UNIVERSITA' DEGLI STUDI DI PARMA
Dipartimento di Ingegneria dell'Informazione

Security protocols: Authentication

Luca Veltri
(mail.to: luca.veltri@unipr.it)

Corso di Sicurezza nelle reti di telecomunicazioni, a.a. 2006/2007
http://www.tlc.unipr.it/veltri

Authentication Authorization Accounting

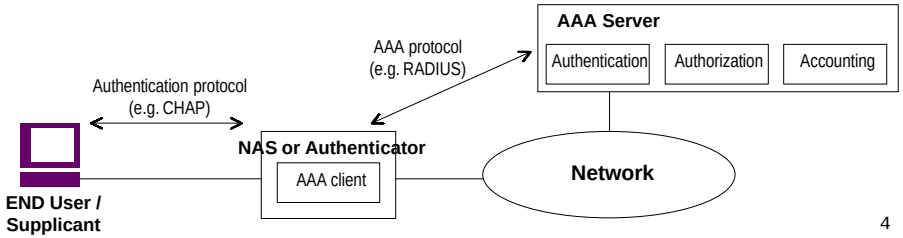
- Authentication
  - **involves validating the end users' identity prior to permitting them network access**
  - **the end-user needs to posses an unique piece of information that serves as unambiguous identification credentials**
    - a username/password combination,
    - a secret key, or
    - biometric data (fingerprints, for example))
  - **the result of the authentication depends on the matching of the user supplied data with the stored data**
- Authorization
  - **defines what rights and services the end user is allowed once network access is granted**
    - this might include providing an IP address, invoking a filter to determine which applications or protocols are supported, and so on
  - **Authentication and authorization are usually performed together**

Authentication Authorization Accounting

- Accounting
  - **the third "A," provides the methodology for collecting information about the end user's resource consumption**
  - **which can then be processed for billing, auditing, and capacity-planning purposes**

AAA general model

- AAA processing can be summarized in the following steps:
  - **End user connects to the point-of-entry device (authenticator) and requests access to the system**
    - this device is typically called NAS (Network Access Server)
    - it contains an AAA client function
  - **authenticator collects and forwards the end user's credentials to the AAA server**
  - **AAA server processes the data and returns an accept or reject response and other relevant data to the AAA client**
  - **authenticator notifies the end user that access is granted or denied for the specified resources**



AAA servers

- the AAA server can be housed on a general-purpose computing system
- A single AAA server can act as a centralized administrative control point for multiple AAA clients contained within different vendor-sourced NAS and network components

AAA protocols

Authentication protocols:

- PAP *(Password Authentication Protocol)*
- CHAP *(Challenge Handshake Authentication Protocol)*
- EAP *(Extensible Authentication Protocol)*

Access control and AAA protocols:

- IEEE 802.1X
- RADIUS *(Remote Authentication Dial In User Service)*
- Diameter
- etc..

PAP

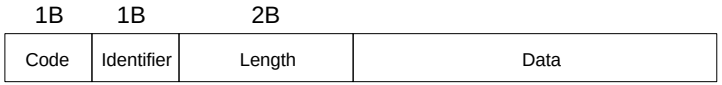
- Password Authentication Protocol (PAP)
  - **Costituisce il protocollo di autenticazione più semplice, impiegato nelle comunicazioni che utilizzano il Point to Point Protocol (PPP)**
  - **Prevede lo scambio di username e password in chiaro tra le entità che intendono autenticarsi (possibilità di autenticazione monodirezionale, bidirezionale)**

CHAP

- Challenge Handshake Authentication Protocol (CHAP)
  - **Fu sviluppato per eliminare i limiti del PAP**
  - **Prevede che l'entità che deve autenticare sfidi l'entità che intende autenticarsi (meccanismo Challenge-response) : la sfida consiste nella verifica che l'entità che intende autenticarsi sia in possesso di una shared secret condivisa con l'entità autenticante**
  - **l'entità che si vuole autenticare a partire dal challenge calcola il valore di response tramite una "one-way hash" function del challenge + un segreto (una chiave o passwd).**
  - **L'entità che deve autenticare controlla la risposta calcolando localmente il valore hash atteso**
  - **Ad intervalli regolari l'autenticatore può inviare nuove sfide (challenges)**
  - **Esistono varie implementazioni**
    - Shiva Propietary-Password Authentication Protocol (SPAP)
    - Appletalk Remote Access Protocol (ARAP)
    - Microsoft CHAP (MSCHAP)
  - **IETF RFC 1994, "PPP Challenge Handshake Authentication Protocol (CHAP)", August 1996**

