Efficient Authentication and Key Distribution in Wireless IP Networks

Luca Salgarelli, Milind Buddhikot, Juan Garay, Sarvar Patel, Scott Miller*

*Abstract— Emerging broadband access technologies such as 802.11 are enabling the introduction of wireless IP services to an increasing number of users. The market forecasts suggest that a new class of network providers, commonly referred to as Wireless Internet Service Providers (WISP), will deploy public wireless networks based on these new technologies. In order to offer uninterrupted IP service combined with ubiquitous seamless mobility, these multi-provider networks need to be integrated with each other, as well as with wide-area wireless technologies, such as third-generation CDMA-2000 and UMTS. Therefore, efficient authentication and dynamic key exchange protocols that support heterogeneous domains as well as networks with roaming agreements across trust boundaries are key to the success of wide-area wireless IP infrastructures. In this paper, we first describe a simple network model that accounts for heterogeneity in network service providers, and put forward the requirements that any authentication and key exchange protocol that operates in such model should satisfy, in terms of network efficiency, security and fraud prevention. We then introduce a new authentication and key exchange protocol, called Wireless Shared Key Exchange (*W-SKE*). We characterize properties and limitations of* W-SKE *against the requirements discussed earlier. Finally, we contrast* W-SKE *against other well-known and emerging approaches.*

1 Introduction

In recent years, ubiquitous access to IP networks has become increasingly important. Current trends indicate that wide area wireless IP networks such as the ones based on third-generation (3G) CDMA-2000 and UMTS, and local area wireless IP networks such as the ones based on IEEE 802.11 will compete and co-exist to provide such access. In fact, 802.11 has become one of the most popular and easy ways to provide wireless access to enterprises, homes and public hot spots and has seen explosive growth due to low cost of deployment.

Two key aspects common to these wireless IP technologies are: (1) authentication of the end-user or terminal by an Authentication, Authorization, Accounting (AAA) server in the network before access to the service is allowed. When service is provisioned, each user is assigned a home area and its authentication credentials are established at a AAA server called Home AAA (H-AAA). The user must be authenticated by the H-AAA before service can be accessed. (2) Encryption of the data before it is transmitted on the air interface between the base station and the user terminal. Often, symmetric encryption methods that use temporary per-session, per-user keys derived or established using data exchanged in the authentication phase are used. Each technology uses its own authentication and encryption schemes. For example, 802.11 networks at present use simple shared key authentication that relies on the end-user's terminal possessing a common shared group key. The same key is used for the Wired Encryption Privacy (WEP) method that employs RSA RC4 encryption. Similarly, 3G CDMA networks use symmetric encryption based on a shared key generated by a Home AAA server and distributed to the base station.

In wireless IP networks, when the user roams to a portion of the network different from its home area, the authentication process involves a Foreign AAA (F-AAA) server that eventually communicates to the user's H-AAA. In scenarios where the network is under the control of a single provider, F-AAA and H-AAA can trust each other completely. However, given the heterogeneity in access technologies and large number of independent service providers, seamless access to roaming customers presents additional security issues. In particular, to allow the setup of roaming agreements, security associations must be maintained between F-AAAs in visited networks and the user's H-AAA. Also, to improve performance and simplify operations, a common set of authentication credentials should be used regardless of the technology used in access networks, or who operates them. The authentication protocols that use these credentials must minimize the number of message exchanges between the end-user, F-AAA and H-AAA to achieve fast authentication and re-authentication. They must guarantee that a malicious entity listening to the protocol exchange cannot modify authentication packets in real-time or use the data contained in them at a later stage to gain fraudulent access to the service. Also, during the authentication process, it must be possible to derive cryptographically strong, per-user, per-session keys. These keys can then be used to ensure confidentiality over the air. At the same time, these keys or any other critical protocol information should not be transmitted in the clear between the involved parties. Finally, these protocols must also be implementable in standard frameworks in use in wireless IP networks, such as the Extensible Authentication Protocol (EAP) [1] and RADIUS [2]. Obviously, authentication and key establishment protocols that satisfy the above requirements are crucial to high performance, seamless mobility across wireless IP networks.

