1

Access Control Solutions for Next Generation Networks

F. Pereniguez-Garcia, R. Marin-Lopez and A.F. Gomez-Skarmeta
Faculty of Computer Science, University of Murcia
Spain

1. Introduction

In recent years, wireless telecommunications systems have been prevalently motivated by the proliferation of a wide variety of wireless technologies, which use the air as a propagation medium. Additionally, users have been greatly attracted for wireless-based communications since they offer an improved user experience where information can be exchanged while changing the point of connection to the network. This increasing interest has led to the appearance of mobile devices such as smart phones, tablet PCs or netbooks which, equipped with multiple interfaces, allow *mobile users* to access network services and exchange information anywhere and at any time. To support this *always-connected* experience, communications networks are moving towards an *all-IP* scheme where an IP-based network core will act as connection point for a set of accessible networks based on different wireless technologies. This future scenario, referred to as the *Next Generation Networks* (NGNs), enables the convergence of different heterogeneous wireless access networks that combine all the advantages offered by each wireless access technology per se.

In a typical NGN scenario users are expected to be potentially mobile. Equipped with wireless-based multi-interface lightweight devices, users will go about their daily life (which implies to perform movements and changes of location) while demanding access to network services such as VoIP or video streaming. The concept of *mobility* demands session continuity when the user is moving across different networks. In other words, active communications need to be maintained without disruption (or limited breakdown) when the user changes its connection point to the network during the so-called *handoff*.

This aspect is of vital importance in the context of NGNs to allow the user to roam seamlessly between different networks without experiencing temporal interruption or significant delays in active communications. Nevertheless, during the handoff, the connection to the network may for various reasons be interrupted, which causes a packet loss that finally impacts on the on-going communications.

Thus, to achieve mobility without interruptions and improve the quality of the service perceived by the user, it is crucial to reduce the time required to complete the handoff. The handoff process requires the execution of several tasks (N. Nasser et al. (2006)) that negatively affect the handoff latency. In particular, the authentication and key distribution processes have been proven to be one of the most critical components since they require considerable time (A. Dutta et al. (2008); Badra et al. (2007); C. Politis et al. (2004); Marin-Lopez et al. (2010); R. M. Lopez et al. (2007)). The implantation of these processes during the *network access control*

demanded by network operators is destined to ensure that only allowed users can access the network resources in a secure manner. Thus, while necessary, these security services must be carefully taken into account, since they may significantly affect the achievement of seamless mobility in NGNs.

In this chapter we are going to revise the different approaches that have been proposed to address this challenging issue in future NGNs. More precisely, we are going to carry out this analysis in the context of the *Extensible Authentication Protocol* (EAP), a protocol which is acquiring an important position for implementing the access control solution in future NGNs. This interest is motivated by the important features offered by the protocol such as flexibility and media independence. Nevertheless, the EAP authentication process has shown certain inefficiency in mobile scenarios. In particular, a typical EAP authentication involves a considerable signalling to be completed. The research community has addressed this problem by defining the so-called *fast re-authentication* solutions aimed at reducing the latency introduced by the EAP authentication. Throughout this chapter, we will revise the different groups of fast re-authentication solutions according to the strategy followed to minimize the authentication time.

The remaining of the chapter is organized as follows. Section 2 describes the different technologies related to the network access authentication. Next, Section 3 outlines the deficiencies of EAP in mobile environments, which have motivated the research community the proposal of fast re-authentication solutions. The different fast re-authentication schemes proposed so far are analyzed in Section 4. Finally, the chapter finalizes with Section 5 where the most relevant conclusions are extracted.

2. Protocols involved in the network access service

2.1 AAA infrastructures: Authentication, Authorization and Accounting (AAA)

Network operators need to control their subscribers so that only authenticated and authorized ones can access to the network services. Typically, the correct support of a controlled access to the network service has been guaranteed by the deployment of the so-called *Authentication, Authorization and Accounting* (AAA) infrastructures (C. de Laat et al. (2000)). AAA essentially defines a framework for coordinating these individual security services across multiple network technologies and platforms.

An overview of the different components is the best way to understand the services provided by the AAA framework.

- *Authentication*. This service provides a means of identifying a user that requires access to some service (e.g., network access). During the authentication process, users provide a set of credentials (e.g., password or certificates) in order to verify they are who they claim to be. Only when the credentials are correctly verified by the AAA server, the user is granted access to the service.
- *Authorization*. Authorization typically follows the authentication and entails the process of determining whether the client is allowed to perform and request certain tasks or operations. Authorization is the process of enforcing policies, determining what types or qualities of activities, resources or services a user is permitted.
- *Accounting*. The third component in the AAA framework is accounting, which measures the resources a user consumes during network access. This can include the amount of time

a service is used or the amount of data a user has sent and/or received during a session. Accounting is carried out by gathering session statistics and usage information, and it is used for different purposes like billing.

The following sections provide a detailed description for the general AAA architecture and the most relevant AAA protocols.

2.1.1 Generic AAA architecture

The general AAA scheme, as defined in (C. de Laat et al. (2000)), requires the participation of four different entities (see Fig. 1) that take part in the authentication, authorization and accounting processes:

- A *user* desiring to access a specific service offered by the network operator.
- A *domain* where the user is registered. This domain, typically referred to as *home domain*, is able to verify the user's identity based on some credentials. Optionally, the home domain not only authenticates but also provides authorization information to the user
- A *service provider* controlling the access to the offered services. The service provider can be implemented by the domain where the user is subscribed to (home domain) or by a different domain in the roaming cases. In the case the service provider is located outside the home domain, the access to the service is provided on condition that an agreement is established between the service provider and the home domain. These bilateral agreements, which may take the form of formal contracts known as *Service Level Agreements* (SLAs), suppose the establishment of a trust relationship between the involved domains that will allow the service provider to authenticate and authorize foreign users coming from another administrative domains.
- A *service provider's service equipment* which will be typically located on a device that belongs to the service provider. For example, in the case of network access service, this role is played by the *Network Access Server* (NAS) like, for example, an 802.11 access point.

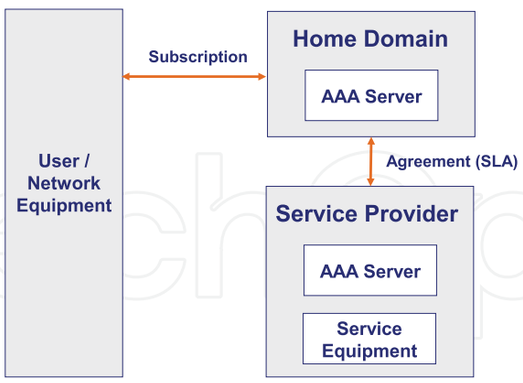


Fig. 1. Generic AAA architecture

2.1.2 Relevant AAA protocols

To allow the communication between AAA servers, it is required the deployment of a *AAA protocol*. Nowadays, the most relevant AAA protocols are RADIUS (C. Rigney et al. (2000)) and Diameter (P. Calhoun & J. Loughney (2003)). Despite Diameter is the most complete AAA protocol, RADIUS is the most widely deployed one in current AAA infrastructures. In the following, it is provided a brief overview of both.

2.1.2.1 RADIUS

RADIUS is a client-server protocol where a NAS usually acts as *RADIUS client*. During authentication procedures, the RADIUS client is responsible for passing user information in the form of requests to the *RADIUS server* and waits for a response from the server. Depending on the policy, the NAS may only need a successful authentication or further authorization directives from the server to enable data traffic to the client. The RADIUS server, on the other hand, is responsible for processing requests, authenticating the users and returning the information necessary for user-specific configuration to deliver the service.

The typical RADIUS conversation consists of the following messages:

- *Access-Request*. This message is sent from the RADIUS client (NAS) to the server to request authentication and authorization for a particular user.
- *Access-Challenge*. This message, sent from the RADIUS server to the client, is used by the server to obtain more information from the NAS about the end user in order to make a decision about the requested service.
- *Access-Accept*. This message is sent from the RADIUS server to the NAS to indicate a successful completion of the request.
- *Access-Reject*. This message is sent by the server to indicate the rejection of a request.

Typically, the main part of a RADIUS conversation consists of several Access-Request/Access-Challenge message exchanges where the RADIUS client and server exchange information transported within RADIUS attributes. Depending on whether the client is successfully authenticated or not, the RADIUS server finalizes the communication with an *Access-Accept* or *Access-Reject*, respectively.

Apart from these main messages, the RADIUS base specification defines some others to transmit accounting information (*Accounting-Request/Accounting-Response*) or the status of the RADIUS entities (*Status-Client/Status-Server*).

Regarding the protocol used to transport RADIUS messages, protocol designers considered that the *User Datagram Protocol* (UDP) was the most appropriate one since the *Transmission Control Protocol* (TCP) session establishment is a time-consuming process requiring the management of connection state. Nevertheless, the lack of a reliable transport causes serious problems to RADIUS. For example, clients are unable to distinguish when a request is received by the server or a communication problem has occurred and the RADIUS packet has not reached its destination. Similarly, a client cannot distinguish whether a server is down or discarding requests.

RADIUS security is another aspect that was not deeply considered. In particular, it is based on the use of shared secrets between the RADIUS client and the server. In real deployments, this basic security mechanism has been known to cause several vulnerabilities:

- Shared secrets must be statically configured. No method for dynamic shared secret establishment is defined in the RADIUS protocol.
- Shared secrets are determined according to the source IP address in the RADIUS packet. This introduces management problems when the client's IP address change.
- When using RADIUS proxies, the RADIUS client only shares a secret with the RADIUS server in the first hop and not with the ultimate RADIUS server. In other words, the trust

relationship between the RADIUS client and the final RADIUS server is transitive rather than using a direct trust relationship. If a server in the chain is compromised, some security problems arise.

- RADIUS does not provide high transport protection. For example, an observer can examine the content of RADIUS messages and trace the content of a specific attribute.