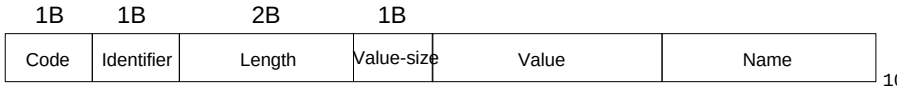
CHAP packet format

- Code field [1B]
  - **identifies the type of CHAP packet**
    1. Challenge
    2. Response
    3. Success
    4. Failure
- Identifier field [1B]
  - **aids in matching challenges, responses and replies**
- Length field [2B]
  - **indicates the length of the CHAP packet**
- Data field
  - **is zero or more octets; the format is determined by the Code field**



CHAP Request/Response format

- Value-Size [1B]
  - **indicates the length of the Value field**
- Value
  - **The Challenge Value is a variable stream of octets; it MUST be changed each time a Challenge is sent;**
    the length depends upon the method used to generate the octets, and is independent of the hash algorithm used
  - **The Response Value is the one-way hash calculated over a stream of octets consisting of the Identifier, concatenated with the "secret", concatenated with the Challenge Value;**
    the length depends upon the hash algorithm (16 octets for MD5)
- Name field
  - **identification of the system transmitting the packet**



CHAP bidirectional authentication

- The authentication can be full duplex
  - **the same protocol can be used in both directions**
- The secret SHOULD NOT be the same in both directions
  - **This would allow an attacker to replay the peer's challenge, accept the computed response, and use that response to authenticate.**

One-Time Password (OTP)

- One-time password systems are designed to counter "replay attack"
- It uses a sequence of one-time (single use) passwords; that are not sent through the network during authentication
- Two entities: generator and server
- The generator
  - **produces the appropriate one-time password from the user's secret pass-phrase and from information provided in the challenge from the server**
- The server
  - **send a challenge that includes the appropriate generation parameters to the generator,**
  - **verify the one-time password received,**
  - **store the last valid one-time passwd received, and the seq number**

OTP - Computation Step

- A sequence of one-time passwords is produced by applying the secure hash function a number times (N) to the output of the initial step (called S)
- The next one-time password to be used is generated by passing S though the secure hash function N-1 times
- The sequence number and seed together constitute a larger unit of data called the challenge
- The syntax of the challenge is:

  otp-<algorithm identifier> <sequence integer> <seed>

EAP (Extensible Authentication Protocol)

- RFC 2284 (March 1998), obsolated by RFC 3748 (June 2004)
- Protocollo generale di autenticazione che può supportare diversi metodi tra cui OTP,TLS,CHAP…
- Esistono 4 tipi di pacchetti: EAP-request, EAP-response, EAP-success, EAP-failure
- Ad ogni request deve corrispondere sempre una response. Le richieste e le relative risposte possono essere di più tipi tra cui:
  - **identity,**
  - **EAP-OTP,**
  - **EAP-TLS,**
  - **EAP-MS-CHAP-v2…**
- Il tipo di richiesta è specificato in un preciso campo (EAP-type) dell'EAP-request, stesso vale per le risposte
- I pacchetti EAP-success ed EAP-failure non contengono dati

EAP properties

- EAP typically runs directly over data link layers such as PPP or IEEE 802, without requiring IP
  - **may be used on dedicated links as well as switched circuits, and wired as well as wireless links**
    - encapsulated in PPP
    - over IEEE 802: EAPOL (EAP over LAN)
- provides its own support for duplicate elimination and retransmission, but is reliant on lower layer ordering guarantees
- It is used to select a specific authentication mechanism
  - **typically after the authenticator requests more information in order to determine the specific authentication method to be used**
- It permits the use of a backend authentication server which may implement some or all authentication methods
  - **the authenticator acts as a pass-through**

- EAP is a peer-to-peer protocol, an independent and simultaneous authentication may take place in the reverse direction
  - **both peers may act as authenticators and authenticatees at the same time**
- EAP protocol can support multiple authentication mechanisms without having to pre-negotiate a particular one
- Network Access Server (NAS) devices (e.g., a switch or access point) do not have to understand each authentication method
  - **they MAY act as a pass-through agent for a backend authentication server**
  - **Separation of the** *authenticator* **from the** *backend authentication server* **simplifies credentials management and policy decision making**