This paper introduces Wireless Shared Key Exchange (W-SKE), an authentication and key exchange protocol that meets the above requirements in a simple and elegant way. Current state-of-the-art protocols that have been standardized [3] or proposed [4, 5, 6, 7] in this area do not satisfy all the requirements briefly described above and elaborated later in this paper. In particular, none of these protocols attempt to optimize their performance in roaming scenarios, where

*L. Salgarelli, M. Buddhikot, J. Garay, S. Patel and S. Miller are with Bell Labs, Lucent Technologies, NJ USA (e-mail: salga@bell-labs.com, mbuddhikot@bell-labs.com, garay@bell-labs.com, sarvar@bell-labs.com, scm@bell-labs.com).

the latency experienced by a roaming user authenticating to its remote Home AAA must be minimized. It is also equally important not to sacrifice full compliance with the security requirements of the wireless IP networks. In this regard, to the best of our knowledge, W-SKE is a first-of-its-kind protocol, in that it meets both the objectives: while specifically designed with network efficiency in mind, W-SKE still conforms to the strictest security requirements outlined in this paper. Also, W-SKE is simple to implement using current as well as emerging standard IETF protocols, and is amenable to rigorous analysis using standard techniques such as those employed in [8, 9, 10].

1.1 Outline

Section 2 describes the wireless IP network architecture that this paper addresses, and the high-level authentication model. In Section 3 we introduce the key concepts of trust relations and intended paths, and we define the terms and assumptions that will be used for the rest of the paper. Section 4 describes the networking and security requirements for authentication and key exchange protocols that operate in the network model outlined above. In Section 5 we introduce the W-SKE protocol. Section 6 contrasts the performance and security properties of W-SKE against the requirements. In Section 7, we compare performance and conformance to the security requirements of our scheme with state-of-the art EAP methods such as EAP-TLS [3], EAP-SIM [5], EAP-AKA[6], EAP-TTLS[7] and EAP-SRP [11]. Finally, Section 8 describes the conclusions and future directions of this work.

2 Background

In this section we introduce a roaming model that is applicable to any wide area wireless IP network, and that forms the basis of our study. This architectural model is independent of the access technology, and applies equally well to emerging wireless data technologies such as CDMA-2000, 802.11, UMTS, etc. Corresponding to this high-level network architecture, we also introduce a high-level authentication model based on AAA servers. These models serve as the base premises for the definition of the W-SKE protocol.

2.1 Network architecture for roaming support

Figure 1 illustrates a generic multi-provider, multi-technology wireless IP network. Its access infrastructure is composed by two separately operated networks, one based on 802.11 and the other on wide-area 3G wireless. Even though they are based on different technologies, the two access networks are composed by elements that support similar functionalities. In the 802.11 network, Access Points (AP) manage the wireless link to the Mobile Node (MN), while a simple IP-based network connects them to the rest of the Internet. In the case of the 3G access network, Base Stations (BS) manage the wireless connectivity with the MN. Although the access infrastructure that interconnects the 3G base stations uses specialized network elements not shown in Figure 1, ultimately it connects to the rest of the Internet via a router.

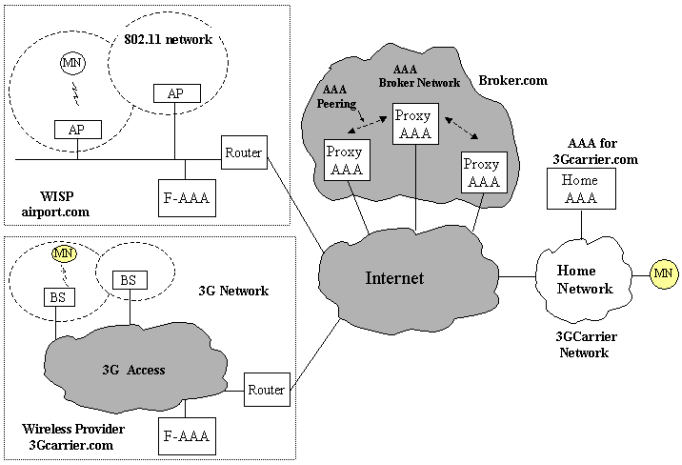

Figure 1: *Multi-provider, multi-technology wireless IP network*

Before a MN can access the network in this scenario, (1) it must be authenticated by the local AAA (F-AAA) to verify its access privileges established with a H-AAA at the time service subscription was setup; (2) temporary session keys have to be generated and distributed to the interested parties in order to enable over-the-air confidentiality and facilitate re-authentication.

Consider an example where 802.11 service is offered by Wireless Internet Service Provider (WISP) *airport.com* at Newark International Airport, while 3G wide-area coverage outside is provided by wireless carrier *3Gcarrier.com*. User John Doe is a California resident that has an account with *3Gcarrier.com*. The network operated by *3Gcarrier.com* is termed as John's home network. As a part of a service contract with his service provider, John's MN is configured with two parameters: (1) a pre-configured network access identifier (NAI, e.g. john.doe@3Gcarrier.com), or another type of user identifier such as a phone or device number, and (2) a pre-configured security association with its Home AAA server (H-AAA).

