Authentication, Authorization, Accounting (AAA)

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

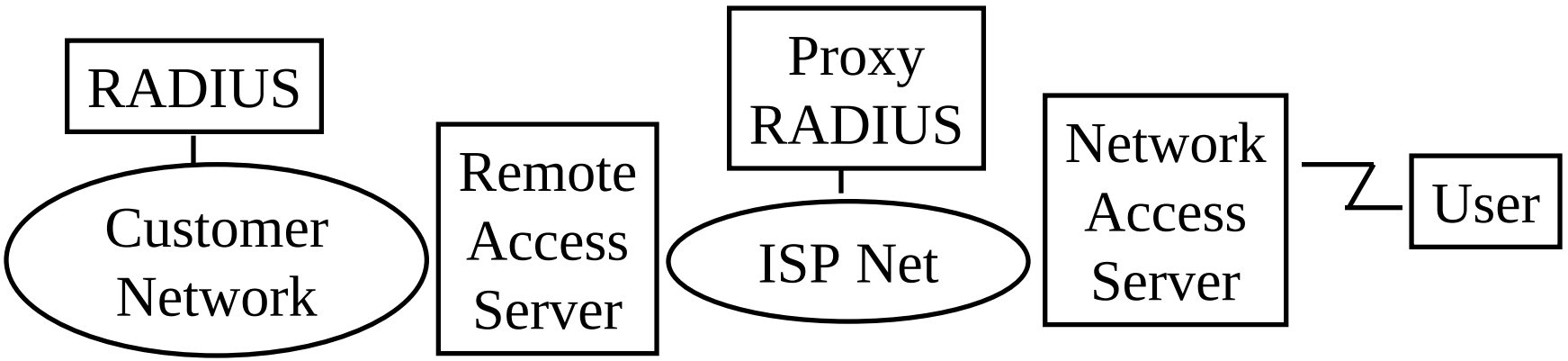
Audio/Video recordings of this lecture are available at:

http://www.cse.wustl.edu/~jain/cse571-07/

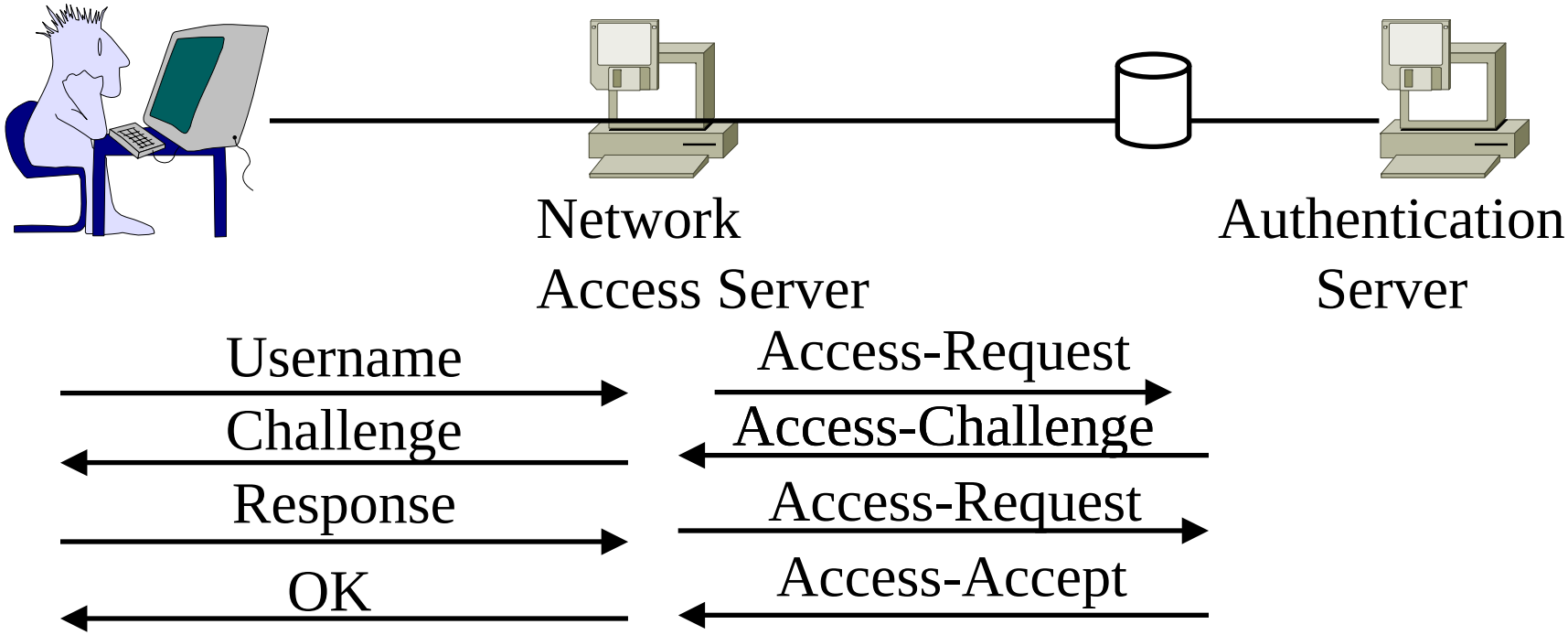
- RADIUS
- Authentication Protocols: PAP, CHAP, MS-CHAP
- Extensible Authentication Protocol (EAP)
- EAP Upper Layer Protocols
- 802.1X

RADIUS

- Remote Authentication Dial-In User Service
- Central point for Authorization, Accounting, and Auditing data ⇒ AAA server
- Network Access servers get authentication info from RADIUS servers
- Allows RADIUS Proxy Servers ⇒ ISP roaming alliances
- Normally runs on UDP ⇒ Can loose accounting packets
- FreeRADIUS and OpenRADIUS implementations available



RADIUS Messages



- Four Core Messages: Request, Challenge, Accept, Reject.
- Message Format: Code is the message type.
  Identifier is used to match request/response.

