Week 11

Lecture 1

User Authentication
Additional Material on Kerberos from the textbook

Lecture 2

Secure Socket Layer
Additional Material on TLS layer from the textbook

Workshop 11: Workshop based on Lectures in Week 9

Quiz 11

Kerberos and Federated Identity Management and Personal identity Verification From William Stallings

COMP90043

Additional Material

Udaya Parampalli

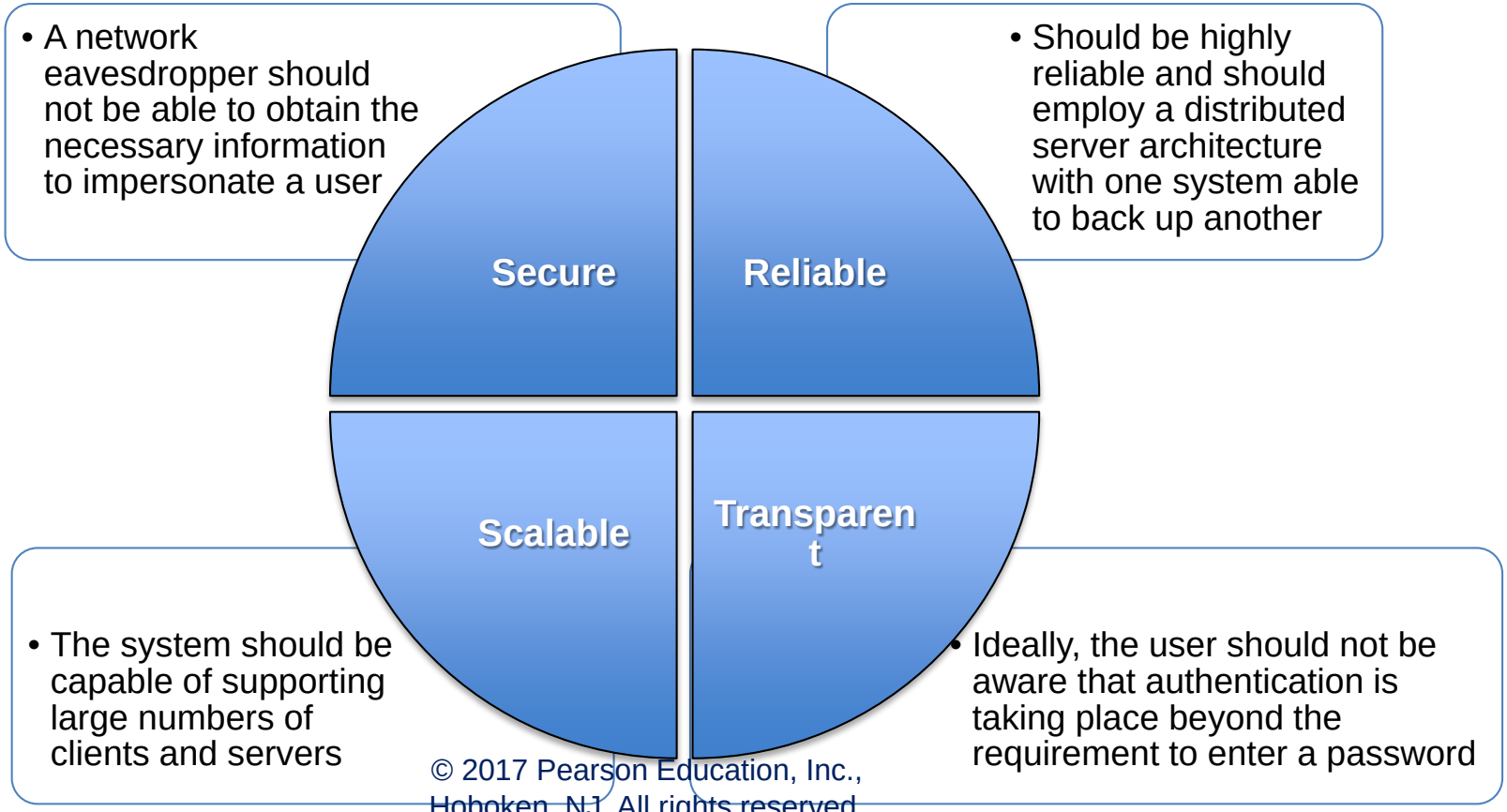
Kerberos

- Authentication service developed as part of Project Athena at MIT
- A workstation cannot be trusted to identify its users correctly to network services
  - A user may gain access to a particular workstation and pretend to be another user operating from that workstation
  - A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation
  - A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations

- Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users
  - Relies exclusively on symmetric encryption, making no use of public-key encryption

Kerberos Requirements

- The first published report on Kerberos listed the following requirements:



Kerberos Version 4

- Makes use of DES to provide the authentication service
- Authentication server (AS)
  - Knows the passwords of all users and stores these in a centralized database
  - Shares a unique secret key with each server
- Ticket
  - Created once the AS accepts the user as authentic; contains the user's ID and network address and the server's ID
  - Encrypted using the secret key shared by the AS and the server
- Ticket-granting server (TGS)
  - Issues tickets to users who have been authenticated to AS
  - Each time the user requires access to a new service the client applies to the TGS using the ticket to authenticate itself
  - The TGS then grants a ticket for the particular service
  - The client saves each service-granting ticket and uses it to authenticate its user to a server each time a particular service is requested

The Version 4 Authentication Dialogue

The lifetime associated with the ticket-granting ticket creates a problem:

- If the lifetime is very short (e.g., minutes), the user will be repeatedly asked for a password
- If the lifetime is long (e.g., hours), then an opponent has a greater opportunity for replay

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

Servers need to authenticate themselves to users

Table 15.1 Summary of Kerberos Version 4 Message Exchanges

(page 470 in textbook)