When John travels to Newark, where the airport network is operated by *airport.com*, he should be able to present his credentials to that WISP's local AAA (F-AAA) to authenticate himself and obtain network access. The access charge for this service is later posted to John's monthly access bill with his carrier *3Gcarrier.com* via a revenue settlement agreement between the two network service providers.

If John roams to a different airport where the local network is operated by another WISP, his home carrier must have a roaming agreement with that provider as well to enable John to get service. Clearly, a service provider may establish roaming agreements with a large number of other providers, and therefore may require pairwise associations for each of them. This approach is unwieldy, error-prone and leads to $O(N^2)$ overhead when establishing roaming agreements among N providers. AAA *broker* networks such as the example *broker.com* in Figure 1 simplify such peering: in this case, every network service provider instead of peering with other providers, peers (connects) only to a AAA broker network, thus reducing the number of security associations from $O(N^2)$ to $O(N)$. The AAA broker sets up appropriate secu-

rity associations and routing information within its network to route AAA messages to the appropriate H-AAA. Therefore, the path between the F-AAA of a visited WISP and the H-AAA in the home network may pass through several hops of intermediate AAA relays that are part of the broker network.

2.2 The basic authentication model

The network architecture introduced in the previous section calls for a corresponding high-level authentication model.

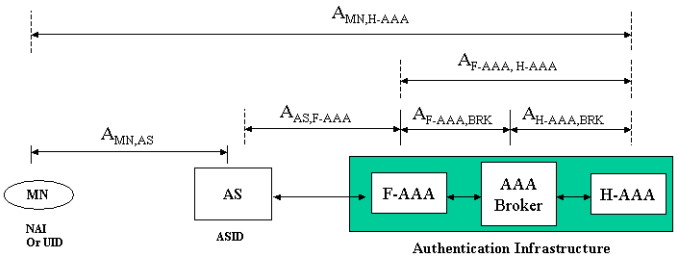

Figure 2: *High level authentication model*

Figure 2 illustrates the various network entities involved in the authentication procedure. In order to protect the exchange of data between these network nodes, several *security associations* need to be setup. At this level of detail, a security association $A_{X,Y}$ between nodes X and Y can be defined as the combination of the nodes' identity information (e.g., NAI), some form of cryptographic keys (e.g., public keys, pre-shared symmetric keys, etc.), and information on cryptographic algorithms to use in order to authenticate and/or protect the data in transit between X and Y.

There are several security associations that we need to identify in our model. Each MN shares a security association $A_{\mathrm{MN,H-AAA}}$ with its Home AAA server. In the authentication phase that precedes validated network access, the MN communicates with an Authentication System (AS) which is part of a network element such as an access point in a 802.11 network, or a combination of Base Station, Radio Network Controller (RNC) and PDSN in a CDMA-2000 network. We assume that each AS has an unique ID (ASID) that is meaningful to the MN. For example, in a 802.11 network, the ASID of an access point could be represented by its ESSID[1], e.g. `newark1.nj.airport.com`.

Each AS maintains a pre-configured security association with a local AAA server. In the roaming case we consider, the AS has an association $A_{\mathrm{AS,F-AAA}}$ with its Foreign AAA server. We also assume that a security association $A_{\mathrm{F-AAA,H-AAA}}$ exists between the F-AAA and H-AAA, which allows them to authenticate and/or encrypt each other's messages. If the F-AAA and H-AAA are part of the same network provider infrastructure, the provider sets up this association. In the case where they belong to separate providers, such an association must be set up via (1) a AAA broker, or (2) explicit pairwise setup as part of a roaming agreement. In the first case, a number of proxy AAA servers may be present in the path between the F-AAA and the H-AAA; in this case we assume that a pre-set security association exists between any pair of adjacent nodes on the network path between the AS and the H-AAA. Also, the component associations, $A_{\mathrm{F-AAA,BRK}}$, $A_{\mathrm{H-AAA,BRK}}$ are setup as a part of the agreement between the AAA broker and the home and visited domains.

One of the objectives of the authentication protocol is the setup of a temporary per-session security association and cryptographic keys between the AS and the MN, $A_{\mathrm{MN,AS}}$. These keys are then used to encrypt and authenticate the data exchanged between the AS and the MN.

Finally, this authentication model assumes that as the MN moves and attaches to different ASs, it will have to re-authenticate with the H-AAA. Theoretically this could be avoided by transferring cached authentication information between adjacent ASs and F-AAAs, therefore enabling re-authentication to be handled locally. However, currently there is no standard state-transfer protocol that could be used to achieve such functionality in a secure way in IP networks. Therefore it is reasonable to assume that in the medium term any re-authentication procedure in the types of networks considered in this paper will have to involve the H-AAA.