Code	Identifier	Length	Authenticator	Attributes

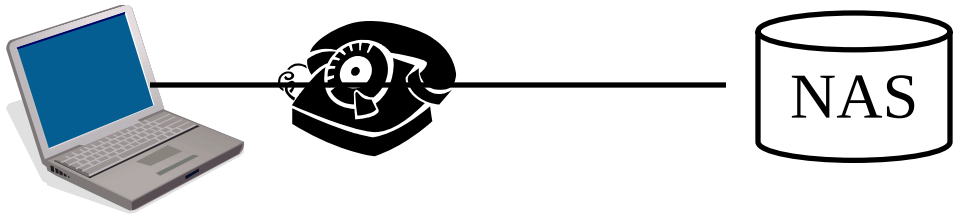
PAP and CHAP

- Point-to-point protocol (PPP) allows two authentication methods:
  - Password authentication protocol (PAP)
  - Challenge Handshake Authentication Protocol (CHAP) – RFC1994

DIAMETER

- Enhanced RADIUS
- Light weight
- Can use both UDP and TCP
- Servers can send unsolicited messages to Clients
  $\Rightarrow$ Increases the set of applications
- Support for vendor specific Attribute-Value-Pairs (AVPs) and commands
- Authentication and privacy for policy messages

Password Authentication Protocol (PAP)



- RFC 1334, Oct 1992
- Authenticator sends a authentication request
- Peer responds with a username and password in plain text
- Authenticator sends a success or failure
- Code: 1=Auth Request, 2=Auth Ack, 3=Auth Nak

Code	ID	Len	Name Len	Name Val	Pswd Len	Pswd Val
1B	1B	2B	1B	Var	1B	Var

Code	ID	Len	Success/Failure Message
1B	1B	2B	1B

CHAP

- Challenge Handshake Authentication Protocol
- RFC 1994, August 1996
- Uses a shared secret (password)
- Authenticator sends a challenge
- Peer responds with a MD5 checksum hash of the challenge
- Authenticator also calculates the hash and sends success or failure
- Requires both ends to know the password in plain text
- Replay attack prevention ⇒ Use a different challenge every time
- LCP option 3 = 0x05 ⇒ CHAP

CHAP (Cont)

Code	ID	Len	Chal. Len	Chal. Val	Name Len	Name Val
1B	1B	2B	1B	Var	1B	Var

Code	ID	Len	Resp. Len	Resp. Val	Name Len	Name Val
1B	1B	2B	1B	Var	1B	Var

Code	ID	Len	Success/Failure Message
1B	1B	2B	1B

- Codes: 1=Challenge, 2=Response, 3=Success, 4=Failure

MS-CHAP

- Microsoft version of CHAP
- MS-CHAP in RFC 2433, Oct 1998
- Does not require password in plain text
- Uses hash of the password
- LCP option 3 = 0x80 ⇒ MS-CHAPv1
- 8B challenge ⇒ 24B LM compatible response, 24B NTLM compatible response and 1B use NTLM flag
- LM passwords are limited to 14 case-insensitive OEM characters
- NT passwords are 0 to 256 case-sensitive Unicode characters
- Flag ⇒ NT response is meaningful and should be used
- Also allows users to change password

MS-CHAPv2

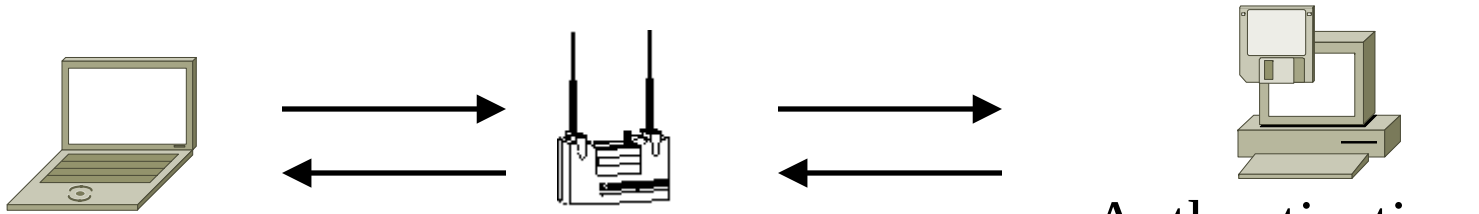
- MS-CHAPv2 in RFC 2759, Jan 2000
- MS-CHAPv2 in Windows 2000 onwards.
- Vista does not support MS-CHAPv1
- LCP option 3 =0x81 ⇒ MS-CHAPv2
- V2 provides mutual authentication between peers by piggybacking a peer challenge on the response packet and an authenticator response on the success packet.
- Does not support change password

Extensible Authentication Protocol (EAP)

- Each authentication protocols required a new protocol
  ⇒ Extensible Authentication Protocol
- Initially developed for point-to-point protocol (PPP)
- Allows using many different authentication methods
- Single-Step Protocol ⇒ Only one packet in flight
  ⇒ Duplicate Elimination and retransmission
  Ack/Nak ⇒ Can run over lossy link
- No fragmentation. Individual authentication methods can deal with fragmentation. One frag/round trip ⇒ Many round trips
- Allows using a backend authentication server ⇒ Authenticator does not have to know all the authentication methods
- Can run on any link layer (PPP, 802, ...). Does not require IP.
- Ref: RFC 3748, "EAP," June 2004.

EAP Terminology

- Peer: Entity to be authenticated = Supplicant
- Authenticator: Authenticating entity at network boundary
- Authentication Server: Has authentication database
- EAP server = Authenticator if there is no backend Authentication Server otherwise authentication server
- Master Session Key (MSK)= Keying material agreed by the peer and the EAP server. At least 64B. Generally given by the server to authenticator.

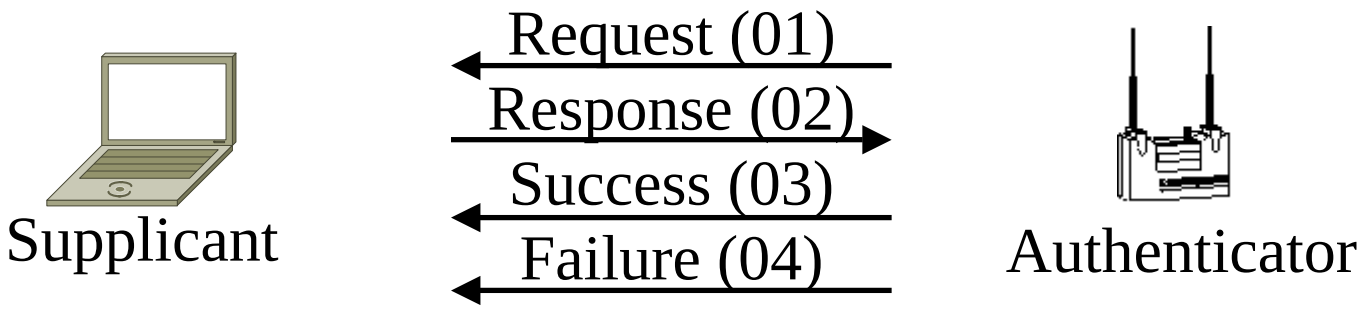


EAP Exchange

- EAP Message Format:

Code	Identifier	Length	Data
8b	8b	16b	

- Only four types of messages:



- Identifier is incremented for each message.
  Identifier in response is set equal to that in request.
- Type field in the request/response indicates the authentication.
  Assigned by Internet Assigned Number Authority (IANA)

Code	Identifier	Length	**Type**	Data

EAP Types

1 = Identity

2 = Notification (messages to be displayed to user)

3 = Nak

4 = MD5 Challenge (CHAP)

5 = One time password

6 = Generic Token card (GTC)

254 = Expanded types (allows vendor specific options)

255 = Experimental

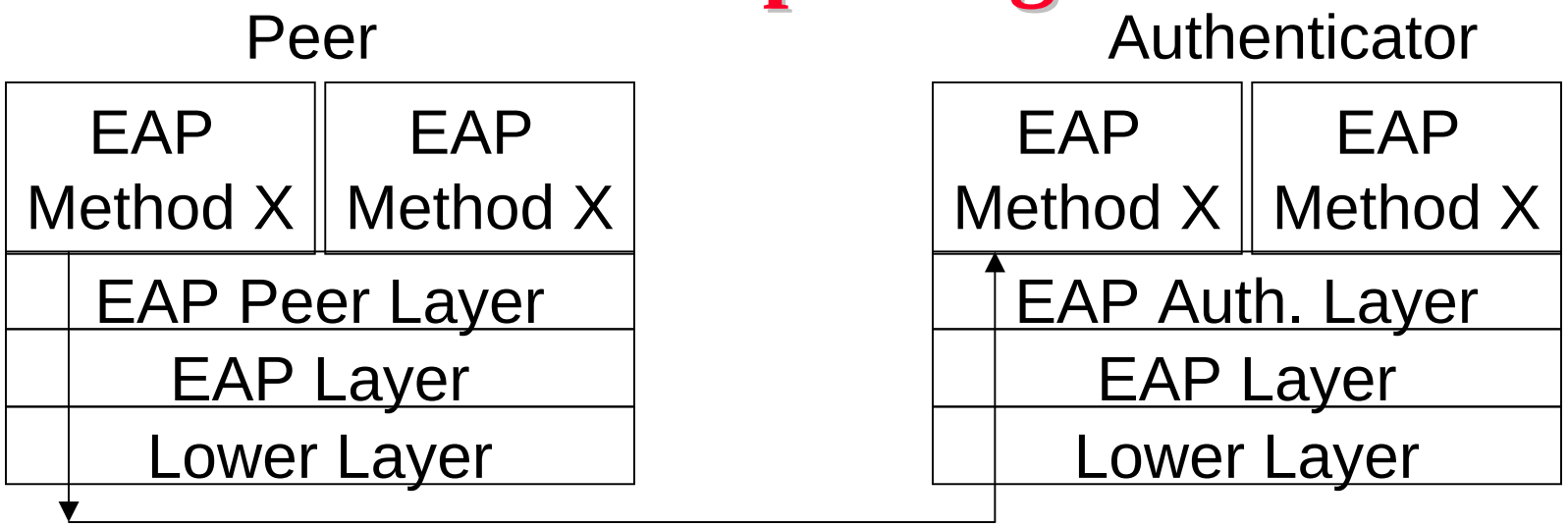
Notification requests are responded by notification responses.

Nak type is valid only for responses.

Expanded types include a 3B vendor ID and 4B vendor msg type.

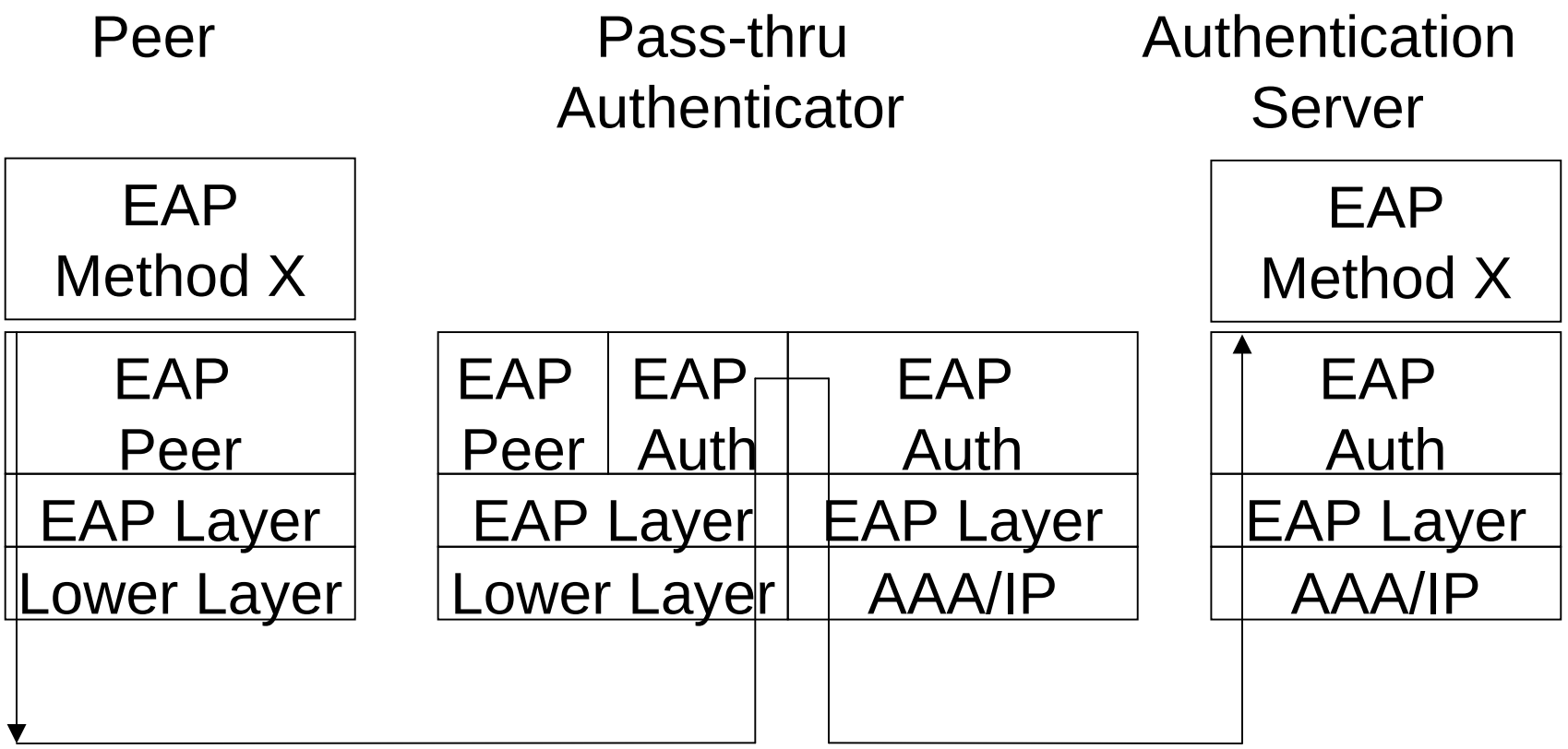
Expanded Nak is used in response to requests of type 254 and may include alternative suggestions for methods.