To overcome these security weakness, it has been proposed the use of TLS (T. Dierks & C. Allen (1999)) to provide a means to secure the RADIUS communication between client and server on the transport layer (S. Winter et al. (2010)). Nevertheless, the main research and standardization efforts have focused on the design of a new AAA protocol called *Diameter*.

2.1.2.2 Diameter

Diameter, proposed as an enhancement to RADIUS, is considered the next generation AAA protocol. Diameter is characterized by its extensibility and adaptability since it is designed to perform any kind of operation and supply new needs that may appear in future control access technologies. Another cornerstone of Diameter is the consideration of multi-domain scenarios where AAA infrastructures administered by different domains are interconnected to provide an unified authentication, authorization and accounting framework. For this reason, Diameter is widely used in 3G networks and its adoption is recommended in future AAA infrastructures supporting access control in NGN.

The Diameter protocol defines an extensible architecture that allows to incorporate new features through the design of the so-called *Diameter applications*, which rely on the basic functionality provided by the *base protocol*. The Diameter *base protocol* (P. Calhoun & J. Loughney (2003)), defines the Diameter minimum elements such as the basic set of messages, attribute structure and some essential attribute types. Additionally, the basic specification defines the inter-realm operations by defining the role of different types of Diameter entities. Diameter applications are services, protocols and procedures that use the facilities provided by the Diameter base protocol itself. Every Diameter application defines its own *commands* and *messages* which, in turn, can define new attributes called *Attribute Value Pair* (AVP) or re-use existing ones already defined by some other applications.

The Diameter base protocol does not define any use of the protocol and expects the definition of specific applications using the Diameter functionality. For example, the use of Diameter for providing authentication during network access is defined in the *Diameter NAS Application* (P. Calhoun et al. (2005)). In turn, this specification is used by the *Diameter EAP Application* (P. Eronen et al. (2005)) to specify the procedure to perform the network access authentication by using the EAP protocol. Similarly, authorization and accounting procedures are expected to be handled by specific applications.

Within a Diameter-based infrastructure, the protocol distinguishes different types of nodes where each one plays a specific role:

1. *Diameter Client*: represents an entity implementing network access control like, for example, a NAS. The Diameter client issues messages soliciting authentication, authorization or accounting services for a specific user.
2. *Diameter Server*: is the entity that processes authentication, authorization and accounting request for a particular domain. The Diameter server must support the Diameter base protocol and the applications used in the domain.

3. *Diameter Agent*: is an entity that processes a request and forwards it to a Diameter server or to another agent. Depending on the service provided, we can distinguish:
   (a) *Relay agents*: which forward messages based on routing-related attributes and routing tables.
   (b) *Proxy agents*: which act as a relay agent that, additionally, may modify the routed message based on some policy.
   (c) *Redirect agents*: instead of routing messages, they inform the sender about the proper way to route the message.
   (d) *Translation agents*: which perform protocol translations between Diameter and other AAA protocols such as RADIUS.

The different types of nodes exchange Diameter messages that carry information. Instead of defining a message type, Diameter uses the concept of *command* to specify the type of function a Diameter message intends to perform. Because the message exchange style of Diameter is synchronous, each command consists of a request and its corresponding answer. Table 1 provides a brief summary of the main Diameter commands defined in the base protocol specification.

Command	**Abbreviation**	**Description**
Capabilities-Exchange- Request /Answer	CER/CEA	Discovery of a peer's identity and its capabilities.
Disconnect-Peer-Request /Answer	DPR/DPA	Used to inform the intention of shutting down the connection.
Re-Auth-Request /Answer	RAR/RAA	Sent to an access device (NAS) to solicit user re-authentication.
Session-Termination-Request /Answer	STR/STA	To notify that the provision of a service to a user has finalized.
Accounting-Request /Answer	ACR/ACA	To exchange accounting information between Diameter client and server.

Table 1. Common Diameter commands

2.2 The Extensible Authentication Protocol (EAP)

The *Extensible Authentication Protocol* (EAP) (B. Aboba et al. (2004)) is a protocol designed by the *Internet Engineering Task Force* (IETF) that permits the use of different types of authentication mechanisms through the so-called *EAP methods* (e.g., based on symmetric keys, digital certificates, etc.). These are performed between an *EAP peer* and an *EAP server*, through an *EAP authenticator* which merely forwards EAP packets back and forth between the EAP peer and the EAP server. From a security standpoint, the EAP authenticator does not take part in the mutual authentication process but acts as a mere EAP packet forwarder.

One of the advantages of the EAP architecture is its flexibility since does not impose a specific authentication mechanism. Additionally, EAP is independent of the underlying wireless access technology, being able to operate in NGNs. Finally, EAP allows an easy integration with existing Authentication, Authorization and Accounting (AAA) infrastructures (B. Aboba et al. (2008) by defining a configuration mode that permits the use of a backend authentication server, which may implement some authentication methods. These advantages have motivated the success of the EAP authentication protocol for network access control in future NGNs.

2.2.1 Components

The EAP protocol consists of request and response messages. Request messages are sent from the authenticator to the peer. Conversely, response messages are sent from the peer to the authenticator. The different messages exchanged during an EAP execution are processed by several components that are conceptually organized in four layers:

- *EAP Lower-Layer*. This layer is responsible for transmitting and receiving EAP packets between the peer and authenticator.
- *EAP Layer*. The EAP layer is responsible for receiving and transmitting EAP packets through the transport layer. The EAP layer not only forwards packets between the EAP transport and peer/authenticator layers, but also implements duplicate detection and packet retransmission.
- *EAP Peer / Authenticator Layer*. EAP assumes that an EAP implementation will support both the EAP peer and the authenticator functionalities. For this reason, based on the code of the EAP packet, the EAP layer demultiplexes incoming EAP packets to the EAP peer and authenticator layers.
- *EAP Method Layer*. An EAP method implements a specific authentication algorithm that requires the transmission of EAP messages between peer and authenticator.

2.2.2 Distribution of the EAP entities

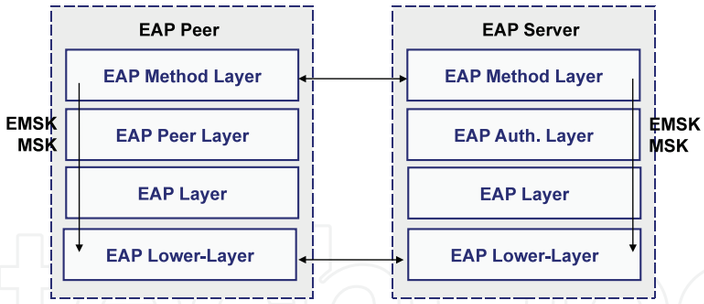
As previously mentioned, an EAP authentication involves three entities: the EAP peer, authenticator and server. Whereas the EAP peer is co-located with the mobile, the EAP authenticator is commonly placed on the *Network Access Server* (NAS) (e.g., an access point or an access router). Depending on the location of the EAP server, two authenticator models have been defined. Figures 2(a) and 2(b) show the *standalone authenticator model* and the *pass-through authenticator model*, respectively. On the one hand, in the standalone authenticator model (Fig. 2(a)), the EAP server is implemented on the EAP authenticator. On the other hand, in the pass-through authenticator model (Fig. 2(b)), the EAP server and the EAP authenticator are implemented in separate nodes.

In order to deliver EAP messages, an *EAP lower-layer* (e.g., IEEE 802.11) is used to transport the EAP packets between the EAP peer and the EAP authenticator. The protocol used to transport messages between the EAP authenticator and the EAP server depends on the authenticator model employed. More precisely, in the standalone authenticator model, the communication between the EAP server and standalone authenticator occurs locally in the same node. In the pass-trough authenticator model, the EAP protocol requires help of an auxiliary AAA protocol such as RADIUS or Diameter.

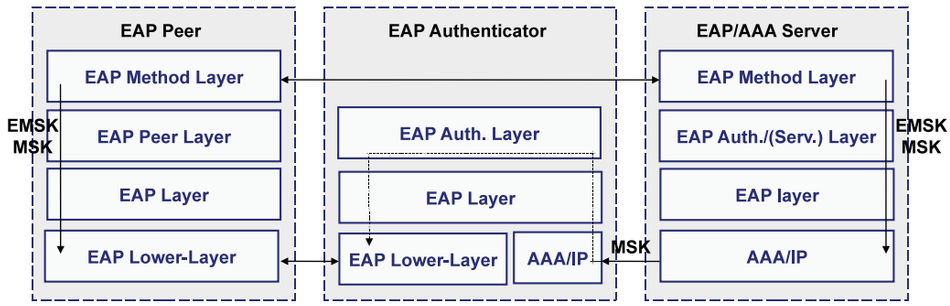
2.2.3 EAP authentication phases

As depicted in Fig. 3, a typical EAP conversation [1] occurs in three different phases. Initially, in the discovery phase (*Phase 0*), the peer discovers the EAP authenticator near to the peer's location with which it desires to start an authentication process. This phase, which is supported by the specific EAP lower-layer protocol, can be performed either manually or automatically.

[1] Without loss of generality, it is assumed an EAP pass-through authenticator model.



(a) Standalone Authenticator Model



(b) Pass-through Authenticator Model

Fig. 2. EAP authenticator models

The authentication phase (*phase 1*) starts when the peer decides to initiate an authentication process with a specific authenticator. This phase consists of two steps. Firstly, the *phase 1a* includes an EAP authentication exchange between the EAP peer, authenticator and server. To start an EAP authentication, the EAP authenticator usually starts the process by requesting the EAP peer's identity through an *EAP Request/Identity* message. The trigger that signals the EAP authenticator to start the EAP authentication is outside the scope of EAP. Examples of these triggers are the *EAPOL-Start* message defined in IEEE 802.1X (IEEE 802.11 (2007)) or simply an 802.11 association process. On the reception of the *EAP Request/Identity*, the EAP peer answers with an *EAP Response/Identity* with its identity. With this information, the EAP server will select the EAP method to be performed. The EAP method execution involves several exchanges of *EAP Request* and *EAP Response* messages between the EAP server and the EAP peer. A successful EAP authentication finishes with an *EAP Success* message.