3 Assumptions and Definitions

In this section we will introduce the notion of *trust*, which is orthogonal to the concept of security association, and classify the expected behavior from the parties. We first define the following terms in the context of our model (Figure 2):

Insider: All the ASs and F-AAAs which share, directly or indirectly, a security association with the H-AAA are called insiders, as opposed to outsiders.

Outsider: Any network entity which does not have a direct or indirect relationship with a H-AAA is considered an outsider.

Intended-AS: The AS that the MN wants to use is called the Intended-AS. In the protocol to come, the ASID will be presented to the end-user/terminal, who/which will verify it before continuing with the protocol. Although in some cases the user might be oblivious to the point of access, the Intended-AS concept reflects those other cases where the user might be comfortable enough with one particular provider's business procedures and reputation to trust its AS to receive service from it.

Intended-Path: The Intended-Path consists of the Intended-AS, the F-AAA associated with the Intended-AS, and the optional proxy AAA servers along the path from F-AAA to H-AAA.

We now distinguish the following two cases in terms of the network entities' allowed behavior. The distinction will not only facilitate the presentation of the protocol and its analysis, but will also highlight what is needed in order to cope with "stronger" adversaries.

Case 1: Honest insiders. In this case, the entities in a security association share *full trust*, and strictly follow the

[1]The *Extended Service Set ID* is used in 802.11 networks to identify the name of each Local Area Network.

protocol. This means, in particular, that they will not divulge the information exchanged over a secure connection to third parties, try to distort session parameters so as to benefit themselves, or simply disrupt communication. Thus, in this case, the elements in the foreign network, as well as the chain of proxies, are fully trusted by the home network, on the assumption that they have valid trust relationships enforced by pre-set security associations.

Case 2: Byzantine insiders. In this case, some of the entities involved in the authentication exchange may arbitrarily deviate from the protocol and be completely malicious. Behavioral examples include revealing or misusing information from protocol exchanges, mounting so-called replay attacks, and causing fraudulent accounting.

We now provide some motivation for these cases. In a given geographic location, multiple ASs may be available from multiple wireless IP service providers; the level of security at some ASs may be different than at others. The selection of an AS from the multiple available ASs at a location can be made either by the home network (the H-AAA) or by the user (the MN). In the former model, used by the cellular providers, the home network has already decided which ASs its MN can use for network access. It is therefore the responsibility of the home network to make sure that the ASs and F-AAAs that it selects have the appropriate levels of security, and the home network tries to assure this in the context of a business relationship. Thus, in this model, the intermediaries appear as trusted, well-behaving entities; this is captured by Case 1 above, which we call the "full-trust" model.

In the second model, captured by Case 2, the user himself may select the AS from multiple available ASs in a location; this is perhaps driven by the fact that the user is comfortable with a particular provider's reputation and business procedures over others'. Such a case can commonly occur in public 802.11 networks operated in hot-spots (e.g., airports, malls). In this case, it is natural to assume that some of the insiders may misbehave. These insiders may try to use information and valid security associations maliciously to stay within the bounds of authentication protocols but steal service from ongoing valid session or overcharge an old session that completed. Although some of these attacks, such as a rogue AS cutting off communication, may not be prevented, and a practical network architecture to guarantee end-to-end security might not currently exist, we shall see in Section 6.3 how a satisfactory degree of security can be achieved under reasonable assumptions. We refer to this case as the *reduced trust* model.[2]

4 Protocol Requirements

We divide the requirements that an authentication and key exchange protocol used in roaming scenarios should satisfy into three categories: (1) Networking and system requirements, (2) Security requirements, and (3) Fraud prevention requirements.

4.1 Networking requirements

N1– *Network efficiency.* In the network model outlined in Section 2, roaming clients might find themselves logging on to foreign networks that are distant – in terms of number of hops – from the their H-AAA and therefore may experience long authentication delays. Minimizing the number of messages that the client has to exchange with its H-AAA is critical to minimizing such authentication delay. Therefore, the protocol must minimize the number of messages to be exchanged between the parties and the associated computational overhead. More precisely, since the distance between the H-AAA and the F-AAA will account for the larger portion of the end-to-end distance between the MN and the H-AAA, the protocol must minimize the number of exchanges between the F-AAA and the H-AAA. Ideally, only one message exchange should take place between the F-AAA and the H-AAA to perform authentication and key distribution. Also, the common case of successful authentication and an abnormal case of failed authentication should not differ significantly in terms of message overheads.

