Computer Communications 33 (2010) 1682–1694

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Privacy-enhanced fast re-authentication for EAP-based next generation network

F. Pereniguez [a,*], G. Kambourakis [b], R. Marin-Lopez [a], S. Gritzalis [b], A.F. Gomez [a]

[a] Department of Information and Communications Engineering (DIIC), University of Murcia, Facultad de Informatica, Campus de Espinardo S/N, Murcia, Spain
[b] Laboratory of Information and Communication Systems Security, Department of Information and Communication Systems Engineering, University of the Aegean, Samos GR-83200, Greece

ARTICLE INFO

Article history:
Available online 24 February 2010

Keywords:
EAP
Fast re-authentication
Next generation networks
Privacy
Seamless handoff

ABSTRACT

In next generation networks one of the most challenging issues is the definition of seamless and secure handoffs in order to assure service continuity. In general, researchers agree on the use of EAP as an authentication framework independent of the underlying technology. To date, efforts have focused on optimizing the authentication process itself, leaving out other relevant but sometimes important aspects like privacy. In this paper we present a solution that provides a lightweight authentication process while preserving user anonymity at the same time. The goal is to define a *multi-layered pseudonym* architecture that does not affect the fast re-authentication procedure and that allows a user to be untraceable. Taking as reference our previous work in fast re-authentication, we describe the extensions required to support identity privacy. Moreover, results collected from an implemented prototype, reveal that the proposed privacy-enhanced fast re-authentication scheme is attainable without significant cost in terms of performance in 4G foreseeable environments.

1. Introduction

During recent years, users have shown high interest in the *always connected* experience. To support the combination of mobility and access to network services anywhere and anytime, communication networks are moving towards an *all-IP* network configuration, integrated by an IP-based network core and a set of access networks based on different wireless technologies. This scenario, which represents the 4*th* generation (4G) of mobile communications, enables the convergence of different heterogeneous wireless access networks in order to combine all the advantages offered by each link layer technology per se.

For the provision of high-quality multimedia services in the 4*th* generation (4G) of mobile communications, one of the most important challenges lies in reducing the time devoted to executing the network access control when the mobile user changes the point of attachment to the network. By decreasing this time, active communications can be re-established faster and therefore the perceived quality by the end-user can be significantly improved. One important factor that is part of the network access control is the authentication process required by network operators in order to control that only legitimate users are able to employ the operator's resources. A common way of performing this process in wireless networks has been guaranteed by the deployment of the so-called *Extensible Authentication Protocol* (EAP) [1]. The success of this protocol has been motivated by three important aspects: flexibility, wireless technology independence and integration with AAA infrastructures.

However, EAP has shown some drawbacks when a mobile scenario is taken into account. Typically, an EAP authentication lasts a considerable time and involves multiple message exchanges [2]. Moreover, these authentication messages have to travel to the user's home domain (HD), which could be situated far from the point of attachment to the network. Furthermore, this process is usually performed each time the mobile user changes the point of attachment during a *handoff*. Consequently, the resulting authentication mechanism is extremely inefficient. To solve this problem, researchers agree that is necessary to define a fast re-authentication process [3] that involves a *local re-authentication server* placed near the mobile user. In particular, it has been argued [4] that a fast and secure three-party key distribution process seems to be a more appropriate model for achieving this goal.

Another challenging issue associated with heterogeneous wireless networks is the privacy of the end-user. The nature of wireless networks allows a malicious user to eavesdrop or capture messages from any active communication that takes place under its coverage area. As a consequence, among other implications, this situation enables the user's activity to be monitored [5]. For this reason, privacy is a serious concern for both emerging applications and mobile users in future wireless networks. In fact, the protec-

[*] Corresponding author. Address: Department of Information and Communications Engineering (DIIC), University of Murcia, Facultad de Informatica, Campus de Espinardo S/N, 30100 Murcia, Spain. Tel.: +34 868 88 78 82; fax: +34 868 88 41 51.

E-mail addresses: pereniguez@um.es (F. Pereniguez), gkamb@aegean.gr (G. Kambourakis), rafa@um.es (R. Marin-Lopez), sgritz@aegean.gr (S. Gritzalis), skarmeta@um.es (A.F. Gomez).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.02.021

tion of user's privacy may become a sine qua non for the so-called *Next Generation Networks* (NGN), since without privacy-preserving mechanisms in place, the end-user can be easily tracked and profiled in the mid or long term. Thus, the user is left defenseless to spamming and/or other related threats that violate his private sphere.

In general, privacy is a complex concept that affects aspects such as location, identification and authentication [6]. While location privacy requires that the location of a mobile user is untraceable to unauthorized parties (including the network), identification privacy mandates users anonymity, except by authorized parties. As we can see, these types of privacy are interrelated. If the user's identity is private, then location data is useless. At the same time, both types of privacy strongly depend on the authentication process, where user permanent identity must be exchanged. If the authentication mechanism does not have an adequate level of privacy to protect identification related data, the location can be revealed to unauthorized third parties.

Given these problems, this paper presents a novel and simple multi-layered architecture for pseudo-random pseudonym generation that offers a privacy-preserving mechanism for fast re-authentication processes in EAP-based NGN. In particular, to show the benefits of our proposal, we apply our solution on our previous work [7] which proposes a secure three-party protocol named 3PFH, especially adapted for performing fast network access in EAP-based wireless networks. As we will show, our solution is even applicable when the handoff takes place between different administrative domains (e.g., different network operators), regardless of the wireless technologies deployed. Additionally, using several real scenarios, we demonstrate that the overload imposed by the privacy-enhanced solution is negligible in comparison with the non-privacy case.

The remainder of the paper is structured as follows: the next section offers the necessary background to understand the proposal. Section 3 presents the proposed *multi-layered pseudonym* architecture together with the required extensions to the 3PFH protocol to support our solution. In Section 4, we provide some implementation details and information about the deployed testbed that implements the privacy framework. In Section 5, over different scenarios, we further demonstrate that the privacy extensions require an insignificant overload and, therefore, their use does not suppose an additional latency during the re-authentication process. Section 6 shows relevant related work. Finally, Section 7 concludes the paper and provides some future directions.

2. Background

We provide here some basic concepts required to understand the context of application of our proposal. In particular, we present a brief overview of EAP and the basics of our secure three-party protocol (3PFH) so as to understand how it has been extended with our multi-layered pseudonym architecture.

2.1. EAP authentication in next generation networks

The *Extensible Authentication Protocol* (EAP) [1] proposes a framework that enables a user (called *EAP peer*) to execute an authentication protocol (*EAP method*) against an authentication server (called *EAP server*) through an *EAP authenticator* which merely forwards packets between the EAP peer and EAP server. While the EAP authenticator is typically placed in the *Network Access Server*, the EAP server can be co-located with the EAP authenticator (*standalone configuration*) or with a backend AAA server (*pass-through configuration*). In order to deliver EAP messages, an *EAP lower-layer* is used to transport the EAP packets between the EAP peer and the EAP authenticator. Additionally, when it is necessary to contact a backend AAA server, an AAA protocol (such as RADIUS [8] or Diameter [9]), is used for the same purpose between the EAP authenticator and the AAA server.

The authentication process starts when the authenticator requests the peer's identity through an *EAP Request/Identity* message. The peer answers with an *EAP Response/Identity* that contains its identity, represented using the *Network Address Identifier* (NAI) format [10]. After several exchanges of EAP Request and Response messages between the peer and EAP server, a successful EAP authentication finishes with an EAP Success message and the provision of keying material [11]: the *Master Session Key* (MSK) and the *Extended Master Session Key* (EMSK).

However, a typical EAP authentication may require several exchanges and must be executed each time the peer changes to a new authenticator. Furthermore, typical deployments require contacting the EAP server located in the peer's HD, which may be far from the EAP authenticator, especially when the peer is visiting a domain. To address all these problems, the authors have designed a novel approach based on a secure three-party protocol named 3PFH [7] and a companion transport based on a new EAP method named EAP-FRM [12]. Nevertheless, the important aspect of user privacy during authentication process was not addressed in these previous works.

2.2. The three-party protocol for fast handoff (3PFH)

The 3PFH protocol is composed by four main exchanges, as Fig. 1 depicts.

As can be observed, 3PFH is executed between three entities namely A, B and S. The protocol assumes that A and S share a symmetric key K_{AS}. This key is dynamically derived from a key hierarchy started from the EMSK exported during a full EAP authentication (or EAP re-authentication) involving A (acting as EAP peer) and S (acting as EAP server). This process is carried out in the so-called *bootstrapping phase*, which usually happens when the mobile gets network access for the first time or when the EMSK lifetime has expired. Similarly, it is also assumed that a pre-established key K_{BS} is only known by B and S.