Certain EAP methods (Dantu et al. (2007)) are able to generate key material. In particular, according to the *EAP Key Management Framework* (EAP KMF) (B. Aboba et al. (2008)) two keys are exported after a successful EAP authentication: the *Master Session Key* (MSK) and the *Extended Master Session Key* (EMSK). The former is traditionally sent (using the AAA protocol) to the authenticator (*Phase 1b*) to establish a security association with the EAP peer (*Phase 2*). Instead, the latter must not be provided to any other entity outside the EAP server and peer. Thus, both entities may use the EMSK for further key derivation. In particular, as we will analyze in Section 4, some authentication schemes propose to employ the EMSK to derive further key material for enabling a fast re-authentication process.

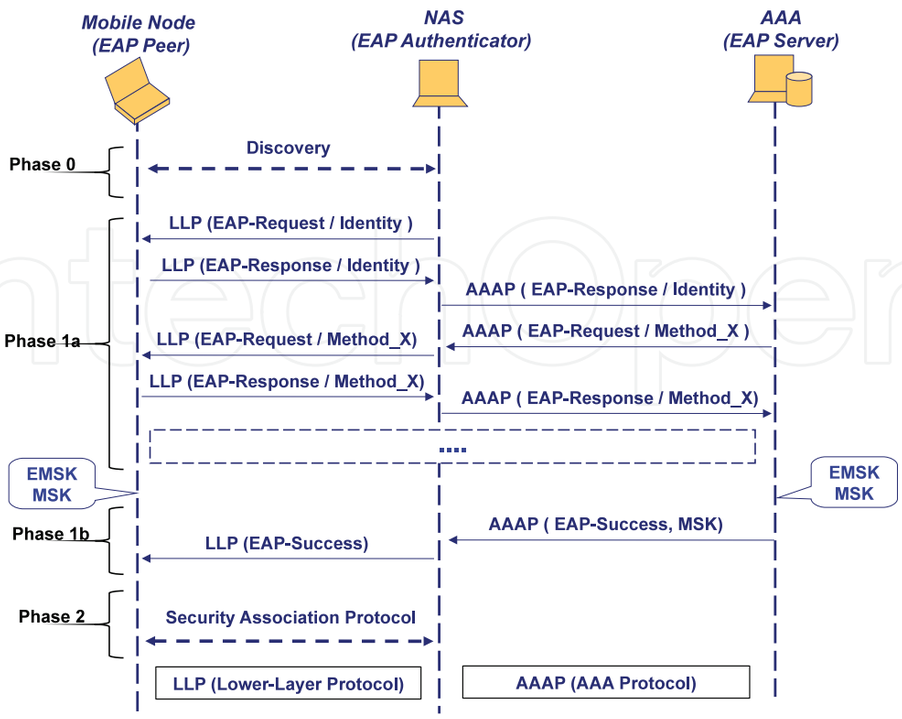


Fig. 3. EAP authentication exchange

2.3 Existing technologies for network access control

The EAP lower-layer protocol allows an EAP peer to perform an EAP authentication process with an authenticator. Basically, the EAP lower-layer is responsible for transmitting and receiving EAP packets between peer and authenticator. Currently, a wide variety of lower-layer protocols can be found since each link-layer technology defines its own transport to carry EAP messages (e.g., IEEE 802.1X, IEEE 802.11, IEEE 802.16e). However, there are also lower-layer protocols operating at network level which are able to transport EAP messages on top of IP (e.g., PANA). Finally, some other lower-layer protocols provide an hybrid solution to transport EAP packets either at link-layer or network layer (e.g., IEEE 802.21 MIH). In the following, the most representative technologies for network access control are analyzed.

2.3.1 IEEE 802.1X

The IEEE 802.1X specification (IEEE 802.1X (2004)) is an access control model developed by the *Institute of Electrical and Electronics Engineers* (IEEE) that allows to employ different authentication mechanisms by means of EAP in IEEE 802 *Local Area Networks* (LANs). As depicted in Fig. 4, there are three main components in the IEEE 802.1X authentication system: *supplicant*, *authenticator* and *authentication server*. In a *Wireless LAN* (WLAN), the supplicant is usually a mobile user, the access point usually represents an authenticator and an AAA server is the authentication server. 802.1X defines a mechanism for port-based network access control. A port is a point through which a supplicant can access to a service offered by a device. The port in 802.1X represents the association between the supplicant and the authenticator. Both the supplicant and the authenticator have a PAE (*Port Access Entity*) that operates the algorithms and protocols associated with the authentication process.

Initially, as depicted in Fig. 4, the authenticator's controlled port is in unauthorized state, that is, the port is *open*. Only received authentication messages will be directed to the authenticator PAE, which will forward them to the authentication server. This initial configuration allows to unauthenticated supplicants to communicate with the authentication server in order to perform an authentication process based on EAP. Once the user is successfully authenticated, the PAE will close the controlled port, allowing the supplicant to access the network service offered by the authenticator's system.

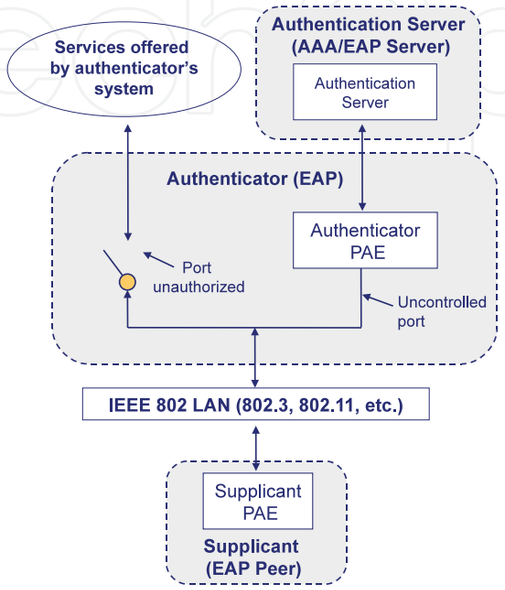


Fig. 4. IEEE 802.1X architecture

2.3.2 IEEE 802.11

IEEE 802.11 extends the IEEE 802.1X access control model by defining algorithms and protocols to protect the data traffic between *station* (STA) and *access point* (AP). More precisely, once the EAP authentication is successfully completed, both STA and AP will share a *Pairwise Master Key* (PMK). This key, derived from the MSK exported by the EAP authentication, is used by a security association protocol (called *4-way handshake*) intended to negotiate cryptographic keys to protect the wireless link between STA and AP. Once the security association is successfully established, the controlled port is closed and access to the network is granted to the supplicant.

The authentication process, described in Fig. 5, involves three entities: an STA acting as supplicant, an AP acting as authenticator and an authentication server (e.g., an AAA server) that assists the authentication process. The process starts with the so-called *IEEE 802.11 association phase* where the STA firstly discovers the security capabilities implemented by the AP *(1)*. Next, the IEEE 802.11 authentication exchange *(2)* is invoked in order to maintain backward compatibility with the IEEE 802.11 state machine. This exchange is followed by an association process *(3)* where the negotiation of the cryptographic suite used to protect the traffic is performed.

In the subsequent *IEEE 802.11 authentication phase*, an EAP authentication is performed where the STA acts as *EAP peer* and the AP acts as *EAP authenticator (4)*. Conversely, the *EAP*

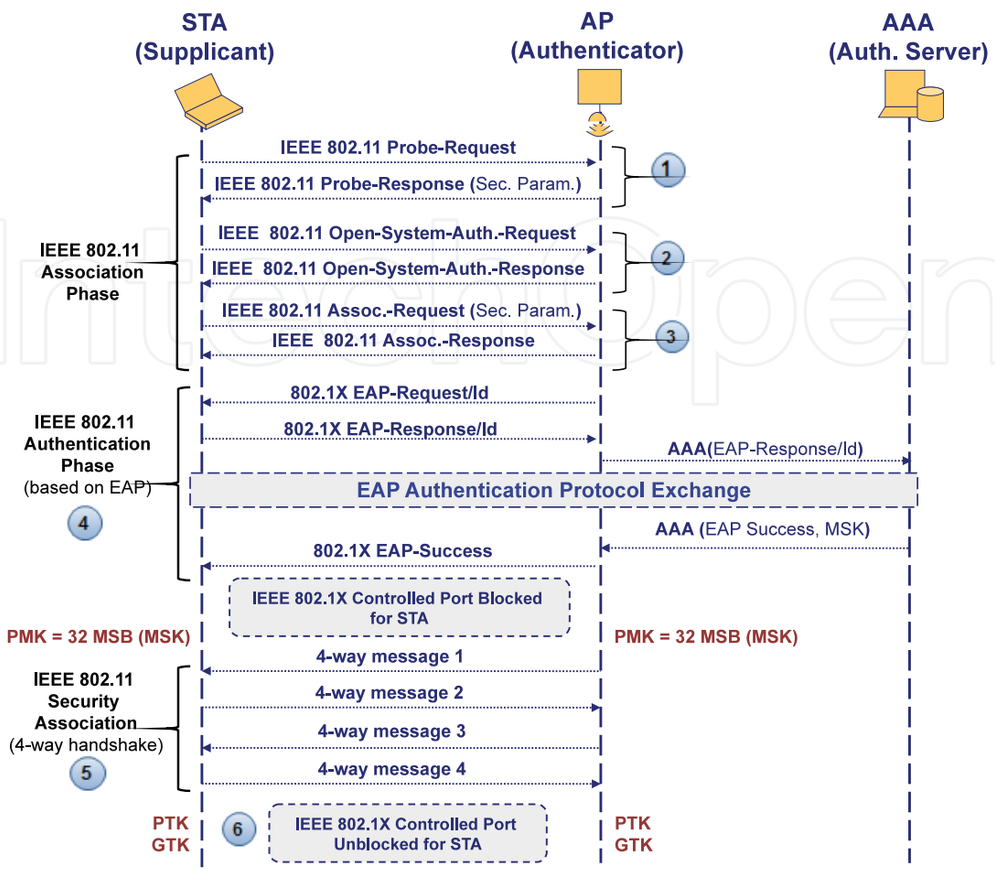


Fig. 5. IEEE 802.11 message flow

server can be co-located with the EAP authenticator (*standalone configuration*) or within an external authentication server (*pass-through configuration*), in which case an AAA protocol (e.g., RADIUS or Diameter) is used to transport EAP messages between the authenticator and the server. Once the EAP authentication is successfully completed, the 32 *more significant bytes* (MSB) from the exported MSK is used as PMK.