N2– *Implementation using existing Internet standards.* The protocol should be easily realizable using current IETF standards such as the Extensible Authentication Protocol [1] and RADIUS [2].

N3– *Statelessness.* The scheme must not require state to be maintained at the AAA servers and at the clients in between sessions. This requirement eliminates the state re-synchronization overheads incurred by stateful protocols such as UMTS AKA [6].

4.2 Security requirements

The main goal of the authentication and key distribution protocol is to mutually authenticate the user and the network to each other, and to guarantee that only the intended parties learn the session security association $A_{\mathrm{MN,AS}}$, while ensuring that the cryptographic material contained in it is fresh, random and unique. An additional requirement, specific to the roaming scenarios under consideration, is that of identification of the network path on which the session is taking place. Specifically, we would like the scheme to support the following:

S1– *Authenticate MN.* Allow H-AAA to authenticate and authorize that the MN has rights to establish a security association with, and receive service from the AS in a foreign domain with which the home domain has a direct or indirect roaming agreement.

S2– *Authenticate H-AAA.* Allow the MN to establish that it is authenticating to a trusted H-AAA with which it shares $A_{\mathrm{MN,H-AAA}}$.

[2] In a nutshell, this model will assume that any network entity can misbehave, except for the entities on the intended path. It is also possible to consider a "full adversary" model, where any entity could misbehave. Although we do not provide a full treatment of this case in this paper, we elaborate some more on this in Section 6.3.

S3– *Session key establishment.* Generate the cryptographic material (specifically, K_{SMS} – the "Session Master Secret") necessary to setup the temporary session security association $A_{\text{MN,AS}}$. Guarantee both MN and H-AAA that such material is fresh, random and unique.

S4– *Forward secrecy.* The concept of forward secrecy refers to the notion that compromise of a session key will permit access only to data protected by that key. In other words, even if an attacker is eventually able to derive the cryptographic keys that make up $A_{\text{MN,AS}}$ for one session, future (and past) session security associations (and, of course, $A_{\text{MN,H-AAA}}$) are not compromised.

S5– *Path authentication by H-AAA.* Allow the H-AAA to verify the identity of the network elements along the path from MN to H-AAA.

S6– *Path authentication by MN.* Allow the MN to verify the identity of the network elements along the path from MN to H-AAA; in particular, that of the Intended-AS.

S7– *Simplicity.* The scheme must be amenable to analysis and formal security proof.

4.3 Fraud prevention requirements

The following requirements state the fairness conditions for both the service provider and the user. Although these requirements follow from the (lower-level) security requirements from the previous section, we find it useful to state them explicitly.

F1– *Fraud protection.* Prevent unauthorized users from receiving service from visited networks.

F2– *Prevent session hijacking.* Prevent users from seizing control of a communication association (session) previously established by another user.

5 The W-SKE Protocol

W-SKE is a simple, shared key-based authentication and key exchange protocol that aims to satisfy all the requirements set forth in section 4. It follows the general techniques from the two-party shared-key model originating in [12] and further developed and analyzed in, e.g., [13, 9, 14, 15]; however, they are extended here to accommodate the scenario of relaying agents such AS and F-AAA in Figure 2. While W-SKE attempts to achieve full conformance with the above requirements, it does so in an elegant and simple way. Although specifically designed for authentication and key exchange in wireless networks for supporting roaming clients, its features may be equally appealing in other applications such as authentication in IEEE 802 wire-line LANs.

In W-SKE the security association between the MN and its H-AAA is formed by two parameters, namely the User Identifier (UID)[3], which uniquely identifies the user to the H-AAA, and the cryptographically strong secret key $K_{\text{MN,H-AAA}}$, shared between the MN and its H-AAA.

In addition to performing mutual authentication between the MN and its H-AAA, W-SKE provides for the setup of the temporary session security association between the AS and the MN. This security association takes the form of a *Session Master Secret*, K_{SMS}, which gets securely distributed to the AS by the H-AAA, and computed by the MN. Ciphersuite-specific authentication keys, initialization vectors and encryption keys can then be derived from K_{SMS} with standard algorithms such as the ones specified in section 3.5 of [3] and in [16].