EAP entities

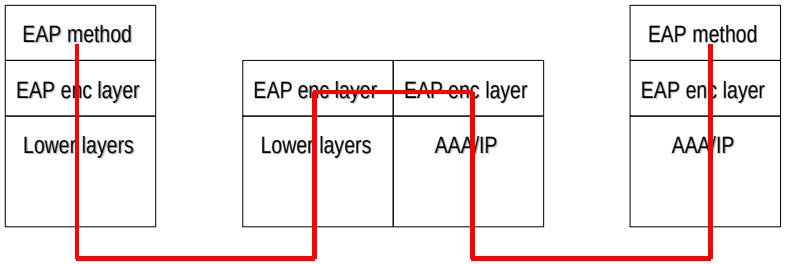
- Authenticator
  - **The end of the EAP link initiating EAP authentication (as in IEEE-802.1X)**
- Peer
  - **The end of the EAP Link that responds to the authenticator (the Supplicant in IEEE-802.1X)**
- Backend authentication server
  - **an entity that provides an authentication service to an authenticator (as in IEEE-802.1X)**
  - **when used, this server typically executes EAP methods for the authenticator**

EAP authentication exchange

- The authenticator sends a Request to authenticate the peer
  - **The type field indicates the type of authentication method**
  - **Typically, the authenticator may send also an initial Identity Request; the identity may not be required where it is determined by the port to which the peer has connected or where the identity is obtained in another fashion (via calling station identity or MAC address, etc.)**
- The peer sends a Response packet in reply to a valid Request
  - **Response packet contains a Type field, which corresponds to the Type field of the Request**
- The authenticator sends an additional Request packet, and the peer replies with a Response
  - **the sequence of Requests and Responses continues as long as needed**
- The authenticator transmits an EAP Failure (Code 4) or an EAP Success (Code 3)

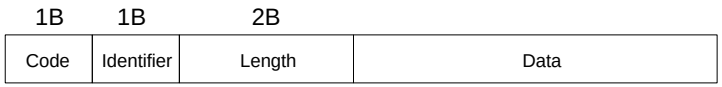
Pass-through Authenticator

- Where an authenticator operates as a pass-through, it forwards packets between the peer and a backend authentication server, based on the EAP layer header fields (Code, Identifier, Length)



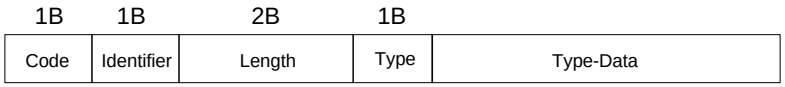
EAP packet format

- Code field [1B]
  - **identifies the type of EAP packet**
    1. Request
    2. Response
    3. Success
    4. Failure
- Identifier field [1B]
  - **aids in matching Responses with Requests**
- Length field [2B]
  - **indicates the length of the EAP packet**
- Data field
  - **is zero or more octets; the format is determined by the Code field**



EAP Request/Response format

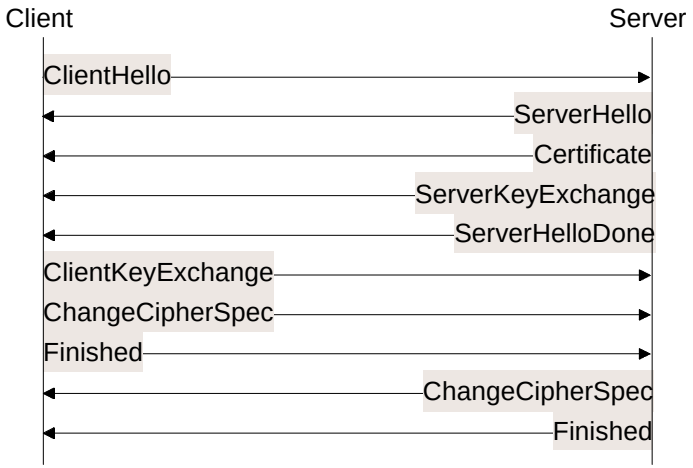
- Type field [1B]
  - **indicates the Type of Request or Response**
  - **some types**
    - **1 Identity (used to query peer identity)**
    - **2 Notification (used to convey displayable messages)**
    - **3 Nak (Response only, utilized for the purposes of method negotiation)**
    - **4 MD5-Challenge**
    - **5 One Time Password (OTP)**
    - **6 Generic Token Card (GTC)**
    - **13 EAP TLS**
  - **normally, the Type field of the Response will be the same as the Type of the Request**
- The Type-Data field
  - **varies with the Type of Request and the associated Response**