Following the establishment of the PMK, a *4-way handshake* protocol is executed during the *IEEE 802.11 security association phase (5)* to confirm the existence of the PMK and selected cryptographic suites. The protocol generates a *Pairwise Transient Key* (PTK) for unicast traffic and a *Group Transient Key* (GTK) for multicast traffic. Thus, as result of a successful *4-way handshake*, a secure communication channel between the STA and the AP is established for protecting data traffic in the wireless link.

2.3.3 IEEE 802.16e

The IEEE 802.16e (*IEEE 802.16e* (2006)) specification is an extension for IEEE 802.16 networks that enables the mobility support and enhances the basic access control mechanism defined for fixed scenarios in order to provide authentication and confidentiality in IEEE 802.16-based wireless networks. In particular, the security architecture is further strengthened by introducing the *Privacy and Key Management* protocol version 2 (PKMv2) which provides mutual authentication and secure distribution of key material between the IEEE 802.16

subscriber station (SS) and the *base station* (BS). The authentication can be performed by using an EAP-based authentication scheme.

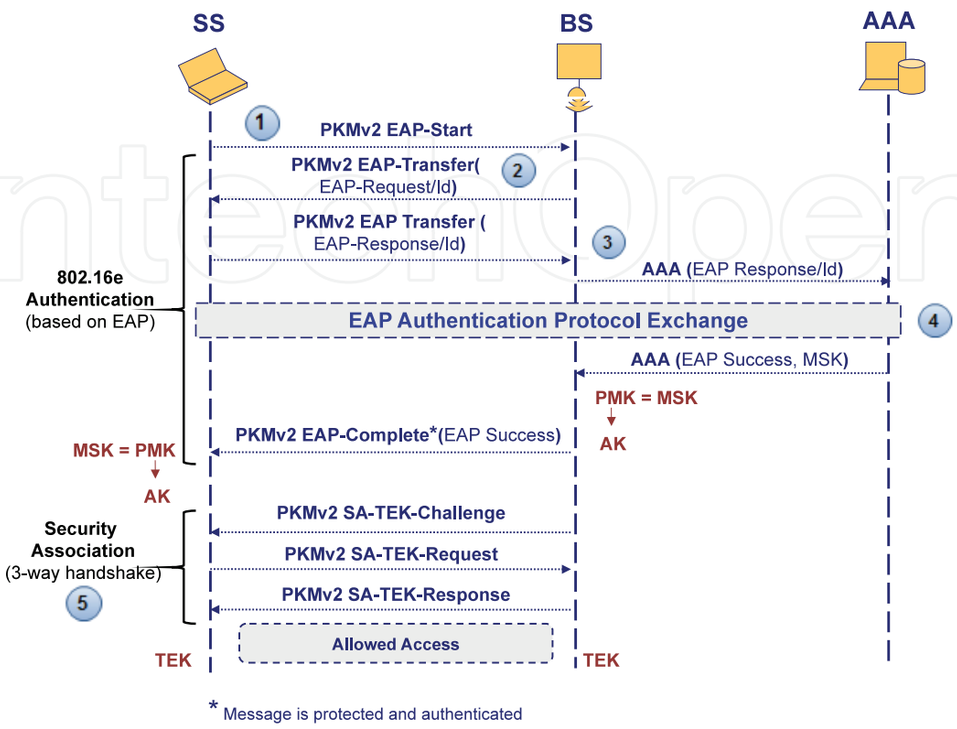


Fig. 6. IEEE 802.16e message flow

Figure 6 shows the authentication process. As observed, while the SS acts as *EAP peer*, the BS implements the *EAP authenticator* functionality. Depending on the EAP configuration mode, the *EAP server* can be placed in the BS (*standalone mode*) or in a AAA server (*pass-through*), which is the case assumed in Fig. 6. As observed, while EAP messages exchanged between SS and BS are transported within the *PKMv2 EAP-Transfer* message, an AAA protocol (e.g., RADIUS or Diameter) is used to convey EAP messages between the BS and the AAA server.

Once the EAP authentication is successfully completed, from the exported MSK a *Pairwise Master Key* (PMK) is derived. In turn, from this PMK, an *Authorization Key* (AK) is generated for the security association establishment. For this reason, the 802.16e specification requires the use of EAP methods exporting key material. Finally, as previously mentioned, the AK shared between SS and BS is employed by a security association protocol called *3-way handshake (5)*, which verifies the possesion of the AK and generates a *Traffic Encryption Key* (TEK) used to protect the traffic in the wireless link.

2.3.4 PANA

The *Protocol for carrying Authentication for Network Access* (PANA) (D. Forsberg et al. (2008)) is a network-layer transport for authentication information designed by the IETF *PANA Working Group* (PANA WG). PANA is designed to carry EAP over UDP to support a variety of authentication mechanisms for network access (thanks to EAP) as well as a variety of underlying network access technologies (thanks to the use of UDP). As highlighted in Fig. 7, PANA considers a network access control model integrated by the following entities:

- The *PANA Client* (PaC) is the client implementation of PANA. This entity resides on the subscriber's node which is requesting network access. The PaC acts as EAP peer according to the EAP model described earlier.
- The *PANA Authentication Agent* (PAA) is the server implementation of PANA. A PAA is in charge of communicating with the PaCs for authenticating and authorizing them to access the network service. The PAA acts as EAP authenticator.
- The *Enforcement Point* (EP) refers to the entity in the access network in charge of inspecting data traffic of authenticated and authorized subscribers. Basically, the EP represents a point of attachment (e.g., access point) to the network.
- The *Authentication Server* (AS) is in charge of verifying the credentials provided by a PaC through a PAA. The AS functionality is typically implemented by an AAA server, which also integrates the EAP server.

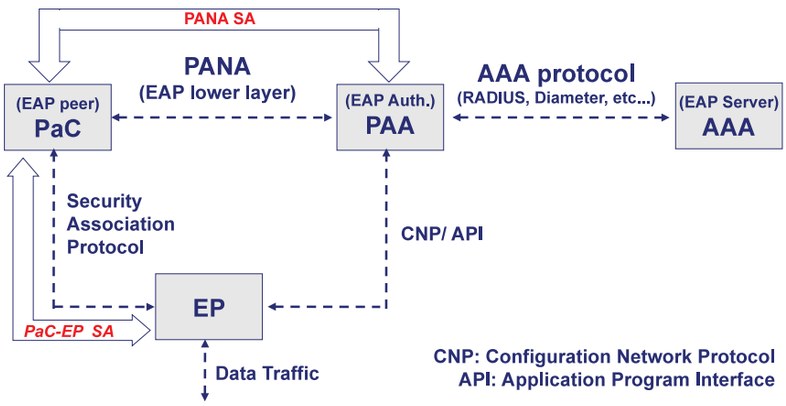


Fig. 7. PANA architecture

Additionally, there are two types of security associations related to PaC in the PANA architecture. On the one hand, a *PANA security association* (PANA SA) is established between the PaC and PAA in order to integrity protect PANA messages. On the other hand, a *PaC-EP SA* is established by performing a security association protocol between the PaC and an EP to protect data traffic.

The PANA operation is developed along four different phases. Initially, during the *authentication and authorization phase*, the PaC and the PAA negotiate some parameters, such as the integrity algorithms used to protect PANA messages. They also exchange PANA messages transporting EAP to perform the authentication and establish a so-called *PANA session*. If the PaC is successfully authenticated, the protocol enters in the *access phase* where the PaC can use the network service by just sending data traffic through the EP. If the PANA session is about to expire, typically a *re-authentication phase* happens to renew this session lifetime. Finally, the PaC or PAA can terminate the session (e.g., the PaC desires to log out the network access session) during *termination phase*, where resources allocated by the network for the PaC are also removed. If neither PaC nor PAA can complete the termination phase, both entities can release the resources once the PANA session lifetime expires.

During each phase, a different set of messages can be sent. Basically we can find four types of PANA messages.

- *PANA-Client-Initiation* (PCI). This message is sent by the PaC requesting the PAA start the authentication process.

- *PANA-Auth-Request/Response* (PAR/PAN). These messages are used during the authentication and authorization phase and the re-authentication phase. They allow to negotiate some parameters between the PaC and the PAA and to carry authentication information in the format of EAP packets.
- *PANA-Notification-Request/Response* (PNR/PNA). These messages are exchanged once PaC is authenticated. They are used as keep-alive mechanism of the PANA authentication session or to signal the beginning of a re-authentication process.
- *PANA-Termination-Request/Response* (PTR/PTA). These messages are used to end up a PANA session.