In the following we describe W-SKE by detailing a successful authentication and key exchange run of the protocol. Figure 3 describes the protocol, which involves a client (MN), an AS, and a Foreign and a Home AAA (F-AAA and H-AAA). (We omit for simplicity any proxy AAA servers from the description.) Note that parameters that are optional in the exchange are identified by square brackets $[\cdot]$. The way optional parameters influence the properties of W-SKE will be discussed in detail in section 6. The steps of the exchange are as follows:

1. **MN sends a start message:** MN discovers the AS it wants to communicate with by listening to ASIDs broadcast by the AS, or by explicitly probing for the presence of the AS. It then initiates the W-SKE protocol by sending a START message.

2. **AS enquires MN ID:** AS requests the MN to present its identification.

3. MN sends its UID and session identifier SID[4] to the AS.

4. AS relays the MN response containing UID and SID to F-AAA.

5. **F-AAA presents a challenge:** F-AAA generates a nonce[5] N_1 for this session with MN and forwards it to AS.

6. AS relays the F-AAA challenge to MN.

7. **MN responds to challenge**: MN generates nonce N_2 and computes AUTH1 as follows[6]:

$$\text{AUTH1} = \text{MAC}_{K_{\text{MN,H-AAA}}}(N_1|N_2|\text{UID}|\text{SID}|[\text{ASID}]). \quad (1)$$

   MN sends $(\text{AUTH1}, N_2)$ to AS.

8. AS forwards MN's response to F-AAA.

[3]The UID uniquely identifies the user to its H-AAA. It can take the form of a NAI, a telephone number, or any unique identifier which associates a particular user to a H-AAA and a home service provider.

[4]The Session ID is chosen by the MN, and uniquely identifies the session from the MN's perspective.

[5]A nonce is a freshly generated random number.

[6]$\text{MAC}_K(\cdot)$ is a Message Authentication Code, which is applied to a piece of information for authentication using a key K. Examples include keyed cryptographic hash functions (e.g., HMAC [17], keyed-MD5, keyed-SHA-1, etc.), and block ciphers (e.g., AES in CBC-MAC mode). We use | to denote the string concatenation operator.