(1) C → AS $ID_c \| ID_{tgs} \| TS_1$

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

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

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

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

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

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$

$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$

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

(5) C → V $Ticket_v \| Authenticator_c$

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

$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$

$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$

(c) Client/Server Authentication Exchange to obtain service

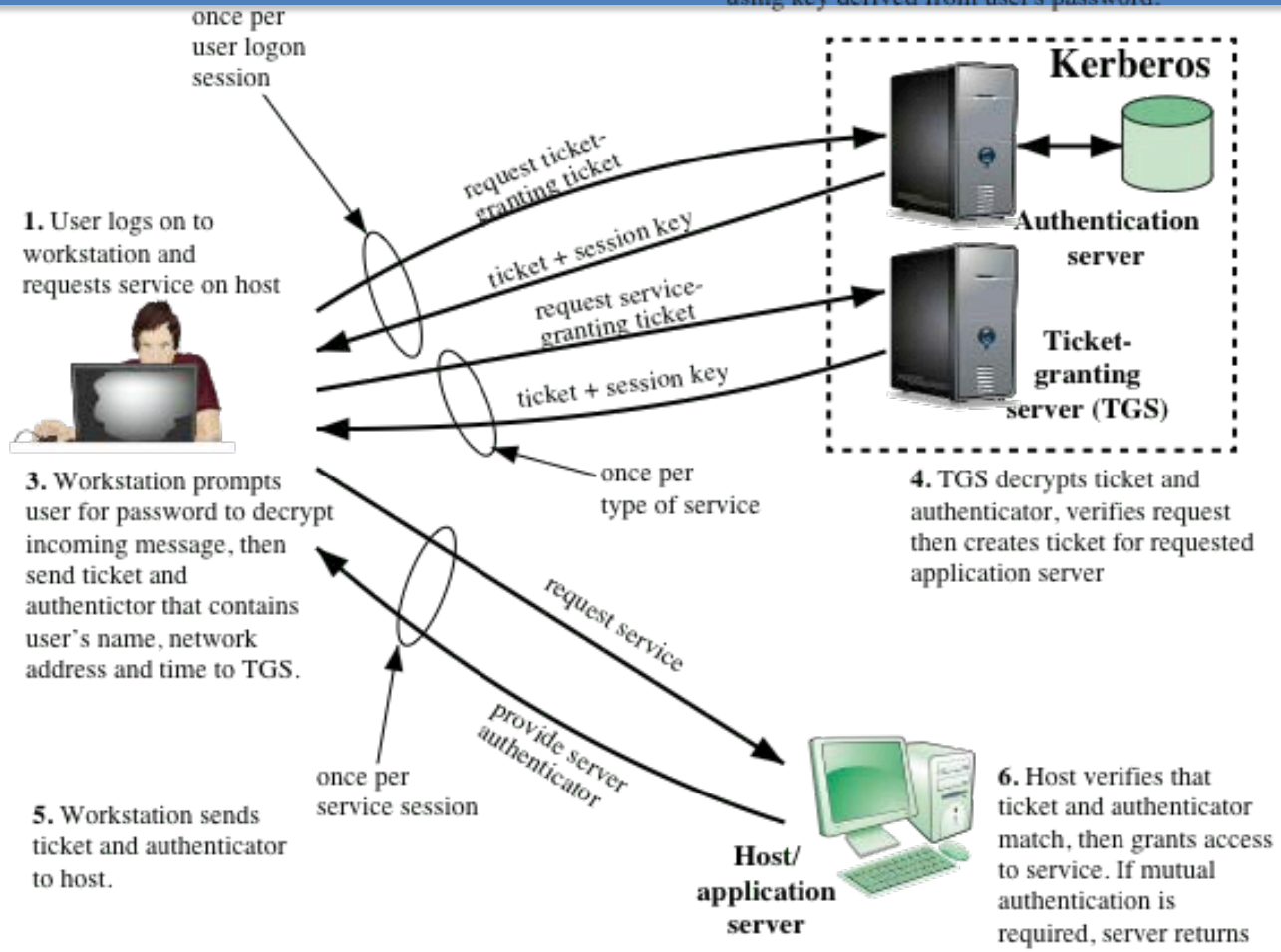


Figure 15.2 Overview of Kerberos

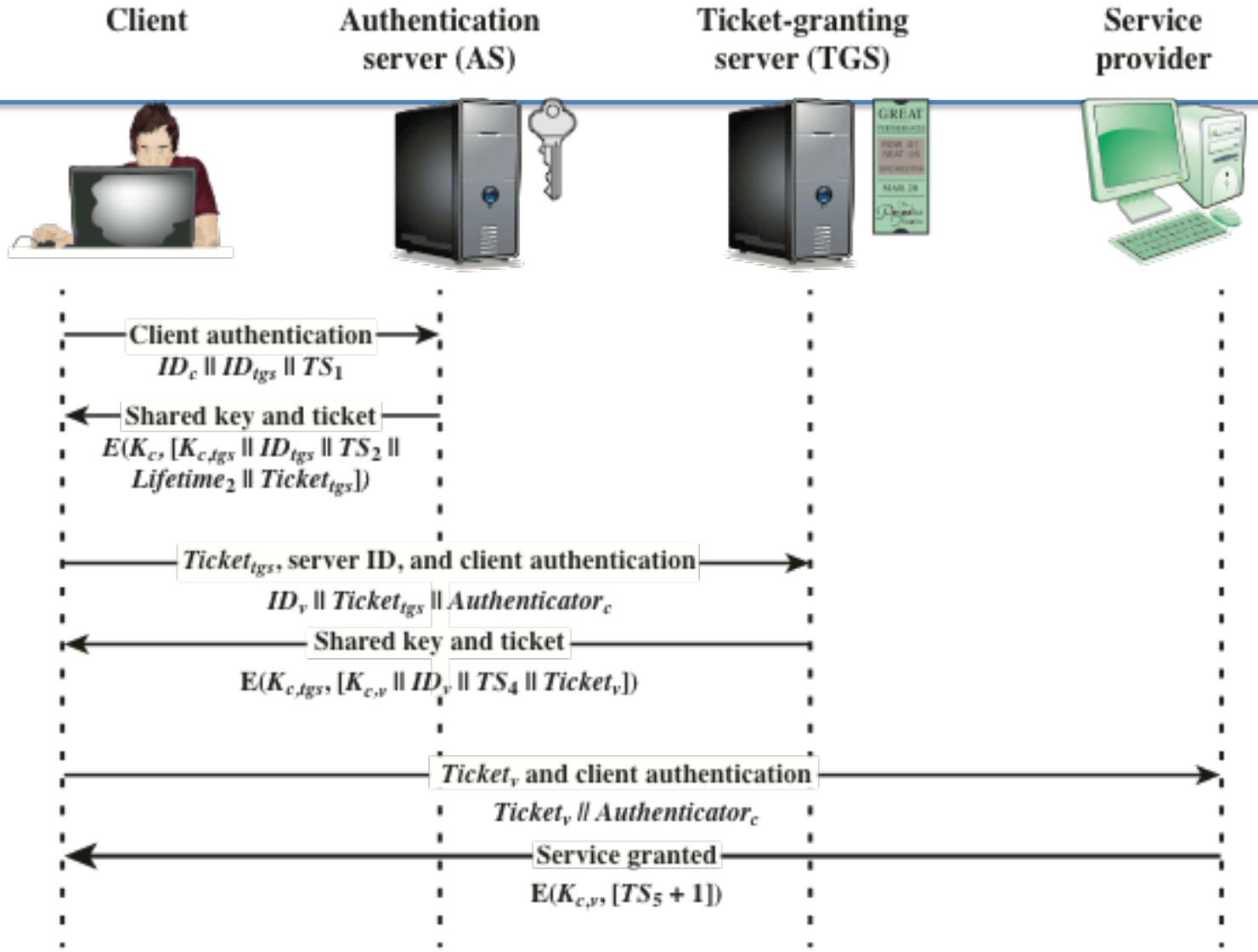


Figure 15.3 Kerberos Exchanges

Table 15.2 Rationale for the Elements of the Kerberos Version 4 Protocol

(page 1 of 3)

Message (1)	Client requests ticket-granting ticket.
ID_C	Tells AS identity of user from this client.
ID_{tgs}	Tells AS that user requests access to TGS.
TS_1	Allows AS to verify that client's clock is synchronized with that of AS.
Message (2)	AS returns ticket-granting ticket.
K_c	Encryption is based on user's password, enabling AS and client to verify password, and protecting contents of message (2).
$K_{c,tgs}$	Copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key.
ID_{tgs}	Confirms that this ticket is for the TGS.
TS_2	Informs client of time this ticket was issued.
$Lifetime_2$	Informs client of the lifetime of this ticket.
$Ticket_{tgs}$	Ticket to be used by client to access TGS.

(This table can be found on pages 473 – 474 in the textbook)

Message (3)	Client requests service-granting ticket.
ID_V	Tells TGS that user requests access to server V.
$Ticket_{tgs}$	Assures TGS that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket .
Message (4)	TGS returns service-granting ticket.
$K_{c,tgs}$	Key shared only by C and TGS protects contents of message (4).
$K_{c,v}$	Copy of session key accessible to client created by TGS to permit secure exchange between client and server without requiring them to share a permanent key.
ID_V	Confirms that this ticket is for server V.
TS_4	Informs client of time this ticket was issued.
$Ticket_V$	Ticket to be used by client to access server V.