2.3.5 IEEE 802.21 MIH

The IEEE 802.21 is a recent effort that aims at enabling seamless service continuity among heterogeneous networks (IEEE 802.21 (2008); Taniuchi et al. (2009)). The standard defines a logical entity, *MIH Function* (MIHF), which facilitates the mobility management and handover process. The MIHF is located within the mobility management protocol stack of a mobile node (MN) or network entity. Through the media independent interface, MIHF supports useful services (events, commands or information) that help in determining the need for initiate a handoff or selecting a candidate network

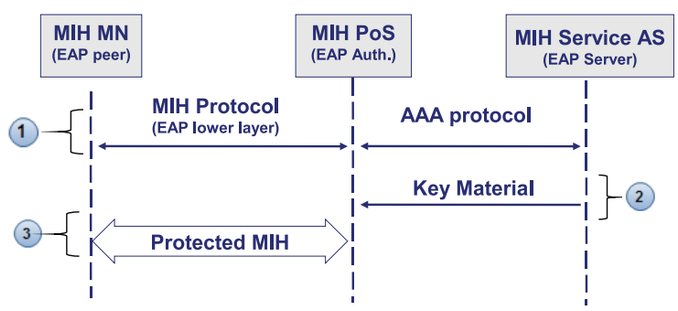


Fig. 8. MIH protocol as EAP lower-layer

Different *tasks groups* (TG) have defined extensions to IEEE 802.21. For example, the standardization task group IEEE 802.21a is defining mechanisms that allow to protect the IEEE 802.21 MIH protocol messages. The solution (EAP over MIH (2010)) designed by the task group proposes that the *mobile node* (MN) must be authenticated and authorized before granting access to the services offered by the *Point of Service* (PoS). In particular, EAP has been proposed as one alternative to carry out this authentication process. Figure 8 depicts the general process followed to perform an EAP-based *Media-Independent Authentication Process*. As observed, the MN and PoS acts as EAP peer and authenticator, respectively. The EAP server functionality is implemented by an entity named *Service Authentication Server* (Service AS). Initially, an EAP authentication (*1*) is performed between the MN and the Service AS through the PoS, which acts as authenticator. While the MIH protocol is used as EAP lower-layer to transport EAP messages between MN and PoS, an AAA protocol is employed between PoS and Service AS for the same purpose. Note that, since MIH protocol is independent from the underlying transport, this is an hybrid solution that can operate either at link-layer or network-layer. When the EAP authentication is completed, the Service AS sends the MSK (*2*) exported by the EAP method to the PoS. From this MSK, a key hierarchy is generated to protect MIH protocol packets (*3*).

3. Fast re-authentication to optimize the network access control

As we can observe, EAP is a promising authentication protocol to be used in NGNs due to its flexibility, wireless technology independence and integration with AAA infrastructures. Furthermore, it is used by a wide variety of network access technologies as standard solution for authentication. However, EAP has shown some drawbacks when mobility is taken into consideration. The reason why the EAP authentication process is not so optimized for mobile scenarios is due to two main motives. First, a typical EAP authentication requires several message exchanges between EAP peer and server. Depending on the EAP method in use (R. Dantu et al. (2007)), this number can vary. For example, one of the most common methods, EAP-TLS (D. Simon et al. (2008)), involves in the best case up to eight messages between peer and server to complete. Secondly, each round-trip is performed with the EAP server placed on the EAP peer's home domain, where the peer is subscribed to. Especially in roaming scenarios, the EAP server may be far from the mobile user (EAP peer) and, therefore, the latency introduced per each exchange increases. These issues are raised when an EAP peer moves from one authenticator to another (*inter-authenticator handoff*). In this case, the peer needs to perform an EAP authentication with the EAP server, through the new EAP authenticator. Therefore, every time the EAP peer moves to a new EAP authenticator, it may suffer from high handoff latency during EAP authentication.

This problem can affect the on-going communications since the latency introduced by the EAP authentication during the handoff process may provoke a substantial packet loss, resulting in a degradation in the service quality perceived by the user. In this sense, the performance requirements of a real-time application will vary according to the type of application and its characteristics such as delay and packet-loss tolerance. The ITU-T G.114 recommendation (ITU-T Recommendation G.114 (1998)) indicates, for Voice over IP applications, an end-to-end delay of 150 ms as the upper limit and rates 400 ms as a generally unacceptable delay. Similarly, a streaming application has tolerable packet-error rates ranging from 0.1 to 0.00001 with a transfer delay of less than 300 ms. As has been proved in (R. M. Lopez et al. (2007)), a full EAP authentication[2] based on a typical EAP method such as EAP-TLS can provoke an unacceptable handoff interruption of about 600 milliseconds (or even in some cases several seconds) for these kind of applications.

To solve this problem, it is necessary to define a *fast re-authentication process* (T. Clancy et al. (2008)) to reduce the authentication time required by a user to complete an EAP-based authentication. Researchers have not ignored this challenging aspect and a wide set of fast re-authentication mechanisms can be found in the literature. Before analyzing the different fast re-authentication schemes in next Section 4, we are going to present both the desired design and security goals that a proper fast re-authentication mechanism should accomplish. To be aware of these requirements is useful to determine advantages and disadvantages when analyzing the different fast re-authentication solutions.

3.1 Design goals

A suitable fast re-authentication solution should accomplish the following requirements and aims (T. Clancy et al. (2008)):

[2] Note that the term *full* is used in comparison with *reduced* to denote that, in the execution of an EAP method, there is no optimization to reduce the number of exchanges during the EAP authentication.

(D1) *Low latency operation*. The fast re-authentication mechanism must reduce the authentication time executed during the network access control process compared with a traditional full EAP authentication. Furthermore, the achievement of a reduced handoff latency must not affect the security of the authentication process.

(D2) *EAP lower-layer independence*. Any keying hierarchy and protocol defined must be independent of the lower-layer protocol used to transport EAP packets between the peer and the authenticator. In other words, the fast re-authentication solution must be able to operate over heterogeneous technologies, which is the expected scenario in NGNs. Nevertheless, in certain circumstances, the fast re-authentication mechanism could require some assistance from the lower layer protocol.

(D3) *Compatibility with existing EAP methods*. The adoption of a fast re-authentication solution must not require modifications to existing EAP methods. In the same manner, additional requirements must not be imposed on future EAP methods. Nevertheless, the fast re-authentication solution can enforce the employment of EAP methods following the *EAP Key Management Framework* (B. Aboba et al. (2008)).

(D4) *AAA protocol compatibility and keying*. Any modification to the EAP protocol itself or the key distribution scheme defined by EAP, must be compatible with currently deployed AAA protocols. Extensions to both RADIUS and Diameter to support these EAP modifications are acceptable. However, the fast re-authentication solution must satisfy the requirements for the key management in AAA environments (B. Aboba et al. (2008); R. Housley & B. Aboba (2007)).

(D5) *Compatibility with other optimizations*. The fast re-authentication solution must be compatible with other optimizations destined to reduce the handoff latency already defined by other standards.

(D6) *Backward compatibility*. The system should be designed in such a manner that a user not supporting fast re-authentication should still function in a network supporting fast re-authentication. Similarly, a peer supporting fast re-authentication should still operate in a network not supporting the fast re-authentication optimization.

(D7) *Low deployment impact*. In order to support the aforementioned design goals, a fast re-authentication solution may require modifications in EAP peers, authenticators and servers. Nevertheless, in order to favour the protocol deployment, the required changes must be minimized (ideally, they should be avoided) in current standardized protocols and technologies.

(D8) *Support of different types of handoffs*. The fast re-authentication mechanism must be able to operate in any kind of handoff regardless of whether it implies a change of technology (intra/inter-technology), network (intra/inter-network), administrative domain (intra/inter-domain) or type of security required by the authenticator (intra/inter-security).

3.2 Security goals

In addition to the aforementioned design goals, a secure fast re-authentication mechanism should accomplish the following security goals (R. Housley & B. Aboba (2007)):

(S1) *Authentication*. This requirement mandates that a management and key distribution mechanism must be designed to allow all parties involved in the protocol execution to authenticate every entity with which it is communicating. That is, it must be feasible to

gain assurance that the identity of the another entity is as declared, thereby preventing impersonation. To carry out the authentication process, it is necessary to define the so-called *security associations* between the involved entities.

(S2) *Authorization*. During the network access control process, the user is not only authenticated but also authorized to access the network service. The authorization decision is taken by the AAA server and the result is communicated to the authenticator. The fast re-authentication solution proposed must not hinder the authorization process performed once the user is successfully authenticated.

(S3) *Key context*. This requirement establishes that any key must have a well-defined scope and must be used in a specific context for an intended use (e.g., cipher data, sign, etc.). During the time a key is valid, all the entities that are authorized to have access to the key must share the same key context. In this sense, keys should be uniquely named so that they can be identified and managed effectively. Additionally, it must be taken into account that the existence of a hierarchical key structure imposes some additional restrictions. For example, the lifetime of lower-level keys must not exceed the lifetime of higher-level keys.

(S4) *Key freshness*. A key is fresh (from the viewpoint of one party) if it can be guaranteed to be recent and not an old key being reused for malicious actions by either an attacker or unauthorized party (A. Menezes et al. (1996)). Mechanisms for refreshing keys must be provided within the re-authentication solution.

(S5) *Domino effect*. In network security, the compromise of keys in a specific level must not result in compromise of other keys at the same level or higher levels that were used to derive the lower-level keys. Assuming that each authenticator is distributed a key to carry out the fast re-authentication process, a key management solution respecting this property will be resilient against the *domino effect* (R. Housley & B. Aboba (2007)) attack, so the compromise of one authenticator must not reveal keys in another authenticators.

(S6) *Transport aspects*. The solution developed must be independent of any underlying transport protocol. Depending on the physical architecture and the functionality of the involved entities, there may be a need for multiple protocols to perform the transport of keying material between entities involved in the fast re-authentication architecture. As far as possible, protocols already designed and used should be used to address the cryptographic material distribution. For example, while AAA protocols can be considered for this purpose between the EAP authenticator and server, the EAP protocol can be used between EAP peer and server.

