**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 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.

**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.

**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.

**2.3 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.

**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 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)):

(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**

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 from an EAP authenticator (current) to a new one (target). When the user moves to the new authenticator, it can use the transferred context (e.g., cryptographic keys and associated lifetimes) to execute a security association protocol with the new authenticator 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.

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 where the mobile user performs a full EAP authentication with a candidate authenticator through the current associated one before it performs the handoff. In this manner, when the handoff happens, 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 with it to protect the wireless link. As we see, pre-authentication decouples the authentication and network access control operations from the handoff.

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. The mobile user initially performs an EAP authentication with the AAA server. Once the EAP authentication is successfully completed, the AAA server pre-distributes keys to authenticators which the user can potentially associate to in a near future. Therefore, when the peer moves to a new authenticator, 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. 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.

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. The user firstly performs a full EAP authentication 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 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 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.

**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).

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.