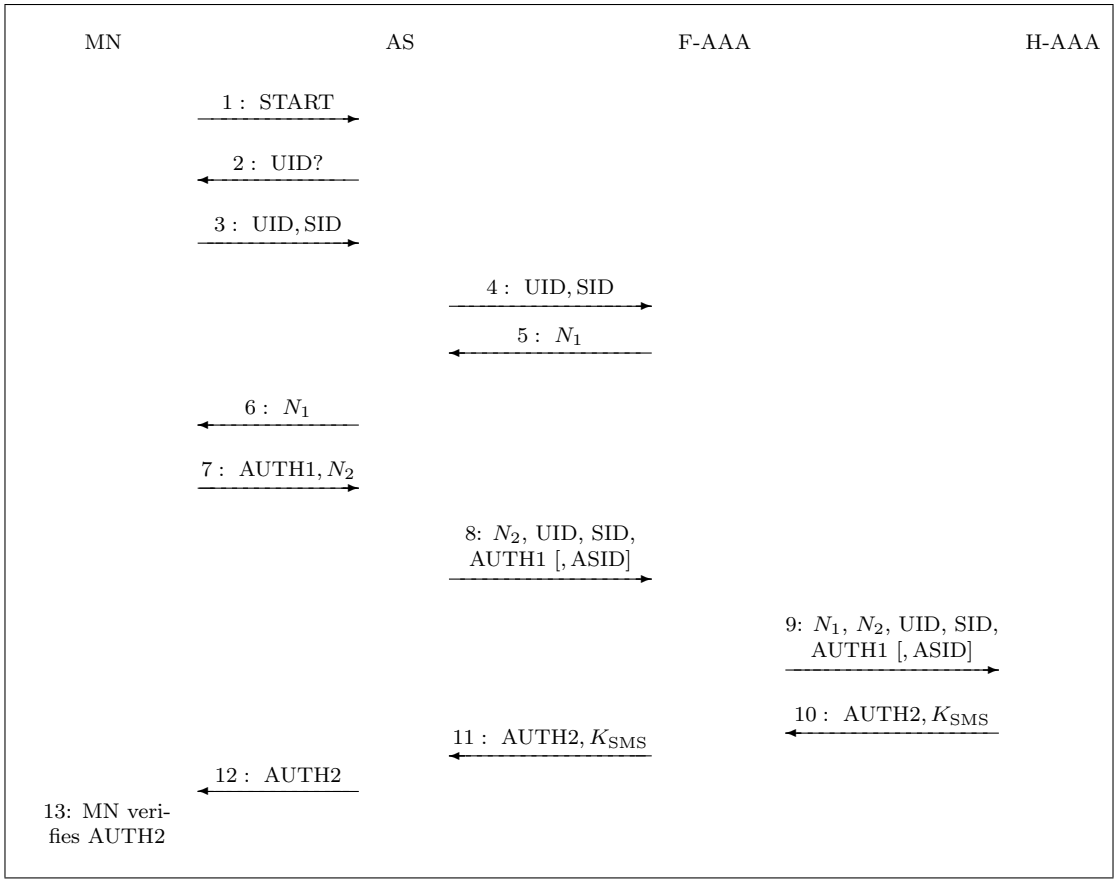

Figure 3: *The* W-SKE *protocol*

9. **F-AAA processing:** F-AAA uses (UID, SID) to verify that MN is a visiting mobile node. Using a pre-established secure channel via a AAA broker network, F-AAA forwards MN's response to the appropriate H-AAA for MN.

10. **H-AAA processing:** H-AAA performs the following steps: (1) It uses UID to look up the user credentials and access the shared secret $K_{\text{MN,H-AAA}}$. (2) Using this, and the values received from F-AAA in step 9, it computes AUTH1′ as in Equation 1. If AUTH1 = AUTH1′, then the authentication of the MN is successful. In case the two values do not match, the authentication of the MN fails, and the H-AAA responds to the F-AAA refusing access to the MN. (3) In case of successful authentication, it computes AUTH2 as follows:

$$\text{AUTH2} = \text{MAC}_{K_{\text{MN,H-AAA}}}(N2|N1|\text{UID}|\text{SID}|[\text{ASID}]) \tag{2}$$

(4) Generates the Session Master Secret, K_{SMS}, as follows[7]:

$$K_{\text{SMS}} = \text{PRF}_{K_{\text{MN,H-AAA}}}(\text{AUTH2}) \tag{3}$$

to be used by MN and AS during this session. (5) Finally, it sends a AAA message containing AUTH2 and K_{SMS} to F-AAA[8].

11. **F-AAA processing of H-AAA response:** F-AAA relays the AAA message from H-AAA to AS.

12. **AS processing of H-AAA message:** AS extracts K_{SMS} from the AAA message and forwards AUTH2 to MN.

13. **MN processing of** AUTH2**:** MN verifies AUTH2 as per Equation 2, which should successfully prove H-AAA's (and AS's) valid authentication. It then generates K_{SMS} locally using Equation 3. Note that the session master secret is not transmitted from the AS to the MN, but is locally computed.

At this point the exchange is concluded. MN and AS can start their exchange using the ciphersuite that the specific wireless technology requires, deriving the necessary keys and initialization vectors from K_{SMS}.

[7] $\text{PRF}_K(\cdot)$ represents a pseudo-random function with key K. Pseudo-random functions [18] are characterized by the pseudo-randomness of their output, namely, each bit in the output of the function is unpredictable if K is unknown. In practice, PRFs are realized using block ciphers or keyed one-way hash functions (see examples of MAC functions above).

[8] We assume, without loss of generality, that the length of AUTH2 is at least 128 bits.

6 Analysis

6.1 Network efficiency

N1– The protocol does minimize the number of messages that the MN and H-AAA have to exchange, therefore minimizing the latency of the authentication procedure. In particular, the protocol allows the exchange to complete in only one Round Trip Time (RTT) between the F-AAA and the H-AAA[9].

However, the scheme requires the MN to re-authenticate to its H-AAA every time a handoff occurs. Even 1 RTT to the H-AAA to perform re-authentication could represent too large a latency for certain environments. In such cases, further optimizations are possible, at the expense of relaxing some of the security requirements. For example, subsequent authentications between the MN and the F-AAA without involving the H-AAA could be performed by means of a MAC function keyed with K_{SMS}, and applied to a (session) counter (and the new ASID); these would get transferred among ASs by means of a context-transfer protocol such as the one being defined in the SeaMoby IETF working-group. Forward secrecy, however, would fail to hold – or at least require a more relaxed definition of a session.

N2– Refer to [19] for a detailed description of an implementation of W-SKE using standard protocols such as EAP and RADIUS.

N3– It follows from the protocol description that neither the AAA servers nor the clients keep state in-between sessions.

6.2 Security: the honest insiders case

In this section we argue how the security and fraud prevention requirements of Section 4 are satisfied; a formal proof of these requirements is beyond the scope of this paper. Recall that in the case of honest insiders, the network elements that have, directly or indirectly, a security association with the H-AAA are trusted and do not misbehave.