4. Overview of existing fast re-authentication schemes

This section analyzes the different efforts that have attempted to reduce the EAP authentication time during the network access control process. According to the strategy followed to achieve this objective, the different fast re-authentication solutions can be classified in different groups: *context transfer*, *pre-authentication*, *key pre-distribution*, *use of a local server* and *modifications to EAP*. In the following, we delve into each of them and detail the mechanism proposed to achieve a reduced handoff latency.

4.1 Context transfer

As depicted in Fig. 9, the context transfer mechanism (T. Aura & M. Roe (2005), H. Kim et al. (2005), C. Politis et al. (2004), *IEEE 802.11 IAPP* (2003), J. Bournelle et al. (2006)) tries

to reduce the time devoted to network access control by transferring cryptographic material (*1*) from an EAP authenticator (*current*) to a new one (*target*). When the user moves to the new authenticator (*2*), it can use the transferred context (e.g., cryptographic keys and associated lifetimes) to execute a security association protocol with the new authenticator (*3*) to protect the wireless link. Thus, the user does not need to be authenticated and can directly start the security association establishment process based on the transferred cryptographic material.

In order to perform a secure transference between both authenticators, it is assumed the existence of a pre-established security association between them. Additionally, context transfer solutions do not propagate the same cryptographic material (*CM*) from one authenticator to another. Instead, the transferred cryptographic material is derived (*CM'*) from that owned by the current authenticator where the user is connected. The process employed to generate the derived cryptographic material is followed by both the peer and the authenticator. While the authenticator transfers the derived material to the new authenticator, the peer employs it to start the security protocol execution.

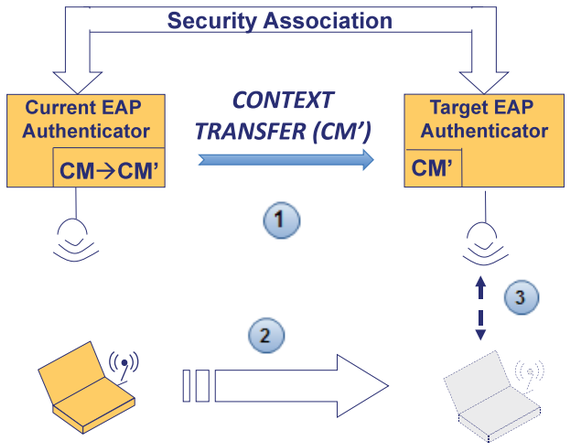


Fig. 9. Context transfer mechanism

Depending on when the transference is performed, we can distinguish between *reactive* and *proactive* schemes. In the proactive mode, the context transfer is performed before the peer performs the handoff. Therefore, when the peer moves to the new authenticator, the cryptographic material has been already transferred to the new authenticator and the peer can immediately establish the security association. Conversely, in the reactive mode, the context transfer is performed once the user performs the handoff and is under the coverage area of the new authenticator. The proactive mode introduces less latency to network access control than the reactive mode since the transference of cryptographic material is performed in advance before the handoff. Nevertheless, reactive solutions are interesting in situations where the handoff happens unexpectedly and there is no anticipation to perform the transference.

An important advantage of context transfer mechanisms relies on their ability to re-authenticate the user without the need of contacting an authentication server located in the infrastructure. Nevertheless, they have been widely criticized as a promising technique to achieve a fast network access due to an important security vulnerability known as the *domino effect* (R. Housley & B. Aboba (2007)). The problem comes from the fact that context transfer re-uses the same cryptographic material (or a derived one following a well-known process) in different authenticators. Therefore, if one authenticator is compromised, the rest of authenticators visited by the same user are also affected.

4.2 Pre-authentication

Pre-authentication solutions propose a scheme (see Fig. 10) where the mobile user performs a full EAP authentication (*1*) with a candidate authenticator through the current associated one *before* it performs the handoff. In this manner, when the handoff happens (*2*), given that the MSK generated during the pre-authentication process will be already present in the candidate authenticator, the peer only needs to establish a security association (*3*) with it to protect the wireless link. As we see, pre-authentication decouples the authentication and network access control operations from the handoff.

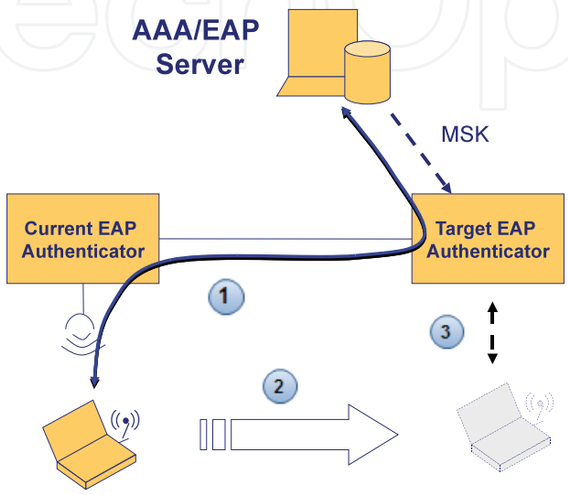


Fig. 10. Pre-authentication mechanism

Depending on the role adopted by the current authenticator during the EAP pre-authentication, we can distinguish two scenarios of EAP pre-authentication signalling (Y. Ohba et al. (2010)):

- *Direct pre-authentication*. In this type of EAP pre-authentication, the current authenticator only forwards the EAP lower-layer messages between mobile node and candidate authenticator as it would be data traffic.
- *Indirect pre-authentication*. Here, the current authenticator plays an active role during pre-authentication process. This type of pre-authentication is useful when the mobile node neither has the candidate authenticator address nor is able to access to the candidate authenticator for security reasons. Therefore, there is a signalling from mobile node to/from current authenticator, and from/to the current authenticator to/from the candidate authenticator. Note that current authenticator does not act as an EAP authenticator; it only translates between different EAP lower-layer protocols.

The first pre-authentication proposal was initially introduced at link layer by the IEEE 802.11i technology (IEEE 802.11i (2005)) and later improved in IEEE 802.11r (IEEE 802.11r (2005)). Nevertheless, the definition of pre-authentication mechanisms at link-layer has some serious limitations since they cannot be applied for cases involving inter-domain or inter-technology handoffs. To avoid this problems, some other solutions propose a pre-authentication procedure at network layer. Network layer solutions (Y. Ohba and A. Yegin (2010), R. M. Lopez et al. (2007), A. Dutta et al. (2008)) have the advantage of being capable to work independent of the underlying access technologies and with authenticators located in different networks or domains.

Despite pre-authentication solutions can potentially achieve an important reduction in the latency introduced by the authentication process during the network access control, this technique presents some drawbacks. First, pre-authentication requires the existence of network connectivity to carry out the pre-authentication process which is a requisite that may not always be satisfied. Second, pre-authentication requires a precise selection of the authenticator with which perform a pre-authentication process. If the user performs a pre-authentication with authenticators where the user finally does not move, the technique may incur in an unnecessary use of network resources. The third disadvantage is related to the previous one. Since pre-authentication implies the pre-reservation of resources in candidate authenticators, in practice, operators are reluctant to pre-reserve resources for users that may or may not roam in the future. Therefore, pre-authentication may have a limited application, specially in inter-domain handoffs. Finally, given that pre-authentication involves a full EAP authentication, special care must be taken to determine the moment to start the pre-authentication process. As a consequence, pre-authentication needs to be performed with a considerable anticipation to the handoff.

4.3 Key pre-distribution

Key pre-distribution solutions (A. Mishra et al. (2004), S. Pack & Y. Choi (2002), Z. Cao et al. (2011), F.Bernal-Hidalgo et al. (2011)) propose the pre-installation of cryptographic material (e.g., keys) in candidate authenticators so that the keys required for secure association are already available when the peer moves to the authenticators. As depicted in Fig. 11, the mobile user initially performs an EAP authentication (*1*) with the AAA server. Once the EAP authentication is successfully completed, the AAA server pre-distributes keys (*2*) to authenticators which the user can potentially associate to in a near future. Therefore, when the peer moves to a new authenticator (*3* and *5*), it is not required to perform a full EAP authentication. Instead, using the key material already present in the authenticator and known by the peer, a security association is established between both entities (*4* and *6*).

Fast re-authentication solutions based on key pre-distribution have two main disadvantages. On the one hand, they require a precise selection of those authenticators to which pre-distribute key material. If the user pre-distributes key material to authenticators where the user finally does not move, the technique may incur in an unnecessary use of resources. Nevertheless, this is a complex problem given the difficulty of predicting future movements of the user. On the other hand, key pre-installation solutions have a significant deployment cost since a modification in existing lower-layer technologies and AAA protocols is required in order to allow pushing a key provided by an external entity instead of being produced as a consequence of a successful EAP authentication executed through the EAP authenticator.

4.4 Use of a local server

According to the EAP authentication model (B. Aboba et al. (2004)), each time a user needs to be authenticated, a full EAP authentication must be performed with the AAA/EAP server located in the user's home domain. This is a serious limitation for roaming scenarios, specially in mobility contexts. The reason is that each time the visited network needs to re-authenticate the client, the home domain must be contacted. This introduces a considerable latency during network access process since the home EAP server could be located far from the current user's location. Furthermore, taking into account that typical EAP methods (e.g., EAP-TLS) require multiple round trips, the home domain needs to be contacted several times in order to complete the EAP conversation, resulting in unacceptable handoff times.