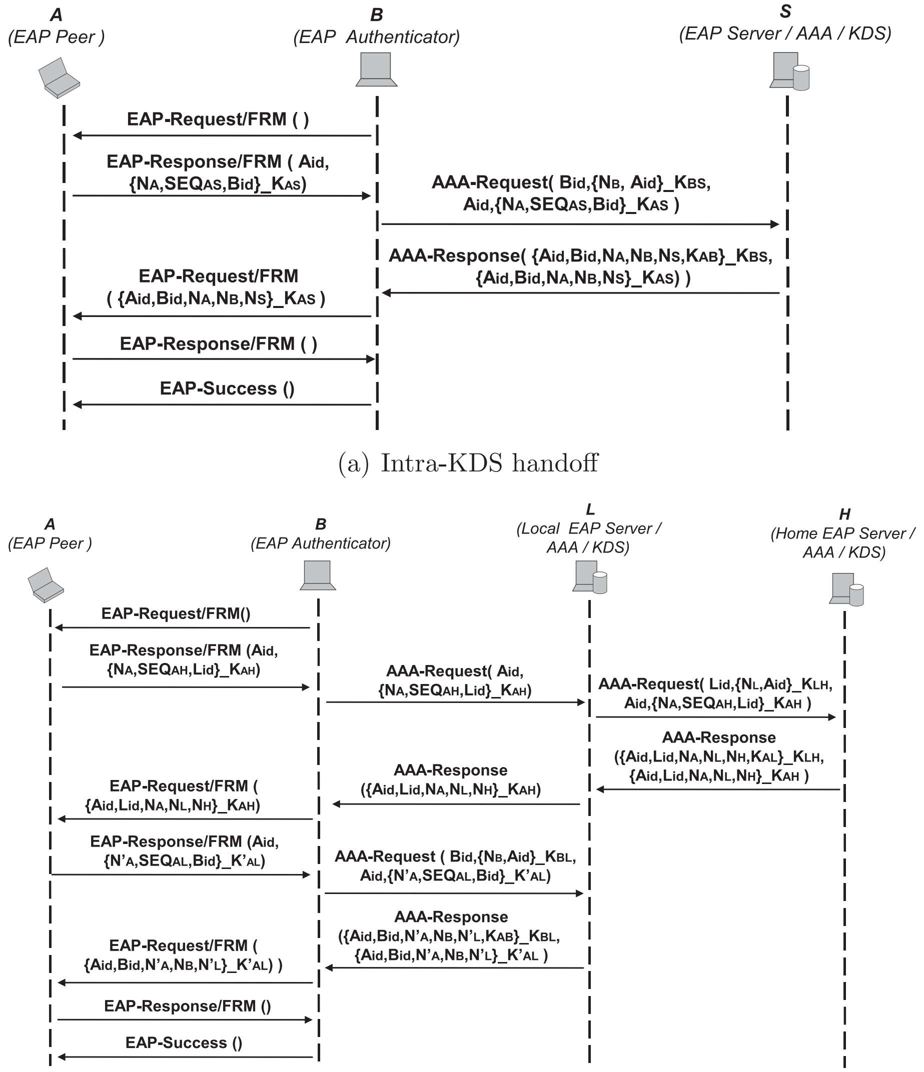
Fig. 2 shows how 3PFH is used in conjunction with the transport EAP-FRM [12] to achieve a complete, fast network access solution. Without loss of generality, we assume that the EAP server and fast re-authentication server (named KDS in the 3PFH context) are implemented over the same AAA server. In particular, Fig. 2(a) summarizes the case where the EAP peer already shares a key K_{AS} with a local server and roams between different EAP authenticators under the same server (*intra-KDS handoff case*). Conversely,

1. A $\Rightarrow$ B: A_{id},$\{N_A, SEQ_{AS}, B_{id}\}_{K_{AS}}$
2. B $\Rightarrow$ S: B_{id},$\{N_B, A_{id}\}_{K_{BS}}$,A_{id},$\{N_A, SEQ_{AS}, B_{id}\}_{K_{AS}}$
3. S $\Rightarrow$ B: $\{A_{id}, B_{id}, N_A, N_B, N_S\}_{K_{AS}}$,$\{A_{id}, B_{id}, N_A, N_B, N_S, K_{AB}\}_{K_{BS}}$
4. B $\Rightarrow$ A: $\{A_{id}, B_{id}, N_A, N_B, N_S\}_{K_{AS}}$

where

- A, B are the entities to which a key is distributed;
- S is key distribution server (KDS) which acts as fast re-authentication server;
- X_{id} denotes the identity of entity X;
- $\{X\}_K$ refers to X encrypted with key K providing confidentiality and integrity;
- K_{XY} refers to a symmetric key shared between parties X and Y;
- N_X refers to pseudo-random number acting as nonce and provided by the entity X;
- SEQ_{XY} is a sequence number maintained by parties X and Y;

Fig. 1. Typical 3PFH execution.


A
(EAP Peer)
B
(EAP Authenticator)
S
(EAP Server / AAA / KDS)
EAP-Request/FRM ( )
EAP-Response/FRM (Aid, {NA,SEQAS,Bid}_KAS)
AAA-Request(Bid,{NB, Aid}_KBS, Aid,{NA,SEQAS,Bid}_KAS)
AAA-Response({Aid,Bid,NA,NB,NS,KAB}_KBS, {Aid,Bid,NA,NB,NS}_KAS))
EAP-Request/FRM ({Aid,Bid,NA,NB,NS}_KAS)
EAP-Response/FRM ( )
EAP-Success ()
A
(EAP Peer)
B
(EAP Authenticator)
L
(Local EAP Server / AAA / KDS)
H
(Home EAP Server / AAA / KDS)
EAP-Request/FRM()
EAP-Response/FRM (Aid, {NA,SEQAH,Lid}_KAH)
AAA-Request(Aid, {NA,SEQAH,Lid}_KAH)
AAA-Request(Lid,{NL,Aid}_KLH, Aid,{NA,SEQAH,Lid}_KAH)
AAA-Response ({Aid,Lid,NA,NL,NH,KAL}_KLH, {Aid,Lid,NA,NL,NH}_KAH)
AAA-Response ({Aid,Lid,NA,NL,NH}_KAH)
EAP-Request/FRM ({Aid,Lid,NA,NL,NH}_KAH)
EAP-Response/FRM (Aid, {N'A,SEQAL,Bid}_K'AL)
AAA-Request (Bid,{NB,Aid}_KBL, Aid,{N'A,SEQAL,Bid}_K'AL)
AAA-Response ({Aid,Bid,N'A,NB,N'L,KAB}_KBL, {Aid,Bid,N'A,NB,N'L}_K'AL)
EAP-Request/FRM ({Aid,Bid,N'A,NB,N'L}_K'AL))
EAP-Response/FRM ()
EAP-Success ()

(a) Intra-KDS handoff

(b) Inter-KDS handoff

Fig. 2. Fast re-authentication phase.

Fig. 2(b) shows the case where the EAP peer moves to an EAP authenticator that is under the control of a local server with which the EAP peer does not share any key (*inter-KDS handoff case*). In this case, 3PFH is used first to establish a trust relationship between the EAP peer A and the local server L. This is accomplished by distributing a shared key K_{AL} from the home server H, which acts as the KDS. By using this K_{AL}, another 3PFH execution is performed to distribute a shared key K_{AB} between EAP peer A and the EAP authenticator B. Now, the local server L acts as the KDS. Following handoffs under the same local server L would then involve simple intra-KDS handoffs.

3. Multi-layered pseudonym architecture for fast re-authentication

The multi-layered pseudonym architecture aims to improve privacy not only during initial EAP authentication (*bootstrapping phase*) but also during the fast handoff process (*fast re-authentication phase*). The process of designing the solution has been guided by the following requirements:

1. *User anonymity.* Mobile users must remain anonymous during the authentication process not only to eavesdroppers but also to the visited domain. Only the HD can have access to users' real identities.
2. *Untraceability.* It must be impossible for an eavesdropper to trace network access activity of a certain user.

In order to accomplish these requirements, our solution relies on a *multi-layered pseudonym* architecture that provides user pseudonyms depending on the action that is to be performed. In general, it is assumed that a mobile user is recognized through a real Permanent Identity (*PI*) employed in every transaction that requires user identification and, in particular, during network access. To preserve identity privacy during the authentication required to provide network access control, the proposed multi-layered pseudonym architecture establishes that a mobile user, besides his permanent identity, has three types of pseudonyms. Each one is destined for a specific usage and must be renewed after it has been employed. These three pseudonyms are

- *Bootstrapping Pseudonym (BP):* This type of pseudonym is used in EAP *authentications* or EAP *re-authentications* during the bootstrapping phase. A specific BP can only be employed to carry out one of these processes. At any time, there is only one instance of this type of pseudonym.

- *Home Fast Pseudonym (HFP):* This type of pseudonym is employed in each fast re-authentication process initiated by the peer where the participation of the home fast re-authentication server is required. Furthermore, a particular HFP can only be employed in one *fast re-authentication* process. So, it must be renewed for the next fast re-authentication involving the home fast re-authentication server. At any time, there is only one instance of this type of pseudonym.
- *Visited Fast Pseudonym (VFP):* This type of pseudonym is selected by the peer when performing a handoff to an authenticator controlled by a specific fast re-authentication server placed inside a visited domain. As before, a given VFP can only be employed to perform one *fast re-authentication* process. After that, the VFP must adopt a new value. At a particular time, there can exist several instances of this type of pseudonym, with each one being associated to a different visited re-authentication server (e.g., when performing an EAP pre-authentication [13]).

According to the multi-layered architecture for privacy, the peer has to store a *n*-tuple of pseudonyms $(BP, HFP_i, VFP_j, VFP_k, \ldots, VFP_m)$. Each one of these pseudonyms is destined for a specific usage and used only once during each authentication or fast re-authentication process. Therefore, their value must be renewed after their use. Regarding the pseudonym renewal process, we follow a *server-controlled pseudonym generation* approach where an authentication server *pseudo-randomly* generates and distributes the different pseudonyms that the peer has to use in future authentication or fast re-authentication processes. More specifically, while the home EAP server renews the BP, the home and visited fast re-authentication servers respectively control the generation of the HFP and VFP. In this way, the server is able at all times to identify the user since it knows the new pseudonym that will be used in the future. Moreover, the distribution of these pseudonyms from the server to the peer (that is, the *renewal process*) is securely protected. In other words, confidentiality, integrity and replay protection are provided by the communication channel used between the server and the peer. Thus, an eavesdropper cannot know what pseudonym the user will employ for the next bootstrapping or fast re-authentication process.

As we can observe, the real identity is never revealed either to eavesdroppers or to the visited domain, and anonymity is thus assured (*req.1*). Furthermore, taking into account that each type of pseudonym is pseudo-random and changes its value after its use; and since an eavesdropper does not know what pseudonym the user will use in later authentications, thanks to the secure distribution, we conclude that an eavesdropper is unable to trace a user's activity (*req.2*) as no relation can be established between two different access control processes.