EAP Multiplexing Model



- Code 1 (request), 3 (success), and 4 (failure) are delivered to the peer layer
- Code 2 (response) is delivered to the EAP authenticator layer.
- Both ends may need to implement peer layer and authenticator layer for mutual authentication
- Lower layer may be unreliable but it must provide error detection (CRC)
- Lower layer should provide MTU of 1020B or greater

EAP Pass through Authenticator



EAP Upper Layer Protocols

- Lightweight EAP (LEAP)
- EAP-TLS
- EAP-TTLS
- EAP-FAST
- Protected EAP (PEAP)
- PEAPv1 or EAP-GTC
- EAP-SIM
- EAP-AKA
- EAP-PSK
- EAP-IKEv2

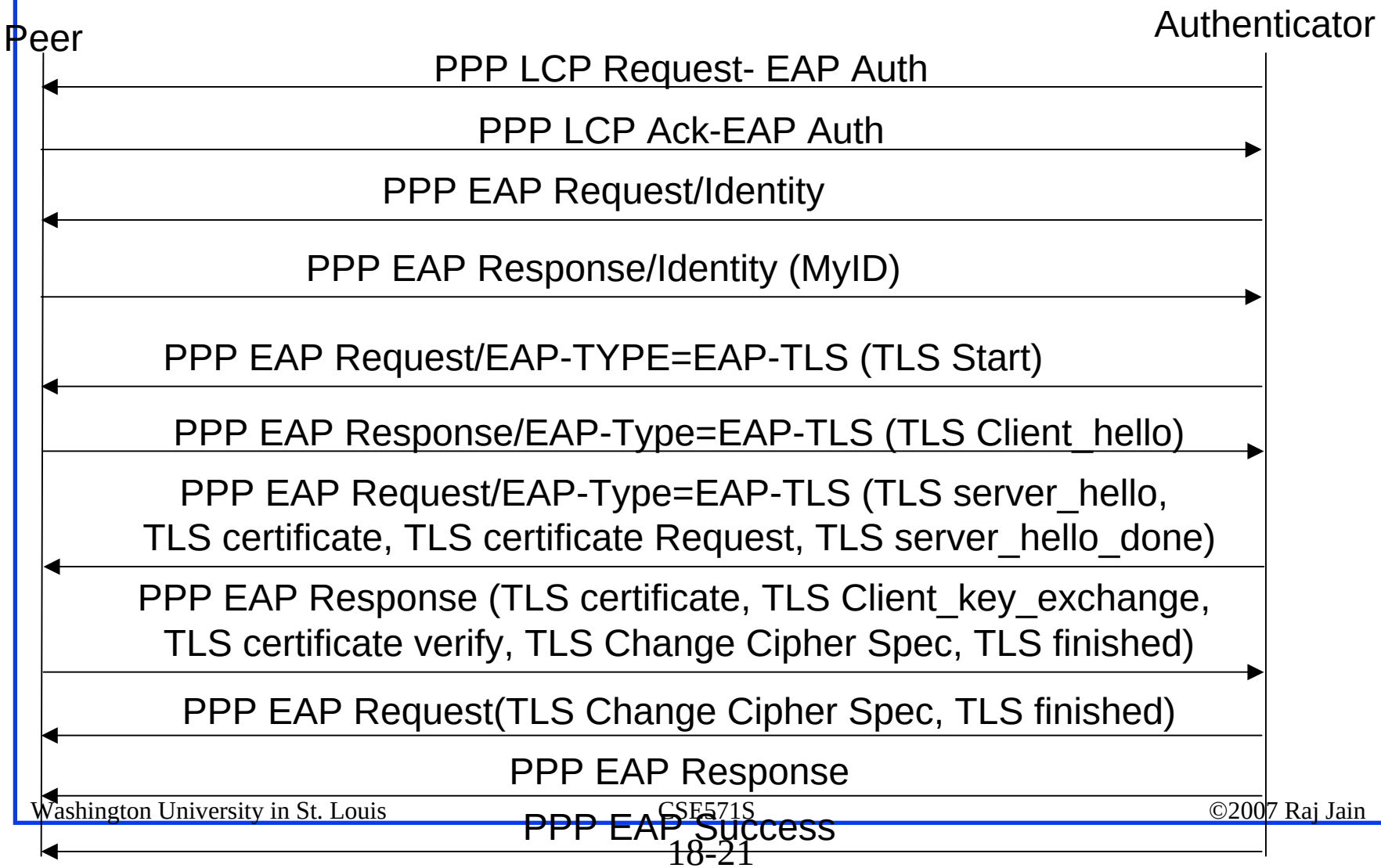
Lightweight EAP (LEAP)

- Cisco proprietary EAP
- Was used in 802.11 networks prior to 802.11i extension
- Widely adopted in networking industry
- No native support in Windows
- Uses a modified version of MS-CHAP for authentication
- An exploit tool ASLEAP has been release to break LEAP $\Rightarrow$ Not recommended now.

EAP-TLS

- TLS over EAP
- RFC 2716, Oct 1999
- Considered most secure, Universally implemented including by Microsoft, Cisco, Apple, Linux
- Supported in MAC X10+, Windows 2000, XP, Vista, Windows Mobile 2003, Windows Server 2003
- But Rarely deployed
- Both sides need a certificate
- Client side private key is housed in a smart card
- Certificate chains are big ⇒ Includes support for fragmentation and reassembly

EAP-TLS Authentication



EAP-TTLS

- Tunneled TLS over EAP
- Only server provides certificates
- Client provides password based authentication using the secure tunnel setup using TLS
- Developed by Funk Software and Centicom
- Widely supported across platforms

EAP-FAST

- Flexible Authentication via Secure Tunneling
- RFC 4851, May 2007
- Developed by Cisco as a replacement for LEAP
- Use of server certificates is optional.
- Uses a protected access tunnel (PAC) to verify credentials
- Optional Phase 0 to provision PAC manually or dynamically.
- Done once for each client-RADIUS server pair.
- In Phase 1, RADIUS server and client use PAC to TLS tunnel.
- In Phase 2, Client credentials are exchanged inside encrypted tunnel.
- Dynamic establishment of PAC is vulnerable to attack
  $\Rightarrow$ use manual provisioning.