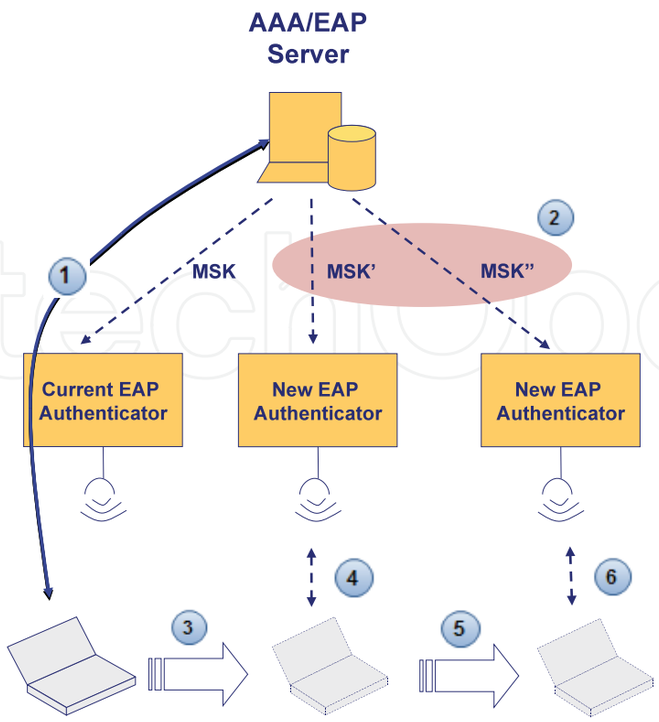


Fig. 11. Key pre-distribution mechanism

To solve this issue, some solutions (3GPP TS 33.102 V7.1.0 (2006), R. Marin et al. (2006), F.Bernal-Hidalgo et al. (2011), V. Narayanan & L. Dondeti (2008)) have proposed the use of a local server near the area of movement of the peer to speed up the re-authentication. The basic idea is to allow the visited domain to play a more active role in network access control by allowing the home AAA server to delegate the re-authentication task to the local AAA server placed in the visited domain. As depicted in Fig. 12, the user firstly performs a full EAP authentication (*1*) with the home AAA/EAP server using the *long-term* credentials that the home domain provides to their subscribers. This initial EAP authentication, commonly named *bootstrapping phase*, is performed the first time the user connects to the network. Next, once the EAP authentication is successfully completed, the home AAA/EAP server sends (*2*) some key material (KM) to the visited AAA/EAP server. This key material, which is used as *mid-term* credential between the mobile and the visited AAA/EAP server, allows to locally perform re-authentication (*3, 4*) when the peer moves to other authenticators located in the visited domain, thus avoiding AAA signalling with the home AAA/EAP server.

Despite this kind of fast re-authentication solutions do not require to contact the home domain to re-authenticate the user, they do not define any optimization for the re-authentication process with the local server. For example, authors in (R. Marin et al. (2006)) propose the use of an EAP method based on shared secret key like EAP-GPSK which requires two message exchanges with the local authentication server. Another serious disadvantage is found in the process followed to distribute the key that establishes a trust relationship between the peer and the local server. Solutions like (F.Bernal-Hidalgo et al. (2011); R. Marin et al. (2006)) use a two-party model to carry out a key distribution process which involves three entities: peer, local re-authentication server and home AAA/EAP server. Since the use of a two-party model is known to be inappropriate (D. Harskin et al. (2007)) from a security standpoint, a three-party approach is recommended.

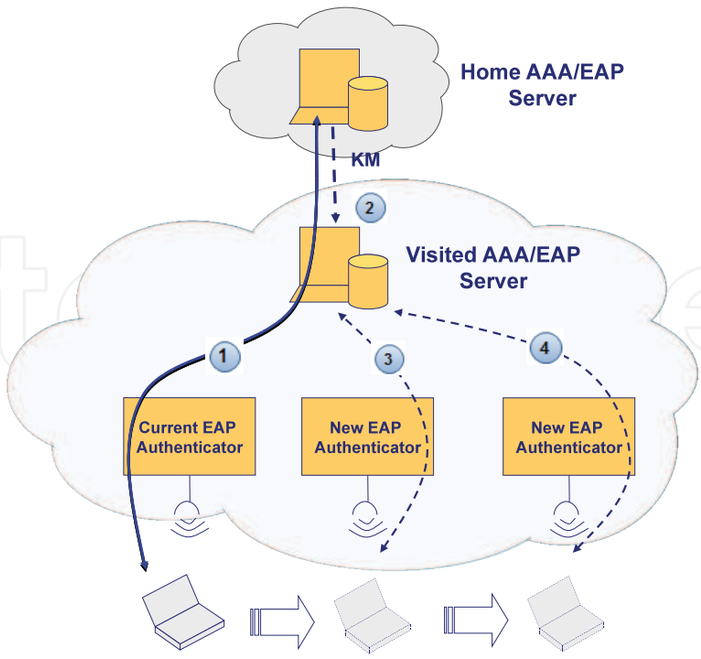


Fig. 12. Use of a local server mechanism

4.5 Modifications to EAP

Finally, another group of solutions try to reduce the EAP authentication time by modifying the EAP protocol itself. Between the different solutions following this approach, the most relevant contribution is the *EAP Extensions for EAP Re-authentication Protocol* (ERP) (V. Narayanan & L. Dondeti (2008)), which has been proposed by the IETF *HandOver KEYing Working Group* (HOKEY WG).

ERP is a method-independent solution that modifies the EAP protocol to achieve a lightweight authentication process. Additionally, ERP relies on the local server optimization (see Section 4.4) and assumes the existence of a local *EAP Re-authentication* (ER) server to optimize the process, which will be in charge of both fast EAP re-authentication and key distribution tasks. The ERP protocol describes a set of extensions to EAP in order to enable efficient re-authentication for a peer that has already established some EAP key material with the EAP server in a previous *bootstrapping phase*. These extensions include three new messages: *EAP-Initiate/Re-auth-Start*, *EAP-Initiate/Re-auth* and *EAP-Finish/Re-auth*.

As shown in Fig. 13, the ERP negotiation involves the peer, the authenticator and the ER server. Beforehand, it is assumed that the peer performs a full EAP authentication with the ER server and both entities share a EMSK. From the EMSK, the peer and the ER server derives a key named rRK. In turn, from the rRK, a new key named *Re-authentication Integrity Key* (rIK) is derived to provide proof of possession and authentication during the re-authentication process.

The ERP re-authentication process is initiated by the authenticator by sending *EAP-Initiate/Re-auth-Start* to the peer. On the reception of this message, the peer sends an *EAP-Initiate/Re-auth* protected with the rIK which is forwarded by the authenticator to the ER server. Once the ER server successfully verifies this messages, it

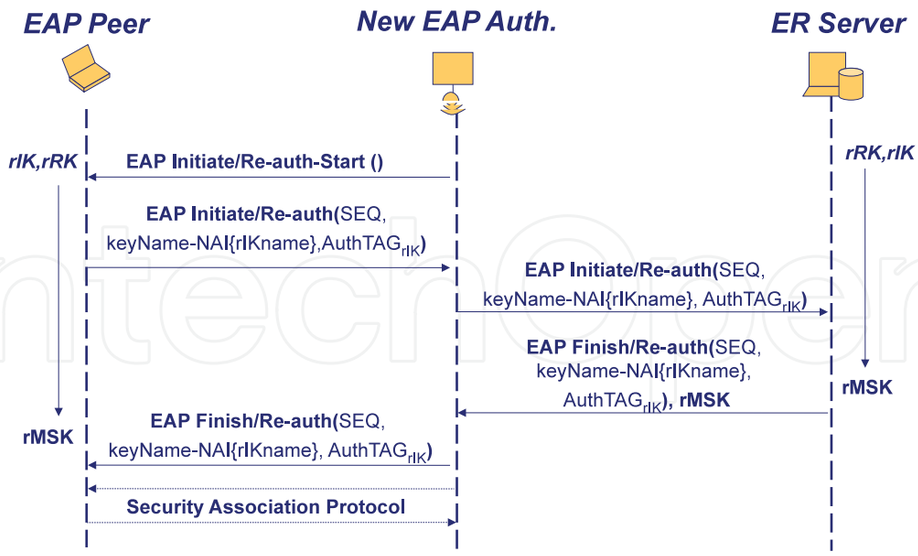


Fig. 13. ERP protocol

replays with a final *EAP-Finish/Re-auth* and derives a rMSK (from the rRK), which is sent to the authenticator to establish a security association with the peer.

On the one hand, in general, the main problem of this kind of proposals relies on their high deployment cost. Since these solutions update the EAP protocol basic operation, they require the modification of existing EAP implementations in order to support the new re-authentication functionality. Consequently, user equipments, authenticators and authentication servers need to be updated, thus complicating the adoption of the solution. On the other hand, in particular, an important drawback of ERP is found on the security of the re-authentication process. Similarly to solutions (F.Bernal-Hidalgo et al. (2011); R. Marin et al. (2006)) previously analyzed in Section 4.4, ERP follows an inappropriate two-party key distribution model to distribute the rMSK from the ER to the authenticator.

5. Conclusion

The provision of *seamless mobility* has created an interesting research field within NGNs in order to find mechanisms which try to provide a continuous access to the network during the handoff. In fact, this is a critical process, where the connection to the network is interrupted, thus causing packet loss that may affect on-going communications. To solve this problem, efforts are directed at reducing the time required to complete the different tasks performed during the handoff. In particular, the network access control process has been demonstrated to be one of the most important factors that negatively affects handoff latency. This process is demanded by network operators in order to control that only legitimate users are able to employ the operator's resources.

This chapter has provided a general overview about the state-of-art of technologies and protocols related to network access control in future NGNs. In particular, we have reviewed the EAP/AAA framework as a promising architecture for network access authentication in future heterogeneous networks. While AAA infrastructures provide an unified framework to handle the authentication, authorization and accounting processes, the EAP protocol is used to implement the authentication service in AAA scenarios. Apart from being easily

deployable within existing AAA infrastructures, EAP exhibits important features such as flexibility to select an authentication mechanism and independence from the underlying wireless technology.

Nevertheless, EAP presents some deficiencies when applied in mobile scenarios. In particular, a typical EAP authentication introduces a prohibitive latency during the handoff which provokes a connection disruption that may affect active communications. This problem has been extensively studied by the research community, which has proposed different fast re-authentication mechanisms.