$Ticket_{tgs}$	Reusable so that user does not have to reenter password.
K_{tgs}	Ticket is encrypted with key known only to AS and TGS, to prevent Tampering.
$K_{c,tgs}$	Copy of session key accessible to TGS used to decrypt authenticator, thereby authenticating ticket.
ID_C	Indicates the rightful owner of this ticket.
AD_C	Prevents use of ticket from workstation other than one that initially requested the ticket.
ID_{tgs}	Assures server that it has decrypted ticket properly.
TS_2	Informs TGS of time this ticket was issued.
$Lifetime_2$	Prevents replay after ticket has expired.
$Authenticator_c$	Assures TGS that the ticket presenter is the same as the client for whom the ticket was issued has very short lifetime to prevent replay.
$K_{c,tgs}$	Authenticator is encrypted with key known only to client and TGS, to prevent tampering.
ID_C	Must match ID in ticket to authenticate ticket.
AD_C	Must match address in ticket to authenticate ticket.
TS_3	Informs TGS of time this authenticator was generated.

Message (5)	Client requests service.
$Ticket_V$	Assures server that this user has been authenticated by AS.
$Authenticator_c$	Generated by client to validate ticket.
Message (6)	Optional authentication of server to client.
$K_{c,v}$	Assures C that this message is from V.
$TS_5 + 1$	Assures C that this is not a replay of an old reply.
$Ticket_v$	Reusable so that client does not need to request a new ticket from TGS for each access to the same server.
K_v	Ticket is encrypted with key known only to TGS and server, to prevent Tampering.
$K_{c,v}$	Copy of session key accessible to client; used to decrypt authenticator, thereby authenticating ticket.
ID_C	Indicates the rightful owner of this ticket.
AD_C	Prevents use of ticket from workstation other than one that initially requested the ticket.
ID_V	Assures server that it has decrypted ticket properly.
TS_4	Informs server of time this ticket was issued.
$Lifetime_4$	Prevents replay after ticket has expired.
$Authenticator_c$	Assures server that the ticket presenter is the same as the client for whom the ticket was issued; has very short lifetime to prevent replay.
$K_{c,v}$	Authenticator is encrypted with key known only to client and server, to prevent tampering.
ID_C	Must match ID in ticket to authenticate ticket.
AD_c	Must match address in ticket to authenticate ticket.
TS_5	Informs server of time this authenticator was generated.