Metodi di autenticazione usati da EAP

- OTP
  - **si basa sull'invio di un challenge da parte dell'authentication server al client**
  - **challenge che viene utilizzata per generare una passwd (diversa ad ogni autenticazione)**
- CHAP
  - **Si basa anch'esso sull'invio di un challenge**
  - **cambia però il tipo di elaborazione che viene fatta su di esso per generare la risposta**
  - **può prevedere authenticazione mutua**
- TLS
  - **può essere utilizzato anche all'interno di EAP come metodo di autenticazione**
  - **grazie al suo protocollo di handshake garantisce mutua autenticazione e la creazione di un master secret utilizzato per generare una sessione di chiavi necessarie per operazioni quali la crittografia e il controllo di integrità**

TLS Handshake



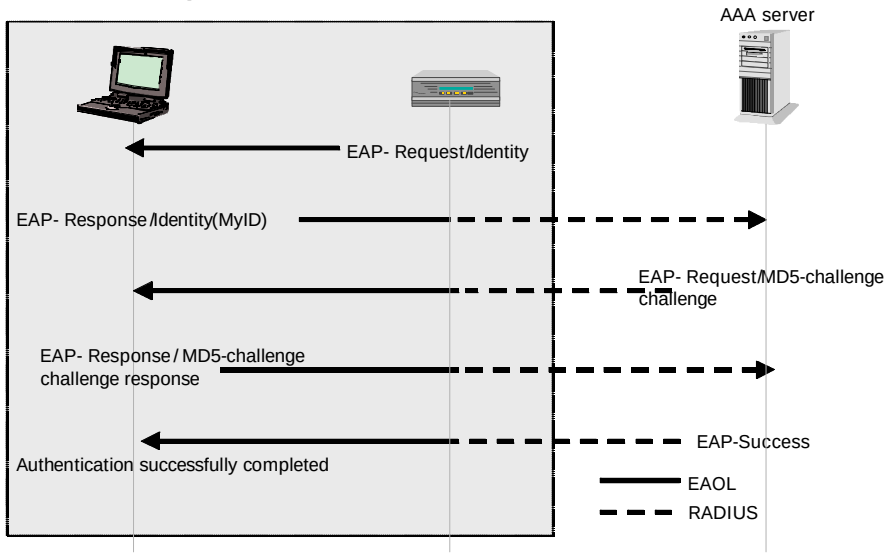
EAP/TLS

- The authenticator will then typically send an EAP-Request/Identity packet to the peer
  - **the peer will respond with an EAP-Response/Identity packet to the authenticator, containing the peer's userId**
- the EAP server sends an EAP-TLS/Start packet
  - **an EAP-Request packet with EAP-Type=EAP-TLS, the Start (S) bit set, and no data**
- The EAP-TLS conversation begins, with the peer sending an EAP-Response packet with EAP-Type=EAP-TLS
  - **The data field of that packet encapsulates one or more TLS records in TLS record layer format, containing a TLS client_hello handshake message**
  - **The current cipher spec for the TLS records is null and null compression**
  - **The client_hello message contains the TLS version, a sessionId, a random number, and a set of ciphersuites supported by the client**

- The EAP server responds with an EAP-Request packet with EAP-Type=EAP-TLS
  - **The data field of this packet encapsulates one or more TLS records, containing a TLS server_hello handshake message, possibly followed by TLS certificate, server_key_exchange, certificate_request, server_hello_done and/or finished handshake messages, and/or a TLS change_cipher_spec message**
  - **If the EAP server is not resuming a previously established session, then it includes a TLS server_certificate handshake message**
  - **A certificate_request message is included when the server desires the client to authenticate itself via public key**
- The peer responds with an EAP-Response packet of EAP-Type=EAP-TLS
  - **The data field of this packet will encapsulate one or more TLS records containing a TLS change_cipher_spec message and finished handshake message, and possibly certificate, certificate_verify and/or client_key_exchange handshake messages**

  - **If the preceding server_hello message sent by the EAP server in the preceeding EAP-Request packet did not indicate the resumption of a previous session, then the peer sends, in addition a client_key_exchange message, which completes the exchange of a shared master secret between the peer and the EAP server**
- If the peers authenticates successfully, the EAP server responds with an EAP-Request packet with EAP-Type=EAP-TLS
  - **it includes one or more TLS records containing TLS change_cipher_spec and finished handshke messages. The latter contains the EAP server's authentication response to the peer**
- If the EAP server authenticates successfully, the peer sends an EAP-Response packet of EAP-Type=EAP-TLS, and no data
- The EAP-Server responds with an EAP-Success message

Esempio di autenticazione tramite EAP



RADIUS

RADIUS

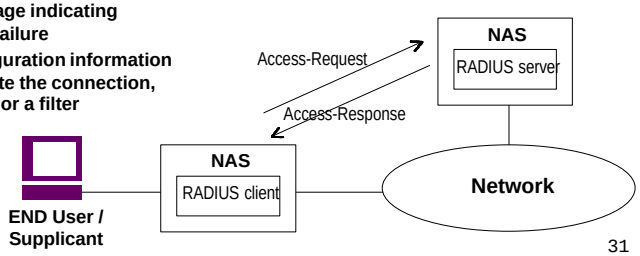
- Remote Access Dial-In User Service (RADIUS) - RFC 2138
- The best-known and most widely deployed AAA protocol
- developed in the mid-1990s by Livingston Enterprises (since acquired by Lucent) for providing authentication and accounting services to their NAS devices
- The IETF formalized that effort in 1996 with the RADIUS WG

RADIUS functional attributes

- Client-server-based operations
  - **A RADIUS client resides on the NAS and communicates over the network with a RADIUS server running on a host computer**
  - **Additionally, a RADIUS server may serve as a proxy client for another RADIUS or authentication server**
- Network security
  - **All communications between a RADIUS client and server are authenticated by virtue of a shared secret key that is never sent over the network**
  - **In addition, user passwords contained in RADIUS messages are encrypted to prevent hackers from reading them by snooping the network**
- Flexible authentication
  - **RADIUS can support multiple authentication mechanisms, including PAP and CHAP**

Typical RADIUS configuration

- An end-user dials into a NAS
- Using a prompt, or perhaps PPP frames, the NAS collects the username and password from the end user
- It then forward an encrypted Access-Request message over the network to the RADIUS server
  - **The message may also contain attributes such as the NAS port ID and IP address**
- The RADIUS server then checks the User-Name attribute and returns an Access-Reject or an Access-Accept message to the NAS
  - **optional text message indicating the reason for the failure**
  - **or additional configuration information required to complete the connection, such as an IP addr or a filter**



RADIUS functional attributes (cont.)

- ***Attribute/Value Pairs***
  - **RADIUS messages carry AAA information encoded in type-length-value fields, called attributes (or attribute/value pairs - AVP)**
  - **Common examples of attributes include**
    - User-Name,
    - User-Password,
    - Framed-Protocol (such as PPP),
    - Framed-IP-Address (IP address for end user),
    - and so on

Diameter

Diameter

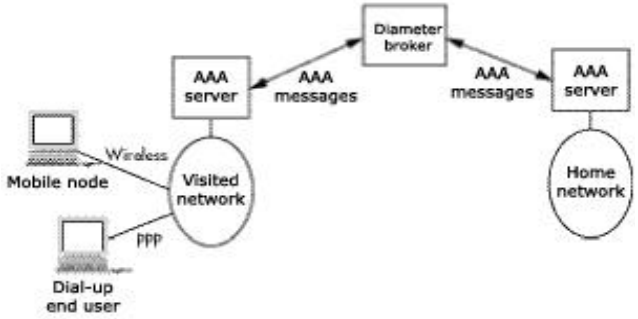
- Both RADIUS and TACACS+ protocols were originally engineered for small network devices supporting just a few end-users requiring simple server-based authentication
- Dial providers must now provide AAA services for hundreds and thousands of concurrent end users accessing network services over a variety of technologies
- They must also support AAA services across ISP boundaries in a secure and scalable manner
- IETF has develop a next-generation AAA protocol: “Diameter”

Diameter characteristics

- Lightweight, peer-based AAA protocol
- Designed to offer a scalable foundation for introducing new policy and AAA services over existing (PPP) and emerging (roaming, mobile IP) network technologies
- It employs many of the same mechanisms as RADIUS, including UDP transport, encoded attribute/value pairs, and proxy server support
- Diameter also attempts to correct limitations inherent in the RADIUS protocol
  - **for example, a RADIUS attribute value cannot exceed 255 bytes..**
- Diameter supports TCP or SCTP (Stream Control Transmission Protocol) transports
- Diameter permits unsolicited commands from server to client
  - **useful to perform a specific accounting function**

Diameter characteristics (cont.)

- Diameter provides an end-to-end security mechanism that is not found in RADIUS
- Diameter was designed from the beginning to support roaming and mobile IP networks



- The broker can act as a certificate authority (CA)

IEEE 802.1X
Port-Based Network Access Control

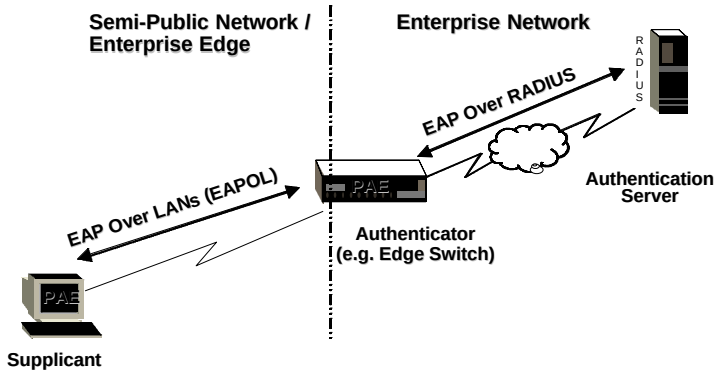
802.1X Motivation and History

- Increased use of 802 LANs in public and semi-public places
- Desire to provide a mechanism to associate end-user identity with the port of access to the LAN
  - **establish authorized access**
  - **enable billing and accounting mechanisms**
  - **personalize network access environment**
- Leverage existing AAA infrastructure currently used by other forms of network access (e.g. dial-up).
- Initially intended for 802.1D (MAC Bridges), but since expanded to include other access devices (e.g. 802.11, smart repeater).

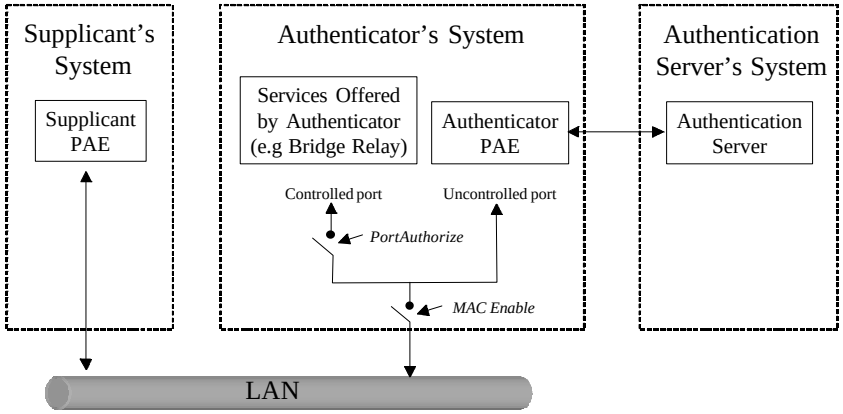
IEEE 802.1X Entities

- Authenticator
  - **The entity that requires the entity on the other end of the link to be authenticated**
- Supplicant
  - **The entity being authenticated by the Authenticator and desiring access to the services of the Authenticator**
- Authentication Server
  - **An entity providing authentication service to the Authenticator**
  - **Maybe co-located with Authenticator, but most likely an external server**
- Port Access Entity (PAE)
  - **The protocol entity associated with a port. May support functionality of Authenticator, Supplicant or both**

General Topology of 802.1X



Principal of Operation



Protocol Overview

- Encapsulate the Extensible Authentication Protocol (RFC 2284) in 802 Frames (EAPOL) with a few extensions to handle unique characteristics of 802 LANs
  - **EAP is a general protocol supporting multiple authentication methods (smart cards, Kerberous, public key, one-time password, etc).**
- Authenticator passes authentication exchanges between supplicant and authentication server
- Authenticator PAE enables the controlled port based upon the result of the authentication exchanges

IEEE 802.1X Conversation