Protected EAP (PEAP)

- One-sided TLS over EAP
- Server provides certificate ⇒ Outer authentication
- Client provides NT password hash (V0) ⇒ Inner Authentication
- Jointly developed by Microsoft, Cisco, and RSA
- Microsoft implements PEAPv0 with Inner = EAP-MS-CHAPv2
- Microsoft also implements PEAP with Client Certificates ⇒ PEAP-EAP-TLS
- Cisco supports PEAPv0 with EAP-MS-CHAPv2, EAP-SIM

PEAPv1 or EAP-GTC

- Developed by Cisco to use Generic Token Cards (GTC)
- RFC 3748, June 2004
- Server generates a challenge, client generates a response using a security token device.

Security Token

- Security Token = Small hardware device carried by users. May store cryptographic keys, biometric data (finger print), PIN entry pad.
- Based on USB, Bluetooth, Cell phones (SMS or Java)
- Use smart cards
- Two-factor authentication = What you have and what you know

[Wikipedia]

One-Time Password

- Three Types:
  1. Use a math algorithm to generate a new password based on previous
  2. Uses time to generate password
     $\Rightarrow$ Synchronized time between server and client
  3. Use a math algorithm to generate a new password based on a challenge from the server and a counter.
- Time synchronized approach allows users to generate password and not use it. The server may compare with the next n passwords to allow for time miss-synchronization.
- Non-time synchronized OTP do not need to be powered all the time $\Rightarrow$ battery lasts long. Have been attacked by phishing. Time-based OTP need to be used right-away.

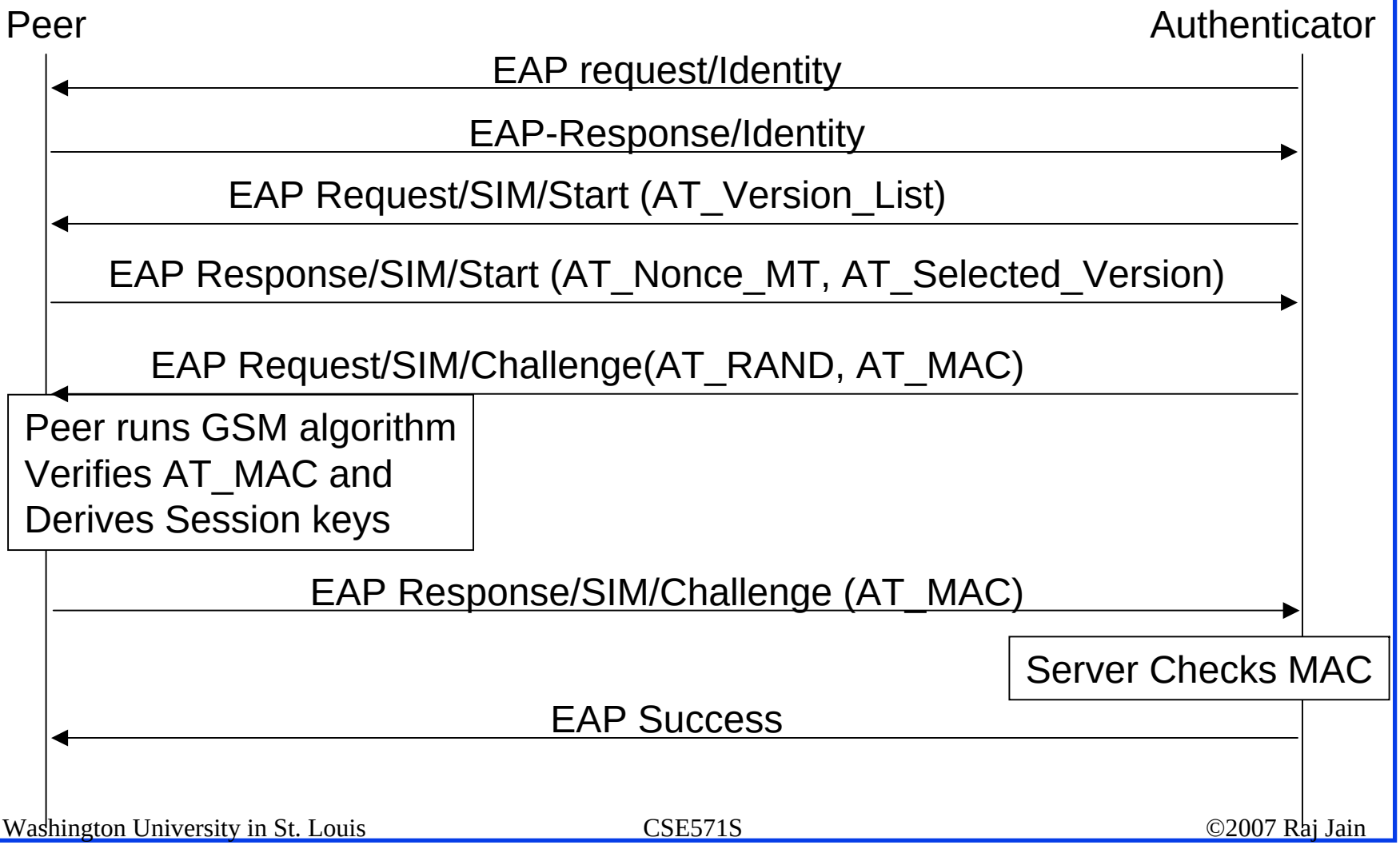
OTP (Cont)

- Most OTP devices use proprietary patented algorithms.
- HMAC-based OTP (HOTP) is proposed by Initiative for Open Authentication (OATH)
- RFC 2289, "OTP," Feb 1998.
- RFC 4226, "HOTP: An HMAC-based OTP Algorithm," Dec 2005.