Precisely, the second part of the chapter is devoted to revise and analyze the different schemes that have tried to reduce the latency introduced by network access control during the handoff. According to the strategy followed to reduce the authentication time, we can distinguish five fast re-authentication schemes: context transfer, pre-authentication, key pre-distribution, use of a local server and modifications to EAP. Throughout this chapter we have analyzed both advantages and disadvantages of each approximation.

6. Acknowledgements

This work is partially supported by the Funding Program for Research Groups of Excellence (04552/GERM/06) and the Spanish Ministry of Science and Education (TIN2008-06441-C02-02).

7. References

3GPP TS 33.102 V7.1.0 (2006). 3rd Generation Partnership Project.

A. Dutta, D. Famolari, S. Das, Y. Ohba, V. Fajardo, K. Taniuchi, R. Lopez & H. Schulzrinne (2008). *Media-Independent Pre-Authentication Suppporting Secure Interdomain Handover Optimization*, *IEEE Wireless Communications* vol. 15(2): 55–64.

A. Menezes, P. van Oorschot & S. Vanstone (1996). Handbook of Applied Cryptography, CRC Press.

A. Mishra, M. Shin, N. Petroni, C. Clancy & W. Arbaugh (2004). *Proactive Key Distribution Using Neighbor Graphs*, *IEEE Wireless Communication* 11: 26–36.

B. Aboba, D. Simon & P. Eronen (2008). *Extensible Authentication Protocol Key Management Framework*. RFC 5247.

B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson & H. Levkowetz (2004). *Extensible Authentication Protocol (EAP)*. RFC3748.

Badra, M., Urien, P. & Hajjeh, I. (2007). *Flexible and fast security solution for wireless LAN*, *Pervasive and Mobile Computing Journal* 3: 1–14.

C. de Laat, G. Gross, L. Gommans, J. Vollbrecht & D. Spence (2000). *Generic AAA Architecture*. IETF RFC 2903.

C. Politis, K. Chew, N. Akhtar, M. Georgiades, R. Tafazolli & T. Dagiuklas (2004). *Hybrid multilayer mobility management with AAA context transfer capabilities for all-IP networks*, *IEEE Wireless Communications 11* pp. pp. 76–88.

C. Rigney, S. Willens, A. Rubens & W. Simpson (2000). *Remote Authentication Dial In User Service (RADIUS)*. IETF RFC 2865.

D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig & A. Yegin (2008). *Protocol for Carrying Authentication for Network Access (PANA)*. IETF RFC 5191.

D. Harskin, Y. Ohba, M. Nakhjiri & R. Marin (2007). *Problem Statement and Requirements on a 3-Party Key Distribution Protocol for Handover Keying*. IETF Internet Draft, draft-ohba-hokey-3party-keydist-ps-01.

D. Simon, B. Aboba & R. Hurst (2008). *The EAP-TLS Authentication Protocol*. IETF RFC 5216.

Dantu, R., Clothier, G. & Atri, A. (2007). EAP Methods for Wireless Networks, *Computer Standards Interfaces* 29(3): 289–301.

EAP over MIH (2010). Option III: EAP to conduct service authentication and MIH packet protection (21-10-0078-08-0sec-option-iii-eap-over-mih-service-authentication).

F.Bernal-Hidalgo, Marin-Lopez, R. & Gomez-Skarmeta, A. (2011). *Key Distribution Mechanisms For IEEE 802.21-Assisted Wireless Heterogeneous Networks*, *Mobile Networks and Management*, Vol. 68, Springer Berlin Heidelberg, pp. 123–134.

H. Kim, K. G. Shin & W. Dabbous (2005). *Improving Cross-domain Authentication over Wireless Local Area Networks*, *Proc. of 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks, SECURECOMM'05*, IEEE Computer Society, Athens, Greece, pp. 103–109.

IEEE 802.11 (2007). Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

IEEE 802.11i (2005). Std., Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security.

IEEE 802.11 IAPP (2003). IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation.

IEEE 802.11r (2005). , Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 8: Fast BSS Transition.

IEEE 802.16e (2006). Air Interface for Fixed and Mobile Broandband Wireless Access System.

IEEE 802.1X (2004). Standards for Local and Metropolitan Area Networks: Port based Network Access Control, IEEE Standards for Information Technology.

IEEE 802.21 (2008). Institute of Electrical and Electronics Engineers, Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services.

ITU-T Recommendation G.114 (1998). ITU-T General Characteristics of International Telephone Connections and International Telephone Circuits: One-Way Transmission Time, ITU-T Recommendation G.114.

J. Bournelle, M. Laurent-Maknavicius, H. Tschofenig, Y. El Mghazli, G. Giaretta, R. Lopez & Y. Ohba (2006). *Use of Context Transfer Protocol (CXTP) for PANA*. IETF Internet Draft, draft-ietf-pana-cxtp-01.

Marin-Lopez, R., Pereniguez, F., Bernal, F. & Gomez, A. (2010). *Secure three-party key distribution protocol for fast network access in EAP-based wireless networks*, *Computer Networks* 54: 2651 – 2673.

N. Nasser, A. Hasswa & H. Hassanein (2006). *Handoffs in Fourth Generation Heterogenous Networks*, *IEEE Communications Magazine* vol. 44(10): pp. 96–103.

P. Calhoun, G. Zorn, D. Spence & D. Mitton (2005). *Diameter Network Access Server Application*. IETF RFC 4005.

P. Calhoun & J. Loughney (2003). *Diameter Base Protocol*. IETF RFC 3588.

P. Eronen, T. Hiller & G. Zorn (2005). *Diameter Extensible Authentication Protocol (EAP) Application*. IETF RFC 4072.

R. Dantu, G. Clothier & Anuj Atri (2007). *EAP methods for wireless networks*, *Elsevier Computer Standards & Interfaces* vol. 29: pp. 289–301.

R. Housley & B. Aboba (2007). *Guidance for Authentication, Authorization, and Accounting (AAA) Key Management*. IETF RFC 4962.

R. M. Lopez, A. Dutta, Y. Ohba, H. Schulzrinne & A. F. Gomez Skarmeta (2007). *Network-Layer Assisted Mechanism to Optimize Authentication Delay during Handoff in 802.11 Networks*, *Proc. of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, ACM Mobiquitous 2007*, ACM, Philadelphia, USA.

R. Marin, J. Bournelle, M. Maknavicius-Laurent, J.M. Combes & A. Gomez-Skarmeta (2006). *Improved EAP keying framework for a secure mobility access service*, *Proc. of International Wireless Communications & Mobile Computing Conference 2006, IWCMC 2006*, Vancouver, British Columbia, Canada, pp. 183–188.

S. Pack & Y. Choi (2002). *Fast Inter-AP Handoff using Predictive-Authentication Scheme in a Public Wireless LAN*, *Proc. of IEEE Networks 2002 (Joint ICN 2002 and ICWLHN 2002)*.

S. Winter, M. McCauley, S. Venaas & K. Wierenga (2010). *TLS encryption for RADIUS*. IETF Internet-Draft.

T. Aura & M. Roe (2005). *Reducing Reauthentication Delay in Wireless Networks*, *Proc. of 1st IEEE Security and Privacy for Emerging Areas in Communication Networks, SECURECOMM 2005*, IEEE, Athens, Greece, pp. 139–148.

T. Clancy, M. Nakhjiri, V. Narayanan & L. Dondeti (2008). *Handover Key Management and Re-authentication Problem Statement*. IETF RFC 5169.

T. Dierks & C. Allen (1999). *The TLS Protocol Version 1.0*. IETF RFC 2246.

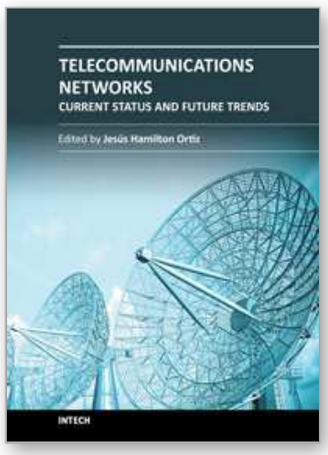
Taniuchi, K., Ohba, Y., Fajardo, V., Das, S., Yuu-Heng, M. T. C., Dutta, A., Baker, D., Yajnik, M. & Famolari, D. (2009). *IEEE 802.21: Media independent handover: Features, applicability, and realization*, *IEEE Communications Magazine* 47(1): 112 –120.

V. Narayanan & L. Dondeti (2008). *EAP Extensions for EAP Re-authentication Protocol (ERP)*. IETF RFC 5296.

Y. Ohba and A. Yegin (2010). *Pre-Authentication Support for the Protocol for Carrying Authentication for Network Access (PANA)*. IETF RFC 5873.

Y. Ohba, Q. Wu & G. Zorn (2010). *Extensible Authentication Protocol (EAP) Early Authentication Problem Statement*. IETF RFC 5836.

Z. Cao, H. Deng, Y. Wang, Q. Wu & G. Zorn (2011). *EAP Re-authentication Protocol Extensions for Authenticated Anticipatory Keying (ERP/AAK)*. IETF Internet Draft, raft-ietf-hokey-erp-aak-06.



Telecommunications Networks - Current Status and Future Trends

Edited by Dr. Jesús Ortiz

ISBN 978-953-51-0341-7
Hard cover, 446 pages
Publisher InTech
Published online 30, March, 2012
Published in print edition March, 2012

This book guides readers through the basics of rapidly emerging networks to more advanced concepts and future expectations of Telecommunications Networks. It identifies and examines the most pressing research issues in Telecommunications and it contains chapters written by leading researchers, academics and industry professionals. Telecommunications Networks - Current Status and Future Trends covers surveys of recent publications that investigate key areas of interest such as: IMS, eTOM, 3G/4G, optimization problems, modeling, simulation, quality of service, etc. This book, that is suitable for both PhD and master students, is organized into six sections: New Generation Networks, Quality of Services, Sensor Networks, Telecommunications, Traffic Engineering and Routing.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen