary



- Kerberos is a symmetric key authentication system. Uses Authentication Server and ticket granting server.
- Kerberos V4 is widely deployed. V5 generalizes the design. Generalized ASN.1 names, General encryption, addresses, names. Allows delegation, post-dated tickets, renewals, Inter-realm authentication
- Federated identity management allows users to authenticate once and use resources on other partner organizations.
- Security Assertion Markup Language (SAML) is used to pass on security tokens for federated identity management.

Homework 15

- In Kerberos V4, when Bob receives a Ticket from Alice:
    a. How does he know that it is genuine?
    b. How does he know that it came from Alice?
    c. When Alice receives a reply, how does she know that it is not a replay of an earlier message from Bob?
    d. What does the Ticket contain that allows Alice and Bob to talk securely

Limit your answer to one sentence each.