EAP-SIM

- EAP for Global System for Mobile Communications (GSM) Subscriber Identity Module (SIM). RFC 4186, Jan 2006
- Optional identity privacy, fast re-auth, result indication
- Uses a challenge response mechanism. Net not authenticated
- Home auth server sends RAND: 128b Random challenge
- SIM uses shared key and generates 64b key seed Kc using a nonce. Kc used to generate encryption key
- SIM sends nonce and the response to the network
- Several challenges are run to produce several Kc which are combined to generate stronger keys for data applications.
- Temporary identifiers are used to hide subscriber identity
- EAP-success + keying material sent by EAP server to the authenticator but not passed on to user who can itself derive it.

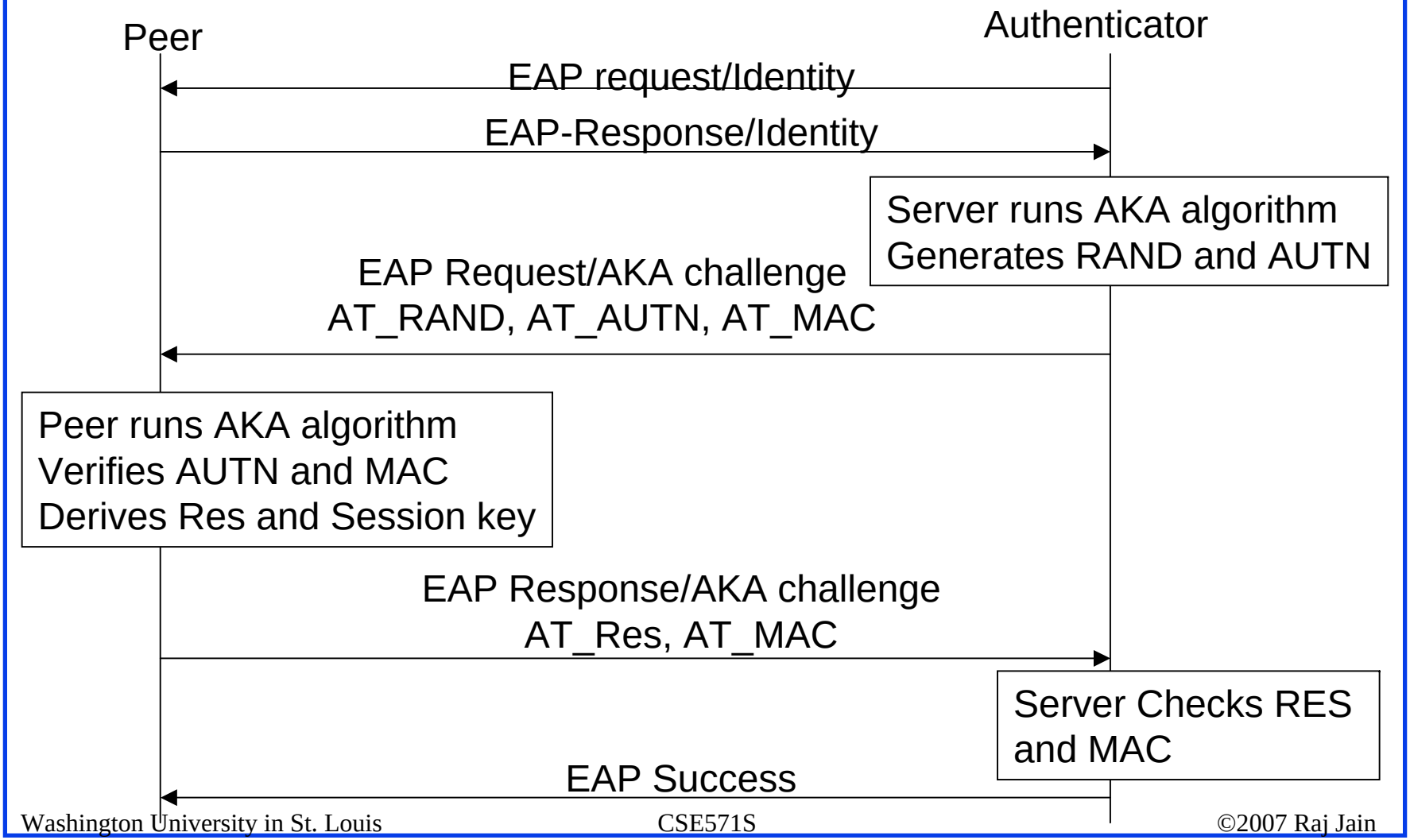
EAP-SIM Full Authentication



EAP-AKA

- EAP for 3G UMTS and CDMA2000 Authentication and Key Agreement. RFC 4187, Jan 2006
- Based on symmetric keys
- Runs in subscriber identity module (SIM)
- Optional identity privacy, fast re-auth, result indication
- Substantially longer key lengths 128b than GSM-SIM
- Network is also authenticated