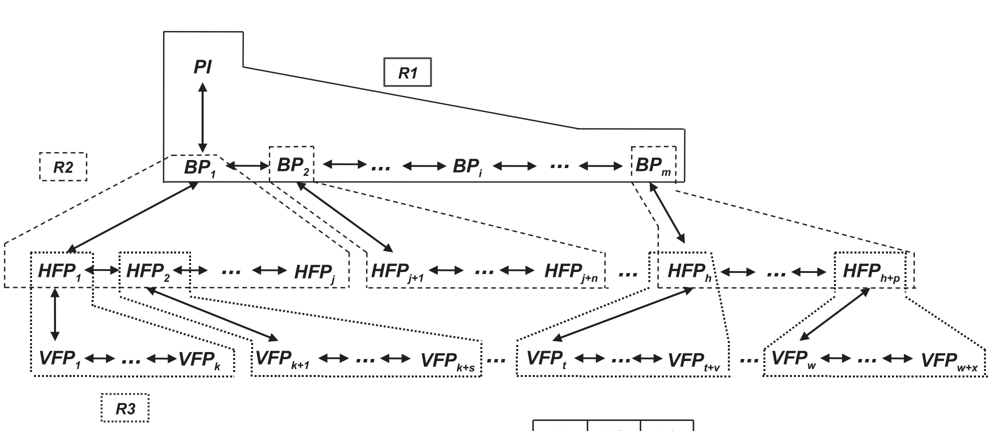
Another important aspect of our solution is related to the use of the pseudonyms. Of the three types of pseudonyms, the BP is the only one used during the bootstrapping phase, which comprises the initial step prior to the fast re-authentication phase. To maintain user anonymity, this kind of pseudonym must be available for use in the next bootstrapping operation, even when the mobile device is turned off and restarted later on. In contrast, this requirement is not necessary for HFP and VFP since they are dynamically established at runtime.

Below we analyze the particular features of the proposed privacy architecture and demonstrate how user anonymity is achieved. Next, we describe in detail how the generation, distribution and use of these pseudonyms (BP, HFP and VFP) are carried out in our solution. Finally, we describe the operation of the privacy solution over a usage scenario. To simplify the explanation, we assume the use of 3PFH as the fast re-authentication solution.

3.1. Privacy analysis

In general, three different authentication servers are involved in an EAP-based fast re-authentication operation: home EAP server, home fast re-authentication server and visited fast re-authentication server. Although the EAP and fast re-authentication server functionalities are typically implemented by the same AAA server, we have designed the privacy solution according to the most general approximation in order to avoid specific implementation aspects. In fact, the multi-layered pseudonym architecture proposes that each authentication server controls (that is, generates, distributes and renews) a different type of pseudonym used by the mobile user in every interaction with the specific server.

The pseudonyms employed by a user during the EAP authentications or fast re-authentications are interrelated, leading to a four-layer pseudonym graph, as depicted in Fig. 3. While the first layer consists of the PI, the remaining ones are composed of the



	R1	R2	R3
MOBILE NODE	✓	✓	✓
HOME EAP SERVER	✓	✗	✗
HOME FAST RE-AUTH. SERVER	✗	✓	✗
VISITED FAST RE-AUTH. SERVER	✗	✗	✓
EAVESDROPPER	✗	✗	✗

Fig. 3. Multi-layered pseudonym architecture.

BP, HFP and VFP pseudonyms, respectively. The relationships, depicted with arrows, are direct relationships established between pairs of pseudonyms. The pair is formed in a direct relationship between the pseudonym used during one action (EAP authentication or fast re-authentication) and the one distributed in the same action.

We distinguish between *intra-layer* and *inter-layer* relationships. While the former refers to relationships between pseudonyms that belong to the same layer (e.g., BP_1 and BP_2), the latter refers to relationships between pseudonyms that belong to different layers (e.g., BP_1 and HFP_1). In general, all the relationships are dynamically created when the user performs EAP authentications or fast re-authentications. However, the initial association between *PI* and BP_1 must be pre-defined. We provide further details about this aspect in Section 3.2.

Both intra-layer and inter-layer relationships are transitive. That is, if pseudonym P_i is associated with P_j (P_i, P_j) and P_j is associated with P_k (P_j, P_k), pseudonyms P_i and P_k are also associated (P_i, P_k). Therefore, other transitive relationships (not shown in Fig. 3) can be established. For example, by applying the transitive property, for any given pseudonym we can find the real user's identification (PI).

In our architecture, the mobile node is the only entity that knows the whole graph and all the relationships, since it receives and uses the different types of pseudonyms. Nevertheless, unless required for other reasons, it is not necessary for the mobile node to store the whole graph. As previously described, we optimize this aspect by defining a *n*-tuple of pseudonyms that strictly contains those that will be necessary in the future. As for the authentication servers, they have a partial vision of the graph (see table in Fig. 3). We distinguish three different groups of relationships:

- *Group R1* collects the relationships between the PI and all the BPs. There is only one group R1 that is known by the home EAP server.
- *Group R2* is formed by the relationships that exist between the BP used in a bootstrapping process and all the HFPs used in subsequent fast re-authentications with the HD. There are as many R2 groups as BPs generated. The home fast re-authentication server knows all these groups of relationships but it is unable to relate one to another, since it does not know the intra-layer relationships between the BPs.
- *Group R3* collects the relationships between the HFP employed in the initial fast re-authentication with the HD (in order to establish a trust relationship with the specific visited domain [4]) and the VFPs generated in the subsequent fast re-authentications inside the visited domain. There are as many R3 groups as HFPs generated. Each visited fast re-authentication server knows one or more groups R3 (e.g., when the mobile node roams to the same visited domain several times). Nevertheless, no visited fast re-authentication server can associate different R3 groups, since they do not know the intra-layer relationships between HFPs.

Although each type of authentication server knows a different group of relationships, it is important to note that the groups have pseudonyms that serve as connection points between the different groups of relationships: the BPs between groups R1 and R2 and HFPs between groups R2 and R3. These connections allows the user to be referred to in the last term if it is required, for example, for accounting purposes between different trusted authentication servers.

Although by using pseudonyms *anonymity* is assured, since the user employs a fictitious identity instead of its permanent identity, this feature in itself does not allow a user to remain completely *untraceable*. As explained, when a mobile user employs a pseudonym he receives a new one for the next action that will be carried out in the future. If an eavesdropper discovers the value of the new pseudonym when it is distributed after using the current pseudonym, it can deduct that both pseudonyms are somehow related. As a result, by applying this inference, he may have access to part of the graph of relationships described in Fig. 3. To avoid this, our solution uses a secure communication channel through which the authentication server distributes new pseudo-random pseudonyms to the peer. In this manner, an eavesdropper cannot observe the distributed pseudonym and cannot relate it in the future with any other one used in the current action (EAP authentication or fast re-authentication). In other words, the eavesdropper is unable to deduce any kind of direct (or transitive) relationship between pseudonyms and, thus, cannot know that these pseudonyms are associated to the same user.

3.2. Privacy-enhanced bootstrapping phase

During the bootstrapping phase, the peer will select from the *n*-tuple of pseudonyms the current BP_i to perform the full EAP authentication (or EAP re-authentication) with its home EAP server. After a successful exchange, the peer must acquire a new BP_{i+1} and HFP_j. While the BP_{i+1} will be used in the next bootstrapping (e.g., when EMSK lifetime expires), the HFP_j may be necessary in the near future for a fast re-authentication process with the home KDS. Additionally, when the bootstrapping process is performed after the EMSK expires, it is important to mention that all the VFPs stored in the *n*-tuple are removed. New VFPs will be established through posterior key distribution processes to the visited fast re-authentication servers (see Section 3.3).

According to the server-controlled pseudonym generation scheme previously described, after a successful EAP authentication the home EAP server generates a random BP_i and HFP_j that it sends to the peer. To implement this solution, the bootstrapping EAP method must allow the secure delivery of these pseudonyms, as discussed in Section 3.1. Usually, tunneled EAP methods [14] may provide this kind of secure channel by establishing a TLS security association between the EAP peer and the EAP server. However, this tunnel mandates asymmetric cryptography and is usually established without authenticating the EAP peer (only server authentication is required).

As an alternative, the authors designed a new EAP method that is specially conceived to extend EAP functionality. In particular, it can be easily extended to support the privacy-related operations that we require during bootstrapping phase. The proposed EAP method named EAP-EXT [15] allows, after some messages for capability negotiation (*negotiation phase*), sequencing of multiple EAP methods (*inner methods*) within itself (*authentication phase*) before establishing a secure tunnel with the key material exported by the inner methods (*binding phase*). It allows operators to use any kind of EAP authentication method (not only those based on asymmetric cryptography) and authenticates the user before the secure channel establishment.

Capitalizing on these properties, we have extended EAP-EXT to provide support to our architecture during the bootstrapping phase (see Fig. 4). In particular, we first define a new capability flag *P* (1). When this flag is activated by the EAP server in an *EAP-Request/EXT*, it notifies the support of privacy extensions to the peer. In this case, if the peer desires these privacy extensions to be enabled, it activates the flag in the *EAP-Response/EXT*. Secondly, the creation of two TLVs (*Tag Length Value*) to convey a BP and a HFP is required (2). These new TLVs are included, during the binding phase, in the *EAP-Request/EXT* sent from the home EAP server to the peer. Additionally, in order to achieve a secure transport, the new TLVs are included inside an *Encrypted* TLV that enables confidentiality by encrypting the information contained in it. On the other hand, all the information exchanged during the binding phase is integrity

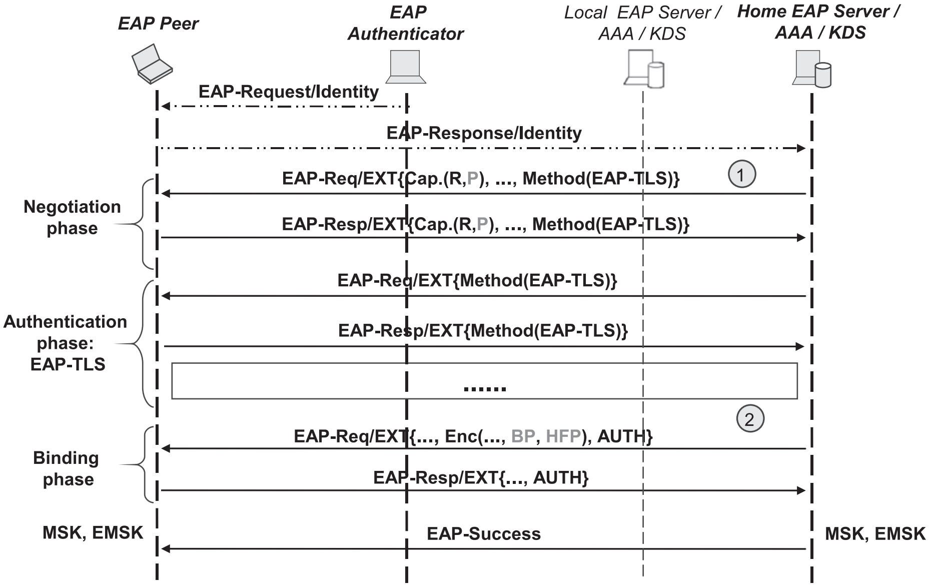

EAP Peer
EAP Authenticator
Local EAP Server / AAA / KDS
Home EAP Server / AAA / KDS
EAP-Request/Identity
EAP-Response/Identity
EAP-Req/EXT{Cap.(R,P), ..., Method(EAP-TLS)}
1
Negotiation phase
EAP-Resp/EXT{Cap.(R,P), ..., Method(EAP-TLS)}
EAP-Req/EXT{Method(EAP-TLS)}
Authentication phase: EAP-TLS
EAP-Resp/EXT{Method(EAP-TLS)}
......
2
EAP-Req/EXT{..., Enc(..., BP, HFP), AUTH}
Binding phase
EAP-Resp/EXT{..., AUTH}
MSK, EMSK
EAP-Success
MSK, EMSK

Fig. 4. Privacy-enhanced bootstrapping.

protected by the use of an *AUTH* TLV. Finally, it is worth noting that replay protection is assured since the keys used by the encryption and integrity algorithms are derived from the fresh key material exported by the inner authentication method.

So far we have assumed that the peer owns a valid BP that it uses to initiate an EAP authentication or EAP re-authentication. However, a special situation exists when the peer performs the very first bootstrapping phase. For example, when a new user accesses the operator's network for the first time. To maintain identity privacy in this situation, we assume that the user has an initial BP_1 delivered by his home network (e.g., pre-installed in the mobile device).

3.3. Privacy-enhanced fast re-authentication phase

During the fast re-authentication phase, the peer only uses the HFP and VFP types from the *n*-tuple of pseudonyms. While the HFP is selected when the home KDS participation is required, one of the VFPs is used to start a fast re-authentication process with the visited KDS associated to the specific VFP instance. After a 3PFH execution employs a certain pseudonym (either HFP or VFP), the involved KDS generates a new one that will be used by the peer in subsequent 3PFH exchange. Below we explain the required extensions to 3PFH to achieve a privacy-enhanced fast re-authentication phase. These modifications differ depending on whether the handoff involves the same KDS (*intra-KDS handoff*) or a different KDS (*inter-KDS handoff*).

3.3.1. Intra-KDS handoff

The intra-KDS handoff is performed between peer (A), authenticator (B) and a KDS (S). This KDS can be either the home KDS or a visited KDS located inside the visited domain. While in the first case an HFP_i is employed, in the second a VFP_j is selected by the user to perform the fast re-authentication operation. Regardless of whether the intra-KDS handoff takes place in the home or visited domain, 3PFH is extended in order to allow the KDS to securely deliver a new pseudonym (either HFP_{i+1} or VFP_{j+1}) to the peer.

As shown in Fig. 5, messages 1 and 2 follow the standard 3PFH operation using FP_1 that represents either the HFP_i or VFP_j pseudonym. When the KDS receives both messages and retrieves the profile associated to the user identified as FP_1, it realizes that this identity is a pseudonym and, therefore, identity privacy is enabled for this specific user. For this reason, once the KDS performs the normal 3PFH operation, it renews the employed pseudonym with a new random one, FP_2. To deliver FP_2 to the peer, we propose to include the new pseudonym inside *message 4* $\{FP_1, B_{id}, N_A, N_B, N_S, FP_2\}_{K_{AS}}$. This message contains a cryptographic token sent from the KDS to the peer. The information included in this token is encrypted (confidentiality) and integrity protected. Furthermore, the random N_A sent by the peer in *message 1* is expected in *message 4* to achieve protection against replay attacks. Note that, except for this modification, the rest of the 3PFH messages remain unchanged.

1. A $\Rightarrow$ B: $\boldsymbol{FP_1},\{N_A, SEQ_{AS}, B_{id}\}_{K_{AS}}$
2. B $\Rightarrow$ S: $B_{id},\{N_B, \boldsymbol{FP_1}\}_{K_{BS}},\boldsymbol{FP_1},\{N_A, SEQ_{AS}, B_{id}\}_{K_{AS}}$
3. S $\Rightarrow$ B: $\{\boldsymbol{FP_1}, B_{id}, N_A, N_B, N_S, \boldsymbol{FP_2}\}_{K_{AS}},\{\boldsymbol{FP_1}, B_{id}, N_A, N_B, N_S, K_{AB}\}_{K_{BS}}$
4. B $\Rightarrow$ A: $\{\boldsymbol{FP_1}, B_{id}, N_A, N_B, N_S, \boldsymbol{FP_2}\}_{K_{AS}}$

Fig. 5. Privacy-enhanced intra-KDS handoff.

Once the KDS finalizes the fast re-authentication process, the user's profile is updated by setting FP_2 as the next pseudonym (i.e. either HFP_{i+1} or VFP_{j+1}) that will be used by the user in subsequent 3FPH exchange with the same KDS.

3.3.2. Inter-KDS handoff

As shown in Fig. 6, the inter-KDS handoff requires the participation of four entities: peer (A), authenticator (B), visited KDS (L)

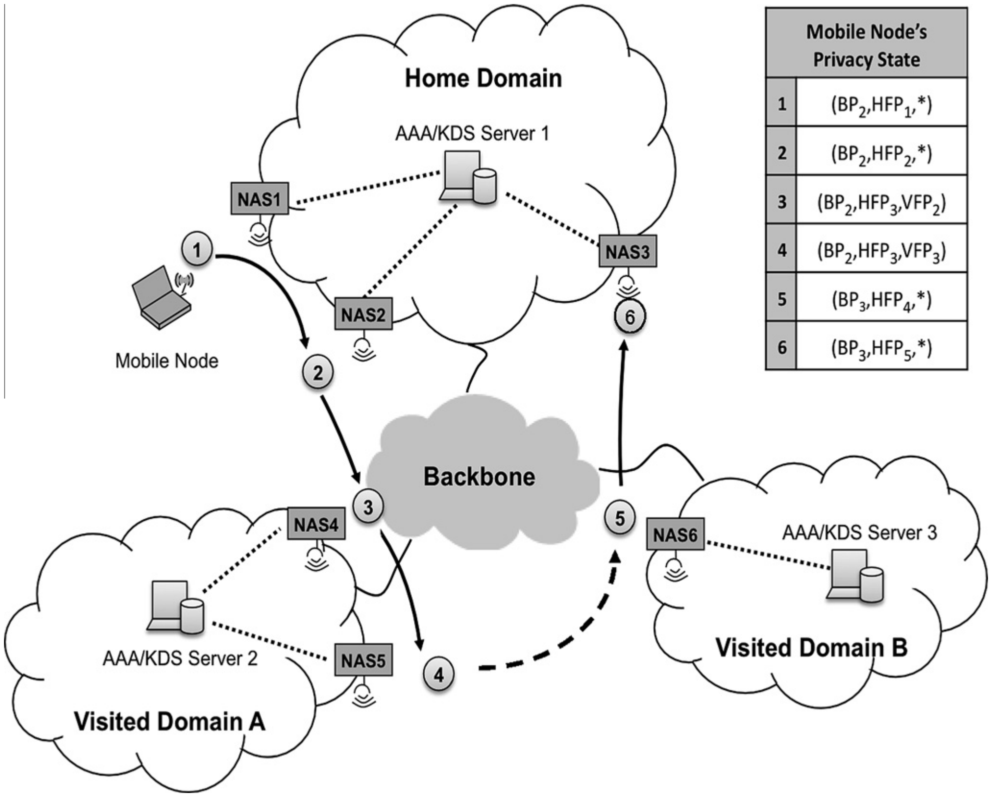
[Fist 3PFH Exchange between A – L – H]

1. A $\Rightarrow$ L: $\boldsymbol{HFP_j}\ ,\{N_A, SEQ_{AH}, L_{id}\}_{K_{AH}}$
2. L $\Rightarrow$ H: $L_{id},\{N_L, \boldsymbol{HFP_j}\}_{K_{LH}},\boldsymbol{HFP_j},\{N_A, SEQ_{AH}, L_{id}\}_{K_{AH}}$
3. H $\Rightarrow$ L: $\{\boldsymbol{HFP_j}, L_{id}, N_A, N_L, N_H, \boldsymbol{HFP_{j+1}}\}_{K_{AH}}$, $\{\boldsymbol{HFP_j}, L_{id}, N_A, N_L, N_H, K_{AL}, \boldsymbol{VFP_k}\}_{K_{LH}}$
4. L $\Rightarrow$ A: $\{\boldsymbol{HFP_j}, L_{id}, N_A, N_L, N_H, \boldsymbol{HFP_{j+1}}\}_{K_{AH}}$, $\boldsymbol{\{N_A, VFP_k\}_{K_{AL}}}$

[Second 3PFH Exchange between A – B – L]

5. A $\Rightarrow$ B: $\boldsymbol{VFP_k},\{N'_A, SEQ_{AL}, B_{id}\}_{K_{AL}}$
6. B $\Rightarrow$ L: $B_{id},\{N_B, \boldsymbol{VFP_k}\}_{K_{BL}},\boldsymbol{VFP_k},\{N'_A, SEQ_{AL}, B_{id}\}_{K_{AL}}$
7. L $\Rightarrow$ B: $\{\boldsymbol{VFP_k}, B_{id}, N'_A, N_B, N'_L, \boldsymbol{VFP_{k+1}}\}_{K_{AL}}$, $\{\boldsymbol{VFP_k}, B_{id}, N'_A, N_B, N'_L, K_{AB}\}_{K_{BL}}$
8. B $\Rightarrow$ A: $\{\boldsymbol{VFP_k}, B_{id}, N'_A, N_B, N'_L, \boldsymbol{VFP_{k+1}}\}_{K_{AL}}$

Fig. 6. Privacy-enhanced inter-KDS handoff.



Mobile Node's Privacy State	
1	$(BP_2, HFP_1, *)$
2	$(BP_2, HFP_2, *)$
3	(BP_2, HFP_3, VFP_2)
4	(BP_2, HFP_3, VFP_3)
5	$(BP_3, HFP_4, *)$
6	$(BP_3, HFP_5, *)$

Fig. 7. Scenario and steps.

placed inside the visited domain and home KDS (H). Let us suppose that the peer initially has the 2-tuple of pseudonyms (BP_i, HFP_j). The process starts with a first 3PFH execution performed between peer (A), visited KDS (L) and home KDS (H), in order to distribute a shared key K_{AL} between peer A and visited KDS L. Given that the home KDS participates by distributing a key to the peer and the visited KDS, HFP_j is selected as the peer's identity during the process. According to the multi-layered pseudonym architecture, the peer must renew HFP_j with a new value HFP_{j+1}. Following the same idea as the intra-KDS handoff, we extend *message 4* $\{\boldsymbol{HFP_j}, L_{id}, N_A, N_L, N_H, \boldsymbol{HFP_{j+1}}\}_{K_{AH}}$ by including the new pseudonym HFP_{j+1} generated by the home KDS.

Furthermore, an initial VFP shared with the visited KDS must be provided to the peer. This VFP will be used by the peer when performing a 3PFH-based fast re-authentication with this new KDS in the future. For this purpose, the home KDS generates a new VFP_k that is included in *message 3*. On the reception of this message, the visited KDS detects the presence of this new identity and knows that privacy support is enabled. The visited KDS can either accept this initial VFP or ignore it and generate a new one. Regardless of how the visited KDS handles this aspect, the VFP is sent to the peer through a new (not defined in the basic 3PFH specification) cryptographic token $\{N_A, VFP_k\}_{K_{AL}}$ included in *message 4*. This token includes the random number N_A for freshness and the new pseudonym VFP_k protected with the distributed key K_{AL}.

After the 3PFH execution successfully ends, the peer updates the n-tuple with the received home and visited identity: (BP_i, HFP_{j+1}, VFP_k). Next, another 3PFH exchange is performed between peer (A), authenticator (B) and visited KDS (L) in order to distribute a key K_{AB} for peer A and authenticator B. In this execution, the peer will use the recently distributed pseudonym VFP_k as identity. Moreover, at the end of this second 3PFH run, a new VFP_{k+1} is acquired for use in a later intra-KDS handoff under this specific visited KDS L.

Privacy during intra-KDS handoffs in the visited domain is managed by the visited KDS. That is, the visited KDS is in charge of renewing the VFP employed by the peer in fast re-authentications that involve this specific visited KDS. It is also worth noting that, according to Fig. 6, the visited domain is giving a privacy service to user HFP_i (pseudonym used during key distribution to local KDS in the first 3PFH exchange).

3.4. Example of usage scenario

For a better understanding of the proposed privacy mechanism we describe an example which shows how our approach works and how user anonymity is preserved. It is supposed that the mobile user carries a device which is able to link up to different kind of wireless networks. In order to simplify the explanation, the existence of multiple VFPs is not considered.

Fig. 7 shows the network topology in the 4G scenario used as an example. Apart from the HD the mobile user is subscribed to, there are two other administrative domains managed by network operators with which the home network maintains bilateral roaming agreements. Access networks in each domain are based on a different wireless access technology. Additionally, each administrative domain deploys a backend server (AAA/KDS) in charge of the access control related tasks such as authentication, authorization and accounting.

Let us suppose a mobile user receives different services that are sensitive to handoff. In order to develop the daily activity, the user changes his location by moving across the different domains. The specific itinerary followed by the user is shown in Fig. 7. It is important to mention that only from step 4 to 5 does the mobile user turn off his device to change location.

Through the aforementioned steps we analyze the authentication process and how our multi-layered pseudonym architecture operates to provide privacy to the authentication. Initially, the user only possess a BP, leaving the remaining types of pseudonyms undefined $(BP_1, \ast, \ast)$.

1. *Bootstrapping phase (full EAP authentication).* When the user attempts to connect to his HD for the first time (step 1), a bootstrapping process is performed employing BP_1. After the full EAP exchange, a fresh EMSK is established (from which the 3PFH key hierarchy is derived), current BP is renewed by BP_2 and a first HFP (HFP_1) is acquired. Thus the state observed by the mobile user is $(BP_2, HFP_1, \ast)$.
2. *Fast re-authentication phase (intra-KDS handoff).* The user selects HFP_1 to perform an intra-KDS handoff in the HD. After the 3PFH execution between the mobile user, NAS2 and AAA/KDS server 1, the user updates the 3-tuple with the new HPF_2 $[(BP_2, HFP_2, \ast)]$.
3. *Fast re-authentication phase (inter-KDS handoff).* In this step an inter-KDS handoff takes place from NAS2 to NAS4. In the first 3PFH execution to establish a trust relationship between the mobile user and AAA/KDS server 2, the pseudonym HFP_2 is used and then renewed with HFP_3 after this 3PFH execution. Moreover an initial VFP_1 is acquired, forming the new 3-tuple (BP_2, HFP_3, VFP_1). To complete the re-authentication process, the second 3PFH exchange with visited KDS uses VFP_1, which is renewed by VFP_2 for the next fast re-authentication process with the visited AAA/KDS server 2.
4. *Fast re-authentication phase (intra-KDS handoff).* This movement inside visited domain A to NAS5 is handled as an intra-KDS handoff using VFP_2. When AAA/KDS server 2 delivers a new VFP (VFP_3) to the mobile user, the 3-tuple is updated as (BP_2, HFP_3, VFP_3).
5. *Bootstrapping phase (full EAP authentication).* The user leaves visited domain A and needs network access service from visited domain B. During this change of location, the mobile device is reinitialized (remember that BP is available in these situations since it is stored in non-volatile memory) provoking the execution of a bootstrapping phase through NAS6. After the full EAP execution (employing BP_2) a new BP and HFP are acquired, and the resulting 3-tuple is $(BP_3, HFP_4, \ast)$.
6. *Fast re-authentication phase (inter-KDS handoff).* Finally, when the user returns to its HD, another inter-KDS handoff occurs. In this situation, given that EMSK has not expired, the 3PFH key hierarchy established in step 5 is still valid. For this reason, the mobile user selects HFP_4 as identifier and starts a 3PFH-based fast re-authentication with the home AAA/KDS. After obtaining the new HFP (HFP_5) generated by AAA/KDS server 1, the final tuple of pseudonyms is $(BP_3, HFP_5, \ast)$.

4. Deployed testbed and implementation details

In this section we provide some details about the prototype we have implemented in order to test the privacy solution behaviour. We will specially comment on aspects related to the anonymous identity format and generation.

4.1. Pseudonym format and generation

According to the EAP specification [1], we use the *Network Address Identifier* (NAI) format [10], which defines the user identifier in the form of *username@domain*. Regarding the anonymous user name generation, we implement this feature by generating a random binary string of a determined length. Nevertheless, a binary identity can easily violate the formal syntax outlined in the standard NAI specification. For this reason, we apply a base-64 encoding [16] to transform the random binary string into a representation which is compatible with NAI syntax.

In this respect, it is important to highlight that the base-64 encoding increases the volume of the resulting encoded data (33% in the worst case). Therefore, this aspect must be carefully taken into consideration because the final user name length may be longer than the randomly generated binary string.

4.2. Implemented prototype

To conduct real experiments, we have implemented the fast re-authentication framework based on 3PFH. In addition to the standard version without privacy support, a privacy-enhanced version has been developed according to the extensions proposed in Sections 3.2 and 3.3.

Regarding the bootstrapping implementation, this relies on the use of EAP-EXT as bootstrapping EAP method. Moreover, EAP-EXT has been implemented using EAP-TLS [17] as inner method. Given that EAP-EXT works over pass-through EAP authenticators, only the EAP peer and EAP server parts have been developed. The former is based on *wpa_supplicant*[1] and the latter on *FreeRadius*[2] software. According to EAP-EXT specification [15], two new TLVs are defined to contain the BP and HFP pseudonyms. Furthermore, to achieve a secure identity transport, two special TLVs (already defined in the specification) are employed in the EAP-EXT binding phase. Firstly, the *Encrypted* TLV (to encrypt the BP and HFP pseudonyms) is implemented through the well-known *Advanced Encryption Standard* (AES) algorithm with a block cipher size of 128 bits and cipher-block chaining (CBC) operation mode. Secondly, the AUTH TLV (to integrity protect the whole message) is computed through the widely used HMAC-SHA-256 function.

With respect to the fast re-authentication operation, the basic implementation we provided in [7] has been adapted to use the standalone EAP method EAP-FRM [12] to transport the 3PFH messages. The authenticator part has been developed based on *hostapd*[3] software. To build the cryptographic tokens, we have used the *AES Key Wrap* algorithm defined in RFC 3394 [18] with a *Key Encryption Key* (KEK) of 128 bits. The implementation follows the guidelines of [19] for wrapping an arbitrary plain text.

4.3. Experimental testbed

In order to provide a proper testbed to analyze the behaviour and to measure the performance of the privacy-enhanced fast re-authentication solution, we build a basic and generic scenario that allow us to represent different situations that can occur in a real mobile environment. In this context, the experimental testbed comprises the elements described in Table 1.

To experiment with real roaming scenarios, these devices are geographically distributed. While M4 is located in the University of the Aegean (Greece), the remaining machines are placed in the University of Murcia (Spain). This organization allow us to simulate two different administrative domains. The average ping time between the two domains is about 100 ms, but this value should only

Table 1
Testbed machines.

Machine	CPU type	Freq (MHz)	RAM (MB)	Role
M1	VIA Nehemiah	1200	488	Mobile node
M2	VIA Nehemiah	1200	488	Access point
M3	Pentium 4	3000	512	AAA/KDS
M4	Pentium 4	3000	512	AAA/KDS

[1] http://hostap.epitest.fi
[2] http://www.freeradius.org
[3] http://hostap.epitest.fi

be considered as an indication of the distance between the two domains.

5. Privacy overload evaluation

In this section we evaluate the penalty imposed by our multi-layered pseudonym architecture in order to determine if it is affordable for next generation networks. For this evaluation, we have conducted several experiments on the implemented testbed described in Section 4. We compare the privacy-enhanced fast re-authentication framework (*privacy-enhanced solution*) with the original version without privacy support (*no-privacy solution*) over four relevant scenarios. On the one hand, two scenarios arise for the bootstrapping phase depending on whether the mobile user is located in the home or visited domain. On the other hand, with respect to the fast re-authentication phase, we distinguish different scenarios for the intra-KDS and inter-KDS handoffs.

In every scenario, we create a network architecture to test the specific authentication scheme. When testing the privacy extensions, pseudonyms are in the form *base64(BinaryString_X_bytes)@domain* to follow the NAI syntax, where *BinaryString* is a randomly generated binary string that has *X* bytes length and *base64* represents the application of base-64 encoding. Although the use of 8 bytes length binary strings is enough to avoid collisions (2^{64} different anonymous user names), we have tested the behaviour of the privacy-enhanced solution with other different values of *X*: 16, 32 and 64 bytes. When illustrating the results, we indicate the specific pseudonym (BP, HFP or VFP) employed to conduct the experiments. We have carried out around 500 authentication tests over every configuration with both the no-privacy and privacy-enhanced scheme. When executing tests for the no-privacy solution we have collected the following performance metrics (used as the reference frame to estimate the privacy overload):

- *Mean authentication time.* This determines a 95% confidence interval that contains the time that a user needs to be authenticated. It does not include the security association establishment following the successful authentication.
- *Message processing time.* This is the time (expressed as a 95% confidence interval) that each entity devotes to message processing from when a packet is received on the network interface until the entity sends the response. Therefore, for example, EAP stack and AAA routing processing times are included.
- *Network time.* This metric measures the time (expressed as a 95% confidence interval) devoted to packet transmission over the network.
- *Message length.* It collects the length of the authentication messages. In the bootstrapping phase (based on EAP-EXT) we measure those messages exchanged during the final binding phase (only affected by privacy extensions). In the fast re-authentication phase, we specifically collect the 3PFH messages length, leaving out other information like AAA or EAP headers.

Table 2
No-privacy bootstrapping in home domain.

Full authentication time (ms)	329.989 ± 7.501	
Message processing time (ms)	Peer	102.421 ± 0.145
	Authenticator	3.161 ± 0.016
	Home AAA/KDS	25.436 ± 0.030
Network time (ms)	202.728 ± 7.499	
Message length (bytes)	Binding request	58
	Binding response	44

Table 3
Privacy-enhanced bootstrapping in home domain.

	Binary string length			
	8 bytes	16 bytes	32 bytes	64 bytes
Privacy computing time (ms)				
Peer	0.315 ± 0.006	0.322 ± 0.001	0.329 ± 0.005	0.358 ± 0.006
Home AAA/ KDS	0.062 ± 0.001	0.071 ± 0.001	0.096 ± 0.001	0.123 ± 0.001
Message length (bytes)				
Binding request	136	160	200	292
Binding response	44	44	44	44

BP/HFP = *base*64(*X_bytes*)@*home.domain*.

In order to perform an accurate measurement with the privacy-enhanced solution, we specifically measure the time required by the privacy-related operations. This time must be added to the time obtained for the no-privacy solution. In particular, the selected metrics are

- *Privacy computing time.* Through information obtained from the source code, we specifically measure the extra time (expressed as a 95% confidence interval) required by each entity to perform the additional tasks imposed by the privacy extensions.
- *Message length.* This is the same metric as described above for the no-privacy solution.

As we will see, results obtained from experiments revealed that the additional latency required by the privacy extensions does not impact substantially on the authentication times obtained in the no-privacy solution used as reference.

5.1. Scenario I: Bootstrapping in home domain

The first scenario is intended to evaluate the privacy penalty during an initial bootstrapping executed in the HD as illustrated in step 1 of Fig. 7. For this, we have created a network architecture consisting of three entities: peer (M1), authenticator (M2) and home AAA/KDS server (M3). Table 2 illustrates a 95% confidence interval for the message processing, authentication and network times in the no-privacy case. As we can see, a full EAP-EXT authentication requires about 330 ms, of which ≈60% is required for message transmission over the network. The peer is the entity that devotes more effort expending more than 100 ms in message processing.

Results obtained for the privacy-enhanced bootstrapping are summarized in Table 3 (let us remember that those times only refer to the latency spent in processing privacy extensions). With respect to the computing time, we observe that the additional time required in peer and home AAA/KDS to perform the privacy-related tasks (recall that pass-through authenticator does not perform any privacy operation) is below 0.36 ms and 0.14 ms, respectively. In

Table 4
No-privacy bootstrapping in visited domain.

Full authentication time (ms)	2469.661 ± 136.760	
Message processing time (ms)	Peer	102.879 ± 0.105
	Authenticator	3.386 ± 0.168
	Local AAA/KDS	2.297 ± 0.042
	Home AAA/KDS	29.237 ± 0.048
Network time (ms)	2330.098 ± 136.740	
Message length (bytes)	Binding request	58
	Binding response	44

Table 5
Privacy-enhanced bootstrapping in visited domain.

	Binary string length			
	8 bytes	16 bytes	32 bytes	64 bytes
Privacy computing time (ms)				
Peer	0.321 ± 0.013	0.324 ± 0.007	0.341 ± 0.008	0.357 ± 0.015
Home AAA/KDS	0.076 ± 0.001	0.085 ± 0.001	0.103 ± 0.008	0.141 ± 0.001
Message length (bytes)				
Binding request	136	160	200	292
Binding response	44	44	44	44

BP/HFP = *base*64(*X_bytes*)@*home.domain.*

both cases, those times represent less than 1% compared with the whole time required to process EAP-EXT messages (showed in Table 2). Furthermore, in the worst case (64 bytes length binary strings), the whole time required to complete the privacy processing is about 0.5 ms, which increases the authentication time only by 0.15%.

In relation to message length during the final binding phase (see Fig. 4), only the EAP-Request/EXT size increases since this message carries the BP and VFP pseudonyms. This augment is due to several factors: (a) base-64 encoding introduces an overload in the encoded binary string; (b) anonymous identities are stored in TLVs that includes the tag and length before the value; (c) the *Encrypted* TLV used to provide confidentiality includes an initialization vector of 16 bytes according to the encryption algorithm we employ.

Although the size of the message for the binding request increases significantly (about 4 times with respect to no-privacy solution), results collected from conducted experiments reveal that this increment in message length does not cause an appreciable penalization to the network time.

5.2. Scenario II: Bootstrapping in visited domain

The second scenario is also destined to evaluating the privacy-enhanced bootstrapping but when the user starts its access to the network from a visited domain (e.g., step 5 in Fig. 7). To implement this scenario, we use all the machines of our testbed, where M3 represents the local AAA/KDS and M4 acts as home AAA/KDS. As explained in Section 4.3, we are testing a real inter-domain scenario because machines M3 and M4 are located in different countries.

Note that the same EAP-EXT exchange (no-privacy or privacy-enhanced) is performed as described in scenario I. However, the local AAA/KDS now acts as AAA proxy by forwarding AAA messages between authenticator and the home AAA/KDS. Comparing the no-privacy solution measurements (shown in Table 4) with those obtained in scenario I (Table 2), we can observe that message processing times of peer, authenticator and home AAA/KDS are very similar. The difference appears in the time required to complete the full EAP-EXT exchange, especially for the network time. The location of the home AAA/KDS (far from the peer) means that every packet exchanged between peer and home AAA/KDS requires an additional delay in reaching its destination. Results reveal that the authentication time is about 2470 ms, of which 95% is dedicated to packet transmission.

Similarly, comparing the results obtained for the privacy-enhanced case (provided in Table 5) with those depicted in Table 3, we observe the same message lengths and very similar privacy computing times for peer and home AAA/KDS. In this scenario, given that the authentication time increases and the privacy latency does not vary and remains at similar values, privacy extensions are going to represent an almost negligible penalization. In fact, when generating pseudonyms from 64 bytes length binary strings, the use of privacy extensions supposes a penalization of only 0.02% in the overall authentication time.

Table 6
No-privacy fast re-authentication during intra-AAA/KDS handoff.

Fast re-authentication time (ms)	70.188 ± 1.723	
Message processing time (ms)	Peer	0.322 ± 0.015
	Authenticator	0.433 ± 0.008
	AAA/KDS	0.348 ± 0.011
Network time (ms)	69.085 ± 1.727	
Message length (bytes)	Message 1	73
	Message 2	68
	Message 3	104
	Message 4	88

A_{id} = username@domain.name//B_{id} = ap@domain.name.

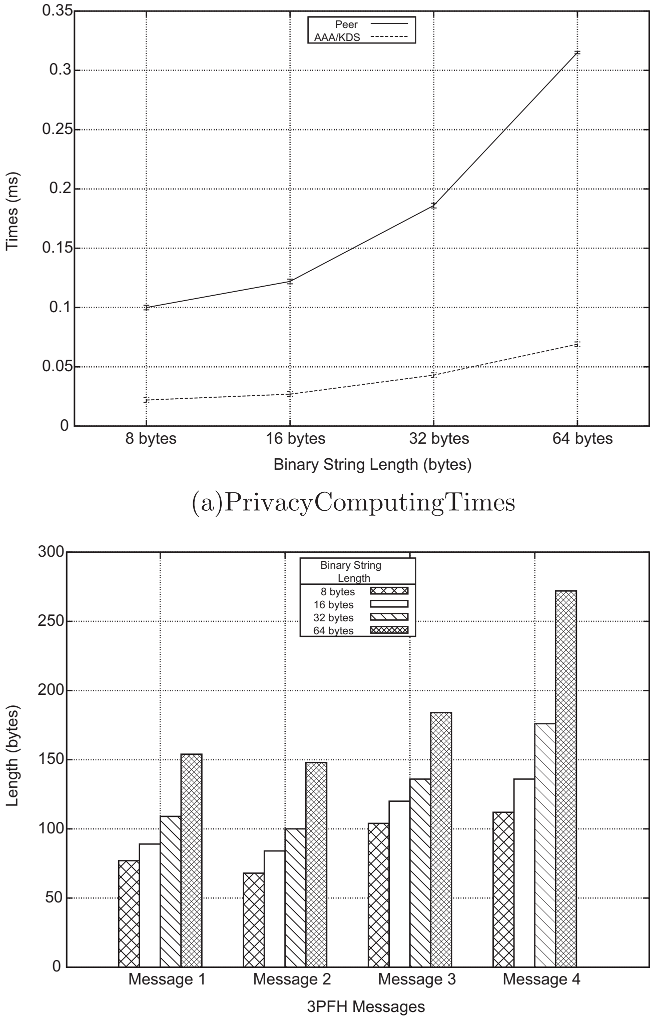
5.3. Scenario III: Fast re-authentication during intra-AAA/KDS handoff

This kind of handoff occurs when the peer is roaming between authenticators controlled by the same local KDS (either home or visited KDS) with which the peer already has a trust relationship based on a shared key (e.g., steps 2 and 4 in Fig. 7). As pointed out in Section 2.2, a 3PFH exchange is performed between peer, authenticator and local AAA/KDS. To carry out tests simulating this scenario, we configure a network topology integrated by the machines M1, M2 and M3 that represent the aforementioned entities, respectively.

Table 6 summarizes the results for the no-privacy fast re-authentication case. As a consequence of using 3PFH, authentication time is reduced to about 70 ms. Most of this time is devoted to packet transmission over the network, and only ≈1 ms is destined to message processing. These results demonstrate that low computing capabilities are required to perform the process.

Results obtained for the privacy-enhanced solution are illustrated in Fig. 8. More specifically, while Fig. 8(a) provides the additional privacy computing times required by peer and AAA/KDS, Fig. 8(b) shows the new 3PFH message sizes. As we can observe, measurements have been taken from different binary string lengths. Taking as reference the case where the peer's pseudonym is generated from an 8 bytes length binary string, we observe that privacy extensions do not impact substantially on the fast re-authentication process. Firstly, regarding the time overload, we observe that privacy support produces an almost insignificant penalty (close to 0.12 ms) in comparison with the overall authentication time. In respect to message length, only messages 1 and 4 are incremented by 4 and 24 bytes,[4] respectively. Again, the results demonstrate that these extra bytes do not suppose an appreciable penalization to the network time.

[4] It is to be noted that key wrap algorithm used to implement 3PFH introduces an overload to the wrapped output length.



(a)PrivacyComputingTimes

(b)MessageLength

Fig. 8. Privacy-enhanced fast re-authentication during intra-AAA/KDS handoff (A_{id}(HFP/VFP) = *base*64(*X_bytes*)*@domain.name*//B_{id} = *ap@domain.name*).

When analyzing the evolution when generating bigger random pseudonyms, we observe that privacy extensions remain stable, introducing a reduced overload. As observed, peer and AAA/KDS computing times depend on the pseudonym's size. Since the peer is expected to be a low-end capacity device, these computing times are greater than those required by a high-end machine like AAA/KDS. Nevertheless, in the worst case, the penalization is about only 0.32 ms for the peer and 0.07 ms for the AAA/KDS. In the case of the authenticator, the results from the experiments show that times remain at a stable value of 0.001 ms regardless of the binary string length. Therefore, we can conclude that with the bigger pseudonym lengths, the privacy solution increases the authentication time by about only 0.39 ms. In terms of the message length, we observe that all values are proportional to the peer's pseudonym length. Message 4 presents the largest increment since it transports the new pseudonym (HFP or VFP) delivered to the peer.

5.4. Scenario IV: Fast re-authentication during inter-AAA/KDS handoff

In this last scenario our intention is to analyze the behaviour of the proposed privacy solution during an inter AAA/KDS handoff (e.g., steps 3 and 6 in Fig. 7). That is, when the peer roams to a new authenticator under the control of a new KDS with which the peer does not share a key. As described in Section 2.2, this kind of handoff requires two executions of 3PFH and involves four entities: peer, authenticator, visited AAA/KDS and home AAA/KDS. Experiments have been conducted using all the machines of our testbed, where M3 and M4 represent the visited and home AAA/KDS, respectively.

Table 7
No-privacy fast re-authentication during inter-AAA/KDS handoff.

Fast re-authentication time (ms)	554.574 ± 10.227	
Message processing time (ms)	Peer	0.748 ± 0.034
	Authenticator	0.869 ± 0.003
	Visited AAA/KDS	0.815 ± 0.001
	Home AAA/KDS	0.462 ± 0.008
Network time (ms)	550.012 ± 10.268	
First 3PFH exchange message length (bytes)	Message 1	73
	Message 2	72
	Message 3	104
	Message 4	88
Second 3PFH exchange message length (bytes)	Message 1	73
	Message 2	68
	Message 3	104
	Message 4	88

A_{id} = username@home.domain//B_{id} = ap@local.domain.
L_{id} = server@local.domain.

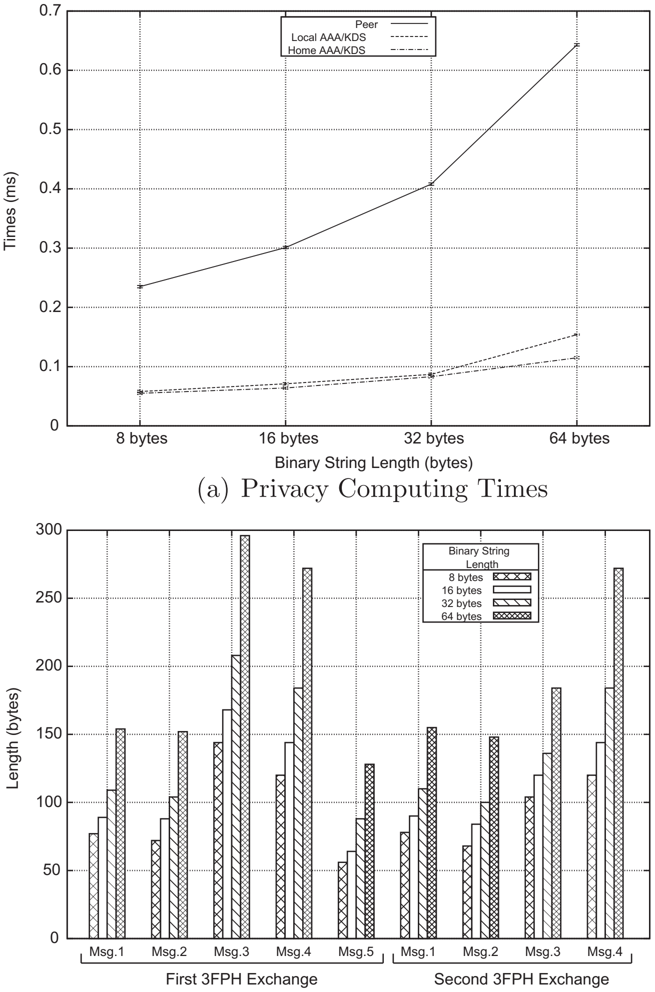
Collected results for the no-privacy solution (Table 7) reveal that authentication time is close to 550 ms, of which 99% is necessary for message transmission and the remaining 1% for message processing. Fig. 9 provides measurements taken from this configuration when our privacy extensions are enabled. More precisely, Fig. 9(a) and (b) shows the additional computing time and the 3PFH message length, respectively. Examining the results when generating 8 bytes length binary strings, we notice that although privacy tasks require ≈0.35 ms, the impact is smaller than in the intra-KDS handoff, increasing the authentication time by only ≈0.06%. Regarding message length, in both exchanges the behaviour is very similar to that observed in scenario III. Nevertheless, two main differences appear in the first 3PFH exchange. Firstly, there is a new message 5 (see Fig. 9(b)) that is required to deliver the first VFP shared with visited AAA/KDS to the peer. Secondly, we appreciate that message 3 values are greater since, in addition to the HFP, it transports the VFP generated by the home AAA/KDS.

When the privacy solution is employed with bigger random pseudonyms, penalty continues to fluctuate under insignificant values. As observed, not all the entities exhibit a significant increment in the privacy computing time. Again, high-end servers devote less time than the peer, who needs about 0.65 ms in the worst case (64 bytes) to complete the privacy-related procedures in an inter-KDS handoff. With respect to privacy computing times measured for the authenticator, executed tests indicate that this entity remains at a stable value close to 0.001 ms. In conclusion, considering the worst case (64 bytes), the whole privacy penalty (taking into account all privacy times for each entity) is ≈0.91 ms, which represents about 0.16% of the authentication time. Finally, considering the message length, the same conclusions mentioned for the intra-KDS case can be drawn here.

Thus, we can conclude that the extensions required by our privacy architecture do not impact on the authentication and fast re-authentication processes. Results obtained from the different scenarios demonstrate that the additional privacy computing times (almost negligible) and the increment in the message length do not affect the network access time appreciable.

6. Related work

The issue of user privacy protection has been the subject of study by researchers, especially in mobile environments [20]. On



(a) Privacy Computing Times

(b) Message Length

Fig. 9. Privacy enhanced fast re-authentication during inter-AAA/KDS handoff (A_{id}(HFP) = *base*64(*X_bytes*)@*home.domain*//A_{id}(VFP) = *base*64(*X_bytes*)@*local.domain*//B_{id} = *ap@local.domain*//L_{id} = *server@local.domain*).

the one hand, in the literature we can find a wide set of authentication protocols for wireless communication networks where user anonymity is integrated in the protocol design. For example, Refs. [21,22] define two different authentication schemes where user anonymity is preserved by using a pseudonym that does not change. This problem is solved by authors in [23,24] that incorporate a pseudonym renewal feature to allow the mobile user to remain untraceable. Nevertheless, all these authentication protocols do not consider any lightweight procedure to avoid contacting the HD in each execution. This drawback is fixed in [25], which defines a re-authentication process while maintaining user anonymity. However, these solutions cannot be applied in EAP networks straightforward since they consider neither the EAP-model analyzed in this paper (Section 2) nor the problem of fast re-authentication associated to EAP-based wireless networks that have influenced the design of our proposal.

On the other hand, the *Universal Mobile Telecommunication System* (UMTS) architecture has important security features like user identity confidentiality, achieved through the adoption of the Authentication and Key Agreement Protocol (AKA) specified by the Third-Generation Partnership Project (3GPP) [26]. AKA provides user anonymity by using temporary identities that are specially formatted. At the same time, AKA defines an optimized handoff procedure that allows authentication information to be retrieved from the previous authentication server visited, avoiding contacting user's home network. Nevertheless, AKA violates some properties of user identity confidentiality. There are two abnormal cases where a user is required to send its real identity. One is that the user performs the first authentication and the other is when the previous authentication server cannot be contacted. This deficiency allows an attacker to correlate real and temporary identities by using traffic analysis tools [27]. To solve these problems, there have been some efforts like in [27,28]. Nevertheless, it is important to note that these improvements are specific to 3G networks, leaving undefined a privacy optimization for other technologies.

In the EAP context, there are EAP methods that define some mechanism to provide anonymity during the authentication process. For example, the last revision of EAP-TLS [17] has a special privacy extension that allows the peer's certificate to be sent within a TLS session providing confidentiality. Conversely, EAP-SIM [29] and EAP-AKA [30] propose a solution similar to our approach which uses different types of pseudonyms that are frequently renewed. However, despite defining some procedure to minimize the signalling, they suffer from their inefficiency to provide a fast re-authentication. Unlike the multi-layered pseudonym architecture, which is optimized to work with a local server, these EAP methods always require contacting the home network.

Thus, to the best of authors' knowledge, previous privacy proposals in the field of authentication and access control are either oriented towards 3G networks or do not take into account the particularities of EAP, as a promising protocol for authentication and network access control in NGNs. In this way, the presented multi-layered pseudonym architecture represents the first novel approach to propose a privacy solution especially designed for integration into an EAP-based fast re-authentication mechanism that employs a local fast re-authentication server.

7. Conclusions

In this paper we have studied the problem of providing identity privacy and fast network access in the context of the so-called next generation heterogeneous networks. In this challenging scenario, EAP is acquiring a relevant position because it offers a framework for carrying out the authentication process independently of the underlying technology. Nevertheless, researchers are working on improving the basic traditional EAP-based network access by defining a fast re-authentication scheme. So far, efforts have been directed to reducing signalling and time to transmit messages, but little work has been done on also providing user anonymity during this process.

In [7], the authors present a fast re-authentication solution based on the definition of a three-party protocol named 3PFH. However, user privacy is not handled in any manner. Using this protocol as reference, we have proposed a privacy framework that is able to maintain user anonymity during authentication and fast re-authentication processes based on EAP. The design has been guided by strong principles such as user anonymity and untraceability. In particular, it relies on a multi-layered pseudonym model that defines a *n*-tuple of pseudonyms for a specific usage (authentication/re-authentication or fast re-authentication) that are frequently renewed. Furthermore, we have elaborated on the extensions required to the basic fast re-authentication solution based on 3PFH in order to implement the privacy architecture.

Finally, we have developed a software prototype in order to show the benefits of our solution. We have detailed some relevant implementation aspects like the pseudonym format and generation. Over a small testbed, we have tested different scenarios and investigated the penalty imposed by our proposal in terms of authentication time and message length. The results reveal that our architecture produces a negligible time penalty for the authentication procedure, and it is particularly viable in 4G deployments. In every tested configuration, the associated privacy cost is considered rather insignificant.

As future work, we plan to apply the privacy-enhanced framework to other fast re-authentication solutions. Similarly, we plan to extend the privacy framework to provide anonymity not only during the authentication process but also during the authorization and accounting processes.

Acknowledgements

This work has been partially supported by a Seneca Foundation grant from the Human Resources Researching Training Program 2007. It has also been funded by the Spanish Ministry of Science and Education under Grant TIN2008-06441-C02-02. Thanks also to the Funding Program for Research Groups of Excellence with code 04552/GERM/06 also granted by the Seneca Foundation. The authors also gratefully thank Fernando Bernal for his valuable help in the implementation process.

References

[1] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Extensible authentication protocol (EAP), RFC3748, June 2004.

[2] A. Dutta, D. Famolari, S. Das, Y. Ohba, V. Fajardo, K. Taniuchi, R. Lopez, H. Schulzrinne, Media-independent pre-authentication supporting secure interdomain handover optimization, IEEE Wireless Communications 15 (2) (2008) 55–64.

[3] T. Clancy, M. Nakhjiri, V. Narayanan, L. Dondeti, Handover key management and re-authentication problem statement, IETF RFC 5169, March 2008.

[4] D. Harskin, Y. Ohba, M. Nakhjiri, R. Marin, Problem statement and requirements on a 3-party key distribution protocol for handover keying, IETF Internet Draft, draft-ohba-hokey-3party-keydist-ps-01, March 2007.

[5] Joy Ghosh, Matthew J. Beal, Hung Q. Ngo, Chunming Qiao, On profiling mobility and predicting locations of wireless users, in: REALMAN '06: Proceedings of the Second International Workshop on Multi-hop Ad Hoc Networks: From Theory to Reality, New York, NY, USA, ACM, 2006, pp. 55–62.

[6] B. Askwith, M. Merabti, Q. Shi, K. Whiteley, Achieving user privacy in mobile networks, in: Proceedings of 13th Annual Computer Security Applications Conference, ACSAC, San diego, CA, USA, IEEE Computer Society, December 1997, pp. 108–116.

[7] R. Marin, P.J. Fernandez, A.F. Gomez, 3-Party approach for fast handover in EAP-based wireless networks, in: On the Move (OTM) Conferences, IS 2007, LNCS, vol. 4804, Vilamoura, Portugal, November 2007, pp. 1734–1751.

[8] B. Aboba, P. Calhoun, RADIUS support for EAP, IETF RFC 3579, June 2003.

[9] P. Eronen, T. Hiller, G. Zorn. Diameter extensible authentication protocol (EAP) application, IETF RFC 4072, August 2005.

[10] B. Aboba, M. Beadles, J. Arkko, P. Eronen, The network access identifier, IETF RFC 4282, December 2005.

[11] B. Aboba, D. Simon, P. Eronen, Extensible authentication protocol key management framework, RFC 5247, August 2008.

[12] R. Marin-Lopez, F. Pereniguez, Y. Ohba, F. Bernal, A.F. Skarmeta, A transport-based architecture for fast re-authentication in wireless networks, in: Proceedings of IEEE Sarnoff Symposium 2009, Princeton, USA, IEEE Computer Society Press, 2009.

[13] Y. Ohba, Q. Wu, G. Zorn, EAP early authentication problem statement, IETF Internet Draft, draft-ietf-hokey-preauth-ps-09, July 2009.

[14] R. Dantu, G. Clothier, Anuj Atri, EAP methods for wireless networks, Elsevier Computer Standards & Interfaces 29 (2007) 289–301.

[15] Y. Ohba, S. Das, R. Marin, An EAP method for EAP extension (EAP-EXT), IETF Internet Draft, draft-ohba-hokey-emu-eap-ext-02, July 2007.

[16] S. Josefsson, The base16, base32, and base64 data encodings, IETF RFC 4648, October 2006.

[17] D. Simon, B. Aboba, R. Hurst, The EAP-TLS authentication protocol, IETF RFC 5216, March 2008.

[18] J. Schaad, R. Housley, Advanced encryption standard (AES) key wrap algorithm, IETF RFC 3394, September 2004.

[19] J. Schaad, R. Housley, Wrapping a hashed message authentication code (HMAC) key with a triple-data encryption standard (DES) key or an advanced encryption standard (AES) key, IETF RFC 3537, May 2003.

[20] H. Chen, Y. Xiao, X. Hong, F. Hu, J. Xi, A survey of anonymity in wireless communication systems, Security and Communication Networks 2 (5) (2008) 427–444.

[21] J. Zhu, J. Ma, A new authentication scheme with anonymity for wireless environments, IEEE Transactions on Consumer Electronics 50 (1) (2004) 231–235.

[22] C.-C. Lee, M.-S. Hwang, I.-E. Liao, Security enhancement on a new authentication scheme with anonymity for wireless environments, IEEE Transactions on Industrial Electronics 53 (5) (2006) 1683–1687.

[23] J. Go, K. Kim, Wireless authentication protocols preserving user anonymity, in: Proceedings of the 2001 Symposium on Cryptography and Information Security (SCIS 2001), Oiso, Japan, IEEE Computer Society Press, January 2001, pp. 159–164.

[24] C.-S. Park, Authentication protocol providing user anonymity and untraceability in wireless mobile communication systems, Computer Networks 44 (2) (2004) 267–273.

[25] Y. Jiang, C. Lin, M. Shi, Mutual authentication and key exchange protocols for roaming services in wireless mobile networks, IEEE Transactions on Wireless Communications 5 (9) (2006) 2569–2577.

[26] 3GPP TS 29.230 diameter applications; 3GPP specific codes and identifiers, Release 7, September 2007, Third Generation Partnership Project.

[27] C. Tang, D. Oliver Wu, Mobile privacy in wireless networks – revisited, IEEE Transactions on Wireless Communications 7 (3) (2008) 1035–1042.

[28] W. Juang, J. Wu, Efficient 3GPP authentication and key agreement with robust user privacy protection, in: Wireless Communications and Networking Conference, WCNC 2007, Kowloon, China, March 2007, pp. 2720–2725.

[29] H. Haverinen, J. Salowey, Extensible authentication protocol method for global system for mobile communications (GSM) subscriber identity modules (EAP-SIM), IETF RFC 4186, January 2006.

[30] J. Arkko, H. Haverinen, Extensible authentication protocol method for 3rd generation authentication and key agreement (EAP-AKA). IETF RFC 4187, January 2006.