Kerberos Realms and Multiple Kerberi

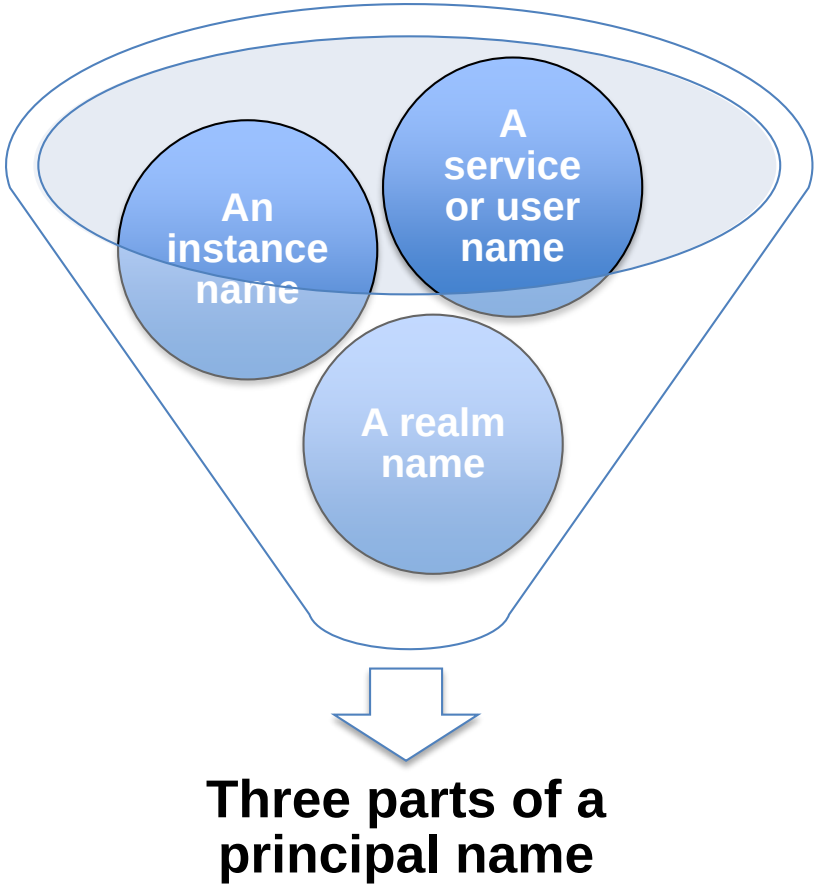
- A full-service Kerberos environment consisting of a Kerberos server, a number of clients, and a number of application servers requires that:
  - The Kerberos server must have the user ID and hashed passwords of all participating users in its database; all users are registered with the Kerberos server
  - 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

Kerberos Realm

- A set of managed nodes that share the same Kerberos database
- The database resides on the Kerberos master computer system, which should be kept in a physically secure room
- A read-only copy of the Kerberos database might also reside on other Kerberos computer systems
- All changes to the database must be made on the master computer system
- Changing or accessing the contents of a Kerberos database requires the Kerberos master password

Kerberos Principal

- A service or user that is known to the Kerberos system
- Identified by its principal name



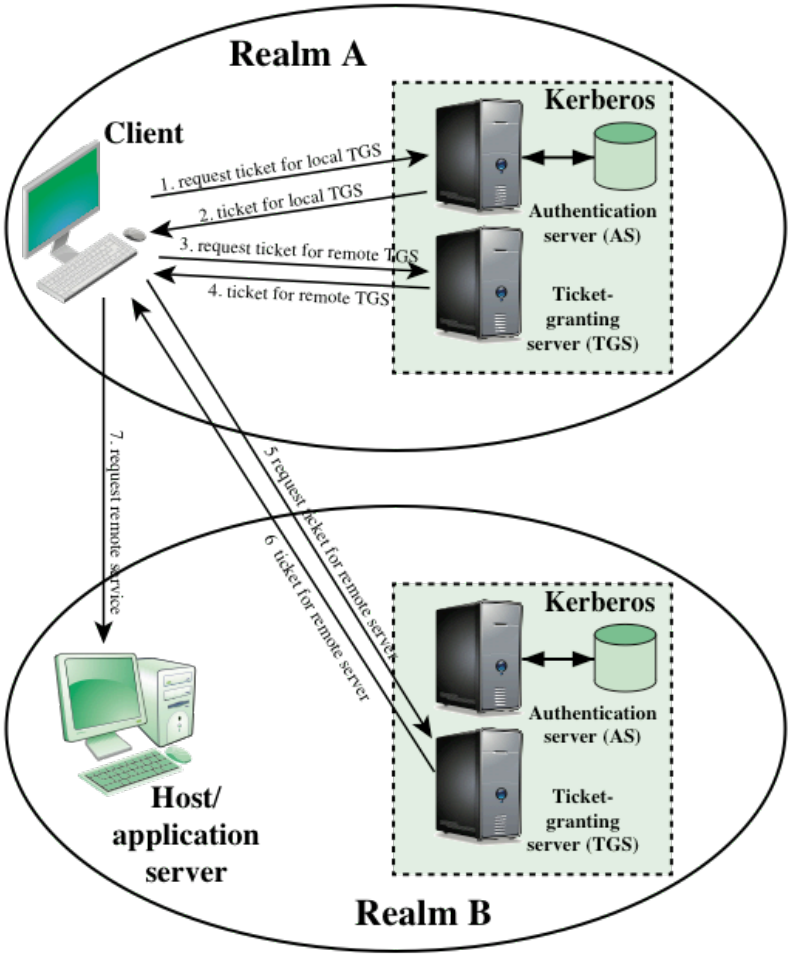


Figure 15.4 Request for Service in Another Realm

Differences Between Versions 4 and 5

Version 5 is intended to address the limitations of version 4 in two areas:

Environmental shortcomings

- Encryption system dependence
- Internet protocol dependence
- Message byte ordering
- Ticket lifetime
- Authentication forwarding
- Interrealm authentication

Technical deficiencies

- Double encryption
- PCBC encryption
- Session keys
- Password attacks

Table 15.3 (page 479 in textbook)

Summary of Kerberos Version 5 Message Exchanges

(1) C → AS $Options \| IDc \| Realmc \| IDtgs \| Times \| Nonce1$

(2) AS → C $Realmc \| IDC \| Tickettgs \| E(Kc, [Kc,tgs \| Times \| Nonce1 \| Realmtgs \| IDtgs])$

$Tickettgs = E(Ktgs, [Flags \| Kc,tgs \| Realmc \| IDC \| ADC \| Times])$

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

(3) C → TGS $Options \| IDv \| Times \| \| Nonce2 \| Tickettgs \| Authenticatorc$

(4) TGS → C $Realmc \| IDC \| Ticketv \| E(Kc,tgs, [Kc,v \| Times \| Nonce2 \| Realmv \| IDv])$

$Tickettgs = E(Ktgs, [Flags \| Kc,tgs \| Realmc \| IDC \| ADC \| Times])$

$Ticketv = E(Kv, [Flags \| Kc,v \| Realmc \| IDC \| ADC \| Times])$

$Authenticatorc = E(Kc,tgs, [IDC \| Realmc \| TS1])$

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

(5) C → V $Options \| Ticket_v \| Authenticator_c$

(6) V → C $E_{K_{C,V}} [TS_2 \| Subkey \| Seq\#]$

$Ticketv = E(Kv, [Flags \| Kc,v \| Realmc \| IDC \| ADC \| Times])$

$Authenticatorc = E(Kc,v, [IDC \| Realmc \| TS2 \| Subkey \| Seq\#])$

(c) Client/Server Authentication Exchange to obtain service

Table 15.4

Kerberos Version 5 Flags

(Table can be found on page 480 in textbook)

Mutual Authentication

- Public-key encryption for session key distribution
  - Assumes each of the two parties is in possession of the current public key of the other
  - May not be practical to require this assumption
- Denning protocol using timestamps
  - Uses an authentication server (AS) to provide public-key certificates
  - Requires the synchronization of clocks
- Woo and Lam makes use of nonces
  - Care needed to ensure no protocol flaws

One-Way Authentication

- Have public-key approaches for e-mail
  - Encryption of message for confidentiality, authentication, or both
  - The public-key algorithm must be applied once or twice to what may be a long message
- For confidentiality encrypt message with one-time secret key, public-key encrypted
- If authentication is the primary concern, a digital signature may suffice

Federated Identity Management

- Relatively new concept dealing with the use of a common identity management scheme across multiple enterprise and numerous applications and supporting many users
- Services provided include:
  - Point of contact
  - SSO protocol services
  - Trust services
  - Key services
  - Identity services
  - Authorization
  - Provisioning
  - Management

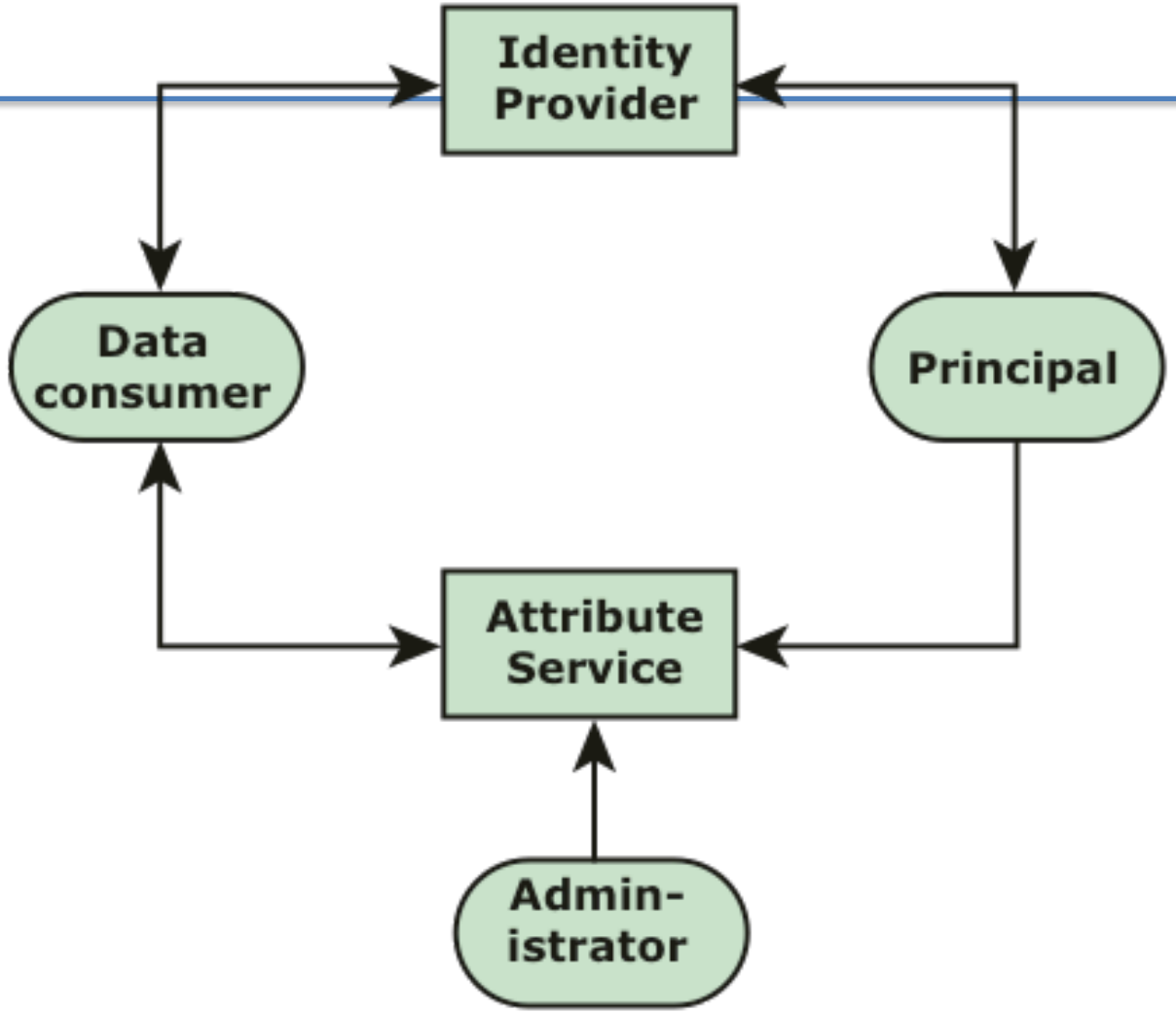
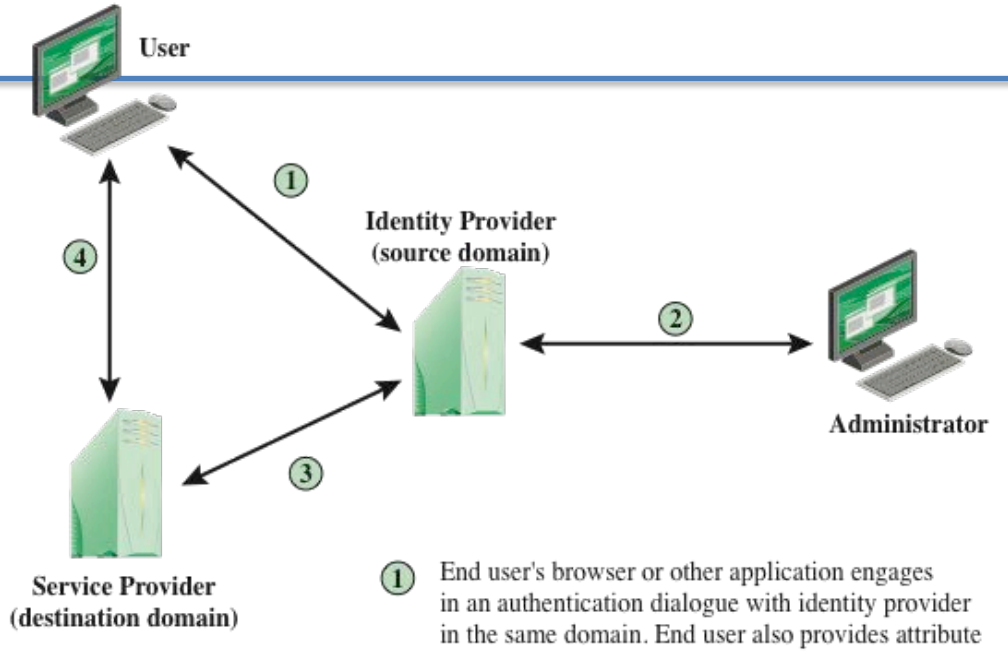


Figure 15.5 Generic Identity Management System



1. End user's browser or other application engages in an authentication dialogue with identity provider in the same domain. End user also provides attribute values associated with user's identity.
2. Some attributes associated with an identity, such as allowable roles, may be provided by an administrator in the same domain.
3. A service provider in a remote domain, which the user wishes to access, obtains identity information, authentication information, and associated attributes from the identity provider in the source domain.
4. Service provider opens session with remote user and enforces access control restrictions based on user's identity and attributes.

Figure 15.6 Federated Identity Operation

Key Standards

The Extensible Markup Language (XML)	The Simple Object Access Protocol (SOAP)	WS-Security	Security Assertion Markup Language (SAML)
A markup language that uses sets of embedded tags or labels to characterize text elements within a document so as to indicate their appearance, function, meaning, or context	Enables applications to request services from one another with XML-based requests and receive responses as data formatted with XML	A set of SOAP extensions for implementing message integrity and confidentiality in Web services	An XML-based language for the exchange of security information between online business partners

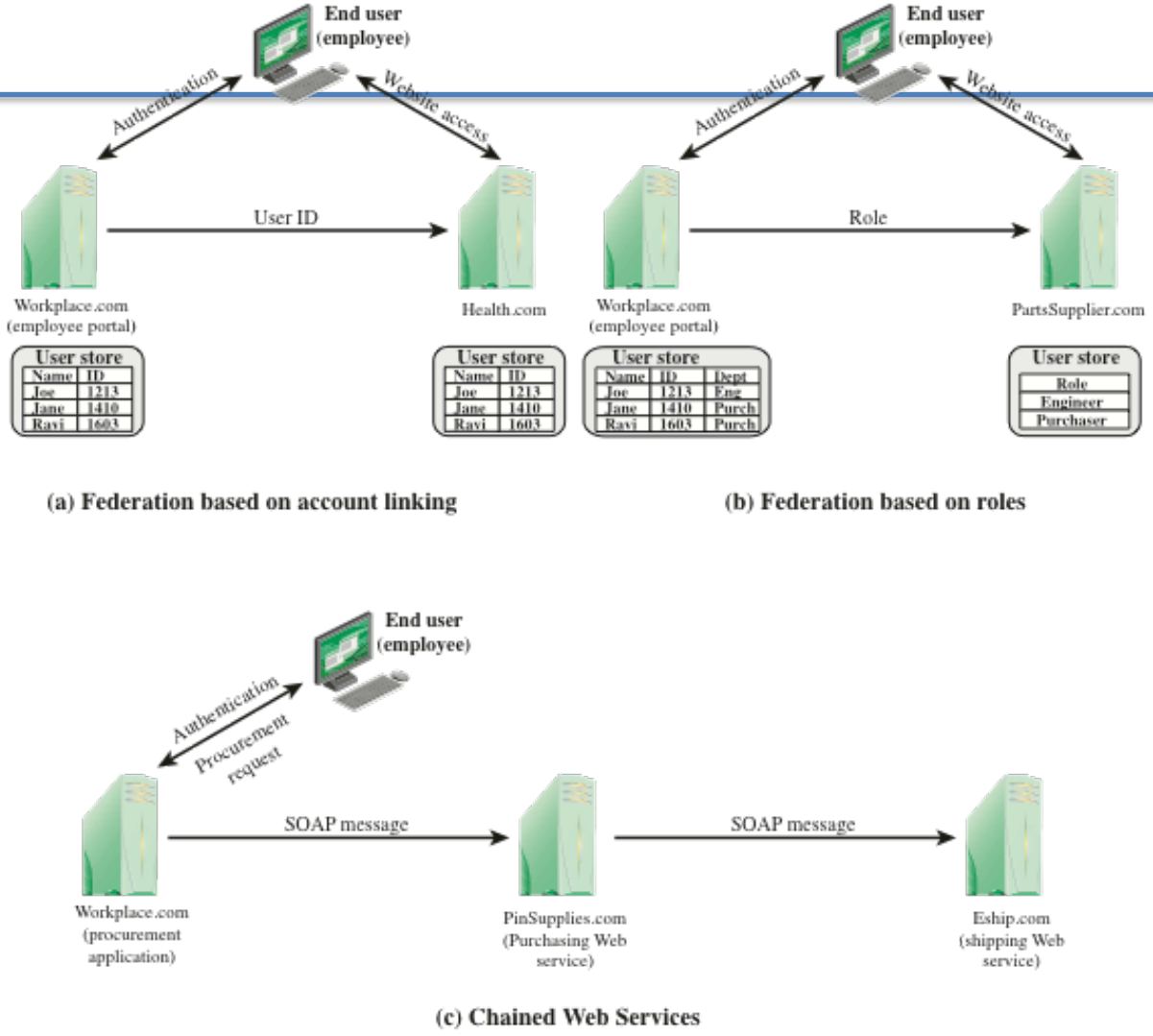


Figure 15.7 Federated Identity Scenarios

Personal Identity Verification

- User authentication based on the possession of a smart card is becoming more widespread
  - Has the appearance of a credit card
  - Has an electronic interface
  - May use a variety of authentication protocols
- A smart card contains within it an entire microprocessor, including processor, memory, and I/O ports
- A smart card includes three types of memory:
  - Read-only memory (ROM) stores data that does not change during the card's life
  - Electronically erasable programmable ROM (EEPROM) holds application data and programs; also holds data that may vary with time
  - Random access memory (RAM) holds temporary data generated when applications are executed

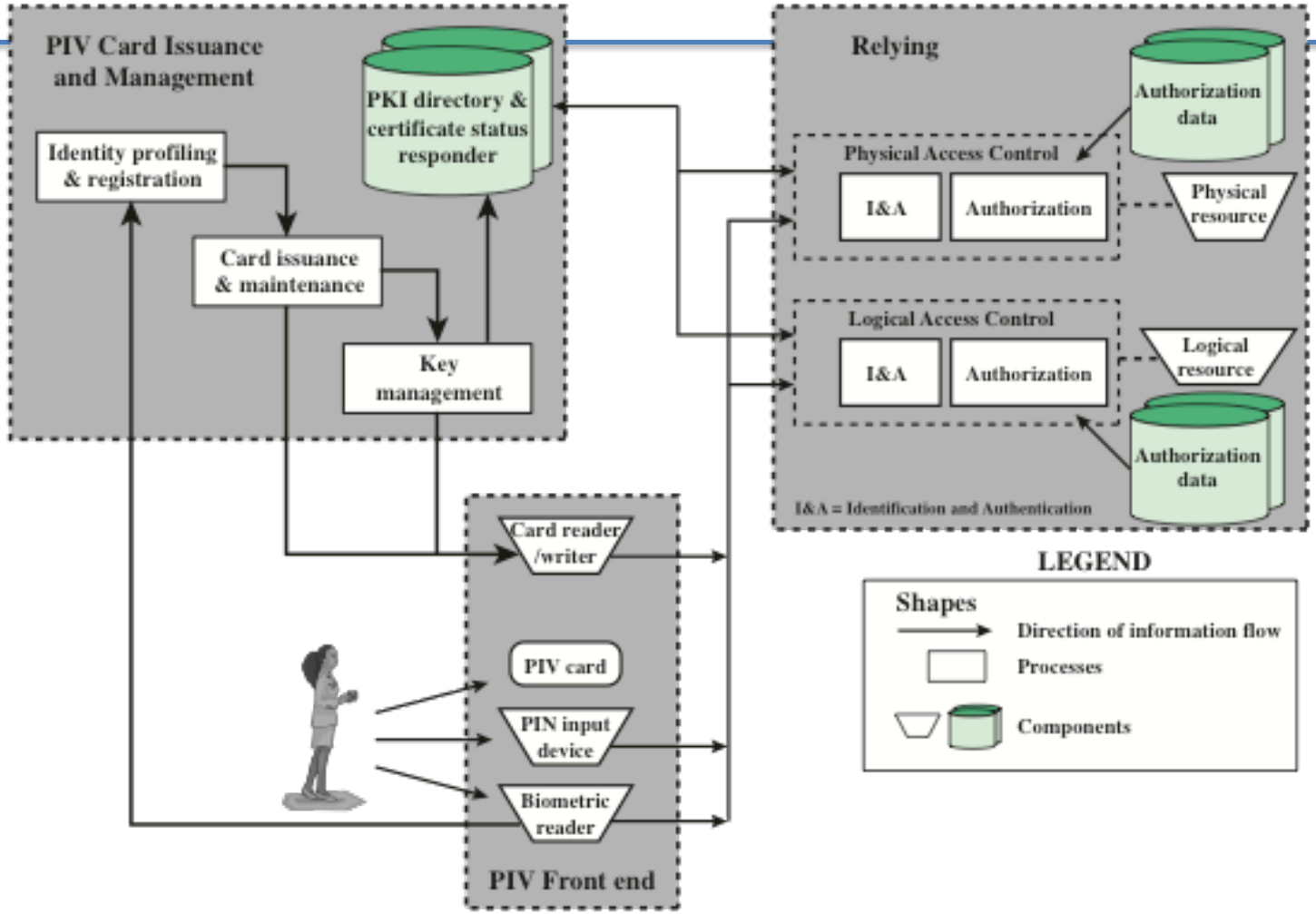


Figure 15.8 FIPS 201 PIV System Model

PIV Documentation

- **FIPS 201-2—Personal Identity Verification (PIV) of Federal Employees and Contractors**
  - Specifies the physical card characteristics, storage media, and data elements that make up the identity credentials resident on the PIV card
- **SP 800-73-3—Interfaces for Personal Identity Verification**
  - Specifies the interfaces and card architecture for storing and retrieving identity credentials from a smart card, and provides guidelines for the use of authentication mechanisms and protocols
- **SP 800-76-2—Biometric Data Specification for Personal Identity Verification**
  - Describes technical acquisition and formatting specifications for the biometric credentials of the PIV system
- **SP 800-78-3—Cryptographic Algorithms and Key Sizes for Personal Identity Verification**
  - Identifies acceptable symmetric and asymmetric encryption algorithms, digital signature algorithms, and message digest algorithms, and specifies mechanisms to identify the algorithms associated with PIV keys or digital signatures
- **SP 800-104—A Scheme for PIV Visual Card Topography**
  - Provides additional recommendations on the PIV card color-coding for designating employee affiliation
- **SP 800-116—A Recommendation for the Use of PIV Credentials in Physical Access Control Systems (PACS)**
  - Describes a risk-based approach for selecting appropriate PIV authentication mechanisms to manage physical access to Federal government facilities and assets
- **SP 800-79-1—Guidelines for the Accreditation of Personal Identity Verification Card Issuers**
  - Provides guidelines for accrediting the reliability of issuers of PIV cards that collect, store, and disseminate personal identity credentials and issue smart cards
- **SP 800-96—PIV Card to Reader Interoperability Guidelines**
  - Provides requirements that facilitate interoperability between any card and any reader

PIV Credentials and Keys

- **Personal Identification Number (PIN)**
  - Required to activate the card for privileged operation
- **Cardholder Unique Identifier (CHUID)**
  - Includes the Federal Agency Smart Credential Number (FASC-N) and the Global Unique Identification Number (GUID), which uniquely identify the card and the cardholder
- **PIV Authentication Key**
  - Asymmetric key pair and corresponding certificate for user authentication
- **Two fingerprint templates**
  - For biometric authentication
- **Electronic facial image**
  - For biometric authentication
- **Asymmetric Card Authentication Key**
  - Asymmetric key pair and corresponding certificate used for card authentication

Optional elements include the following:

- **Digital Signature Key**
  - Asymmetric key pair and corresponding certificate that supports document signing and signing of data elements such as the CHUID
- **Key Management Key**
  - Asymmetric key pair and corresponding certificate supporting key establishment and transport
- **Symmetric Card Authentication Key**
  - For supporting physical access applications
- **PIV Card Application Administration Key**
  - Symmetric key associated with the card management system
- **One or two iris images**
  - For biometric authentication

Authentication

- **Using the electronic credentials resident on a PIV card, the card supports the following authentication mechanisms:**
  - **CHUID**

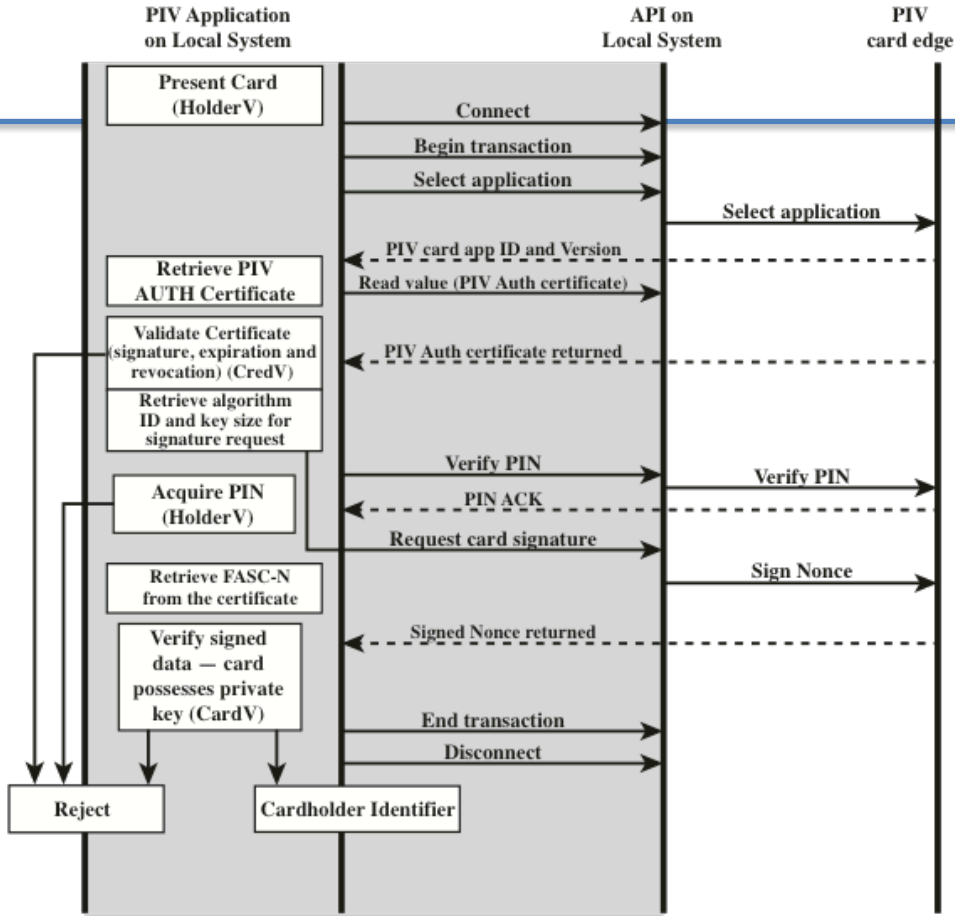
    The cardholder is authenticated using the signed CHUID data element on the card. The PIN is not required. This mechanism is useful in environments where a low level of assurance is acceptable and rapid contactless authentication is necessary
  - **Card Authentication Key**

    The PIV card is authenticated using the Card Authentication Key in a challenge response protocol. The PIN is not required. This mechanism allows contact (via card reader) or contactless (via radio waves) authentication of the PIV card without the holder's active participation, and provides a low level of assurance
- **BIO**

  The cardholder is authenticated by matching his or her fingerprint sample(s) to the signed biometric data element in an environment without a human attendant in view. The PIN is required to activate the card. This mechanism achieves a high level of assurance and requires the cardholder's active participation is submitting the PIN as well as the biometric sample
  - **BIO-A**

    The cardholder is authenticated by matching his or her fingerprint sample(s) to the signed biometric data element in an environment with a human attendant in view. The PIN is required to activate the card. This mechanism achieves a very high level of assurance when coupled with full trust validation of the biometric template retrieved from the card, and requires the cardholder's active participation is submitting the PIN as well as the biometric sample
- **PKI**

  The cardholder is authenticated by demonstrating control of the PIV authentication private key in a challenge response protocol that can be validated using the PIV authentication certificate. The PIN is required to activate the card. This mechanism achieves a very high level of identity assurance and requires the cardholder's knowledge of the PIN



CardV = Card validation
CredV = Credential validation
HolderV = Cardholder validation
FASC-N = Federal Agency Smart Credential Number

Figure 15.9 Authentication using PIV Authentication Key

Week 11

Lecture 1

User Authentication
Additional Material on Kerberos from the textbook

Lecture 2

Secure Socket Layer
Additional Material on TLS layer from the textbook

Workshop 11: Workshop based on Lectures in Week 9

Quiz 11