EAP-AKA Full Authentication



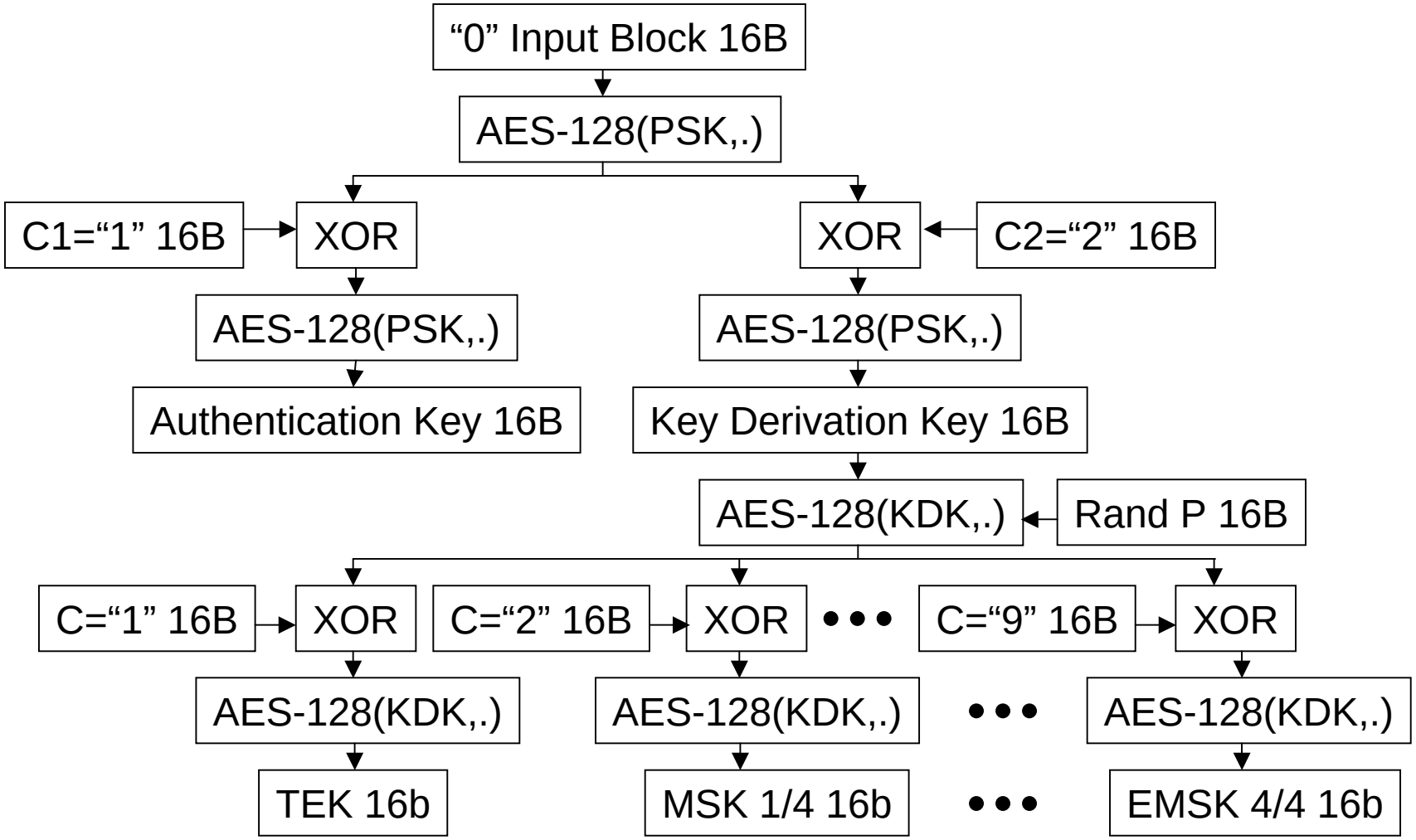
EAP-PSK

- EAP using pre-shared key
- RFC 4764, Jan 2007
- Designed for IEEE 802.11
- Does not require any infrastructure
- Uses AES-128
- Does not use Diffie-Hellman
- Does not have perfect forward secrecy, identity hiding

EAP-PSK Keys

- Pre-Shared Key (PSK): 16B
- Authentication Key (AK): 16B Derived from PSK that peer and server use for mutual authentication
- Key Derivation Key (KDK): 16B Derived from PSK to generate TEK, MSK, EMSK
  AK and KDK are derived once from PSK. Used for long time
- Master Session Key (MSK): Derived by peer and server.
  Sent by server to authenticator.
- Extended Master Session Key (EMSK): Reserved for future.
- Transient EAP Key (TEK): 128b Session key for AES-128 encryption used during authentication.
  Data encryption can use any other method
- Nonce N is a monotonically increasing sequence number starting from 0. Zero's pre-pended to 16B.

Key Derivation in EAP-PSK



EAP-IKEv2

- IKEv2 over EAP
- Mutual authentication
- Allows certificates, passwords, shared secrets
- Ref: draft-tschofenig-eap-ikev2-15.txt

EAP Upper Layer Protocols: Summary

- Lightweight EAP (LEAP): Uses MS-CHAP. Not secure.
- EAP-TLS: Both sides need certificates
- EAP-TTLS: Only server certificates. Secure tunnel for peer.
- EAP-FAST: Certificates optional. Protected tunnels.
- Protected EAP (PEAP): Server Certificates. Client password.
- PEAPv1 or EAP-GTC: Client uses secure tokens.
- EAP-SIM: Used in GSM. 64b keys.
- EAP-AKA: Used in 3G. 128b keys.
- EAP-PSK: Pre-shared key+AES-128 to generate keys
- EAP-IKEv2: Mutual authentication. Certificate, Password, or Shared secret

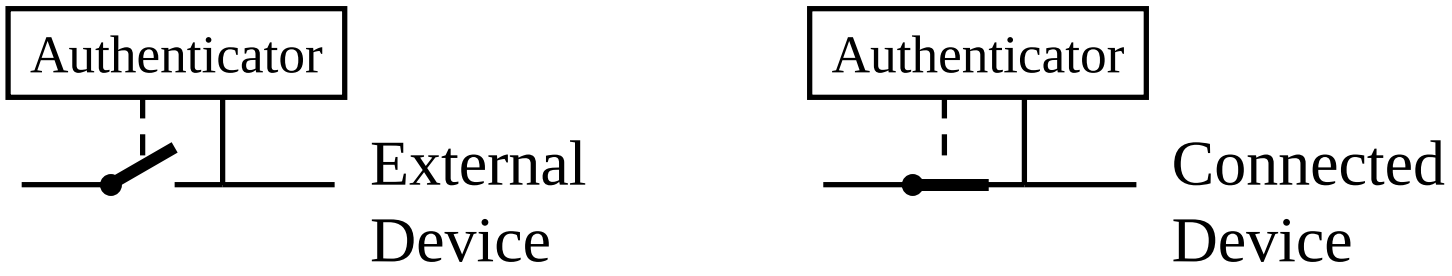
EAP over LAN (EAPOL)

- EAP was designed for Point-to-point line
- IEEE extended it for LANs ⇒ Defines EAPOL
- Added a few more messages and fields
- Five types of EAPOL messages:
  - EAPOL Start: Sent to a multicast address
  - EAPOL Key: Contains encryption and other keys sent by the authenticator to supplicant
  - EAPOL packet: Contains EAP message
  - EAPOL Logoff: Disconnect
  - EAPOL Encapsulated-ASF-Alert: Management alert
- Message Format: Version=1, Type=start,key,…,

Ethernet Header	Version	Type	Packet Body Len	Packet Body

802.1X

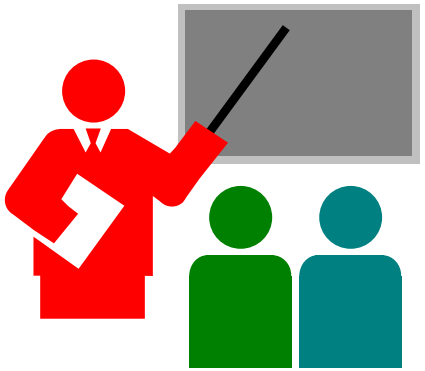
- Authentication *framework* for IEEE802 networks
- Supplicant (Client), Authenticator (Access point), Authentication server
- No per packet overhead ⇒ Can run at any speed
- Need to upgrade only driver on NIC and firmware on switches
- User is not allowed to send any data until authenticated



802.1X Authentication

- Authentication method can be changed without upgrading switches and access points
- Only the client and authentication server need to implement the authentication method

Summary



- RADIUS allows centralized authentication server and allows roaming
- EAP allows many different authentication methods to use a common framework => Authenticators do not need to know about authentication methods
- Many variations of EAP authentication methods depending upon certificates, shared secrets, passwords
- 802.1X adds authentication to LAN and uses EAPOL

Homework 18

- How would you implement Kerberos v4 over EAP in a LAN environment. Show the sequence of EAP messages that will be sent for authentication and key generation. Show also EAPOL headers on the messages.

References

- J. Edney and W.A. Arbaugh, "Real 802.11 Security: Wi-Fi Protected Access and 802.11i," Addison-Wesley, 2004, 451 pp., ISBN:0321136209
- http://en.wikipedia.org/wiki/RADIUS
- http://en.wikipedia.org/wiki/DIAMETER
- http://en.wikipedia.org/wiki/Password_Authentication_Protocol
- http://en.wikipedia.org/wiki/Challenge-handshake_authentication_protocol
- http://en.wikipedia.org/wiki/MS-CHAP
- http://en.wikipedia.org/wiki/Extensible_Authentication_Protocol#_note-0
- http://en.wikipedia.org/wiki/EAP-FAST

References (Cont)

- http://en.wikipedia.org/wiki/Protected_Extensible_Authentication_Protocol
- http://en.wikipedia.org/wiki/Security_token
- http://en.wikipedia.org/wiki/One-time_password
- http://en.wikipedia.org/wiki/EAP-SIM
- http://en.wikipedia.org/wiki/EAP-AKA
- http://en.wikipedia.org/wiki/EAP-TTLS#EAP-FAST

EAP RFCs

- RFC 2716 "PPP EAP TLS Authentication Protocol," October 1999.
- RFC 3579 "RADIUS Support For EAP," September 2003.
- **RFC 3748 "EAP," June 2004.**
- RFC 4017 "EAP Method Requirements for Wireless LANs," March 2005.
- RFC 4072 "Diameter EAP Application," August 2005.
- RFC 4137 "State Machines for EAP Peer and Authenticator," August 2005.
- **RFC 4186 "EAP Method for GSM SIMs (EAP-SIM)," January 2006.**
- **RFC 4187 "EAP Method for 3G Authentication and Key Agreement (EAP-AKA)," January 2006.**

EAP RFCs (Cont)

- RFC 4284 "Identity Selection Hints for the EEAP," January 2006.
- RFC 4746 "EAP Password Authenticated Exchange," November 2006.
- RFC 4763 "EAP Method for Shared-secret Authentication and Key Establishment (EAP-SAKE)," November 2006.
- **RFC 4764 "The EAP-PSK Protocol: A Pre-Shared Key EAP Method," January 2007.**
- **RFC 4851 "The Flexible Authentication via Secure Tunneling EAP Method (EAP-FAST)," May 2007.**

AAA RFCs

- RFC2903, "Generic AAA Architecture," Aug 2000.
- RFC2904, "AAA Authorization Framework," Aug 2000.
- RFC2905, "AAA Authorization application examples," Aug 2000.
- RFC2906, "AAA Authorization requirements," Aug 2000.
- RFC2989, "Criteria for Evaluating AAA Protocols for Network Access," Nov 2000.
- RFC3141, "CDMA2000 Wireless Data Requirements for AAA," Jun 2001.
- RFC3539, "AAA Transport Profile," Jun 2003.
- RFC3957, "AAA Registration keys for Mobile IPv4," Mar 2005.
- RFC4962, "Guidance for AAA Key Management," Jul 2007

RADIUS RFCs

- RFC2548 Microsoft Vendor-specific RADIUS Attributes, March 1999.
- RFC2809 Implementation of L2TP Compulsory Tunneling via RADIUS. April 2000.
- **RFC2865 RADIUS. June 2000.**
- RFC2866 RADIUS Accounting. June 2000.
- RFC2867 RADIUS Accounting Modifications for Tunnel Protocol Support. June 2000.
- RFC2868 RADIUS Attributes for Tunnel Protocol Support. June 2000.
- RFC2869 RADIUS Extensions. June 2000.
- RFC2882 Network Access Servers Requirements: Extended RADIUS Practices. July 2000.

RADIUS RFCs (Cont)

- RFC3162 RADIUS and IPv6. August 2001.
- RFC3575 IANA Considerations for RADIUS. July 2003.
- RFC3576 Dynamic Authorization Extensions to RADIUS. July 2003.
- RFC3579 RADIUS Support For Extensible Authentication Protocol (EAP). September 2003.
- RFC3580 IEEE 802.1X RADIUS Usage Guidelines. September 2003.
- RFC4014 RADIUS Attributes Suboption for the DHCP Relay Agent Information Option. February 2005.
- RFC4590 RADIUS Extension for Digest Authentication. July 2006.
- RFC4668 RADIUS Authentication Client MIB for IPv6. August 2006.

RADIUS RFCs (Cont)

- RFC4669 RADIUS Authentication Server MIB for IPv6. August 2006.
- RFC4670 RADIUS Accounting Client MIB for IPv6. August 2006.
- RFC4671 RADIUS Accounting Server MIB for IPv6. August 2006.
- RFC4672 RADIUS Dynamic Authorization Client MIB. September 2006.
- RFC4673 RADIUS Dynamic Authorization Server MIB. September 2006.
- RFC4675 RADIUS Attributes for Virtual LAN and Priority Support. September 2006.
- RFC4679 DSL Forum Vendor-Specific RADIUS Attributes. September 2006.

RADIUS RFCs (Cont)

- RFC4818 RADIUS Delegated-IPv6-Prefix Attribute. April 2007.
- RFC4849 RADIUS Filter Rule Attribute. April 2007.
- RFC5030 Mobile IPv4 RADIUS Requirements. October 2007.

Diameter RFCs

- RFC3588 Diameter Base Protocol. September 2003.
- RFC3589 Diameter Command Codes for 3GPP Release 5. September 2003.
- RFC4004 Diameter Mobile IPv4 Application. August 2005.
- RFC4005 Diameter Network Access Server Application. August 2005.
- RFC4006 Diameter Credit-Control Application. August 2005.
- RFC4072 Diameter EAP Application. August 2005.
- RFC4740 Diameter SIP Application. November 2006.