S1– Consider authenticator AUTH1. The nonce N_1 in the authenticator acts as challenge to MN to "prove" to H-AAA in step 10 that it possesses the pre-shared key $K_{\text{MN,H-AAA}}$. Moreover, including N_1 assures the H-AAA that the authenticator is fresh for every session. The fact that N_1 is generated by the F-AAA and not by the H-AAA does not invalidate this claim, since the F-AAA is trusted by the H-AAA by virtue of $A_{\text{H-AAA,F-AAA}}$, under the assumption of *full trust* (Honest insiders) which we are examining in this section. The included identities (i.e., the username and realm parts of the NAI) serve to reassure the parties of the correct binding between the shared key and their identities.

[9]The protocol requires the exchange of multiple messages between the MN and the F-AAA. However, given that the F-AAA and the MN are topologically close, these exchanges will not impact the overall latency of the exchange as much as the RTTs between the F-AAA and the H-AAA.

S2– Consider authenticator AUTH2 (note change in order of arguments with respect to AUTH1). Similar to the previous case, MN gets convinced in step 13 of possession of the pre-shared key $K_{\text{MN,H-AAA}}$ by the generating party, and of the authenticator's freshness, given the inclusion of N_2.

S3– The freshness and randomness of the Session Master Secret, generated according to Equation 3, follows from the freshness of AUTH2 and the properties of pseudo-random functions; specifically, the value is (computationally) independent of any other value output by the function.

S4– Forward secrecy follows from the properties of pseudo-random functions, and the fact that the protocol reveals no information to an adversary on the value of $K_{\text{MN,H-AAA}}$, with which the pseudo-random function is keyed.

S5– The security association between H-AAA and F-AAA allows the H-AAA to authenticate the F-AAA, and, transitively, the AS. This also applies to the case where proxy AAA servers are present on the path, by virtue of the chain of security associations that each intermediate AAA server shares with its peers.

S6– Even though the MN does not cryptographically authenticate the AS, the case of honest insiders precludes rogue ASs from having valid security associations with the F-AAA. Thus, successful completion of the protocol guarantees the path (Intended-AS) authenticity.

S7– The security of the protocol relies on the well-defined properties of MACs and pseudo-random functions. These transformations are carefully applied to arguments so as to guarantee authentication, session uniqueness, and key material freshness and randomness. A formal proof of the properties of this protocol can be derived using techniques similar to those employed in [8, 9, 10]. Given the space constraints and the context of this paper, a formal proof for W-SKE will be presented in a subsequent publication.

Replay attacks by illegitimate network elements (outsiders) are detected by the freshness of the authenticators, given that the nonces are freshly generated every session by MN, F-AAA and H-AAA. The fraud prevention requirements easily follow from the security properties above. Assuming that the honesty assumption on the network elements hold, the authentication of the MN guarantees the service provider that a valid user is receiving service (**F1**), while the secrecy of the Session Master Secret K_{SMS} guarantees the user that no unauthorized user will be able to hijack an existing session (**F2**).

6.3 Security: the Byzantine insiders case

The security analysis of the last section assumes that all the insiders are trustworthy. We now investigate the case where that is not the case. We first provide some motivation.

The MN and H-AAA have a security association and share a secret key. A standard two party session key agreement between them would have no further security implications.

However, in the wireless IP case that we are considering, the session key needs to be delivered to the AS, since all the encryption and message integrity happens at the AS. This introduces questions about session security and service fraud if some of the insiders misbehave. In the general model of Figure 2 it is not possible to make security guarantees without unreasonable assumptions. However, in the *direct association* model, where the F-AAA has a direct security association with the H-AAA without relying on intermediary AAA-brokers, the W-SKE protocol provides the security guarantees S1-S7 under reasonable and practical assumptions.

Recall from Section 3 that the current model captures situations where the user himself selects the AS. Since ASIDs are broadcast but not cryptographically authenticated, displaying and "verifying" the ASID is not sufficient for session security, as a rogue AS could overpower the Intended-AS's signal, and then use its own ID when communicating with its F-AAA or H-AAA. Thus, we enhance the protocol of Figure 3 by making the optional ASID parameter mandatory in the AUTH1 and AUTH2 calculations.

Furthermore, we make the additional assumptions that (1) the H-AAA "knows" the list of ASs that are associated with a particular F-AAA;[10] and (2) the entities *on the intended path* are trustworthy. Both assumptions are required in this case because in the communication model we are considering there is no direct security association between the AS and the H-AAA; if there were, then the H-AAA would discover the inconsistencies created by a rogue AS when verifying AUTH1 (thus making (1) unnecessary), and no rogue F-AAA on the path would have access to the session key (solving (2)).

We first elaborate on how the path authentication requirements are satisfied.

S5– The H-AAA authenticates the F-AAA through a direct security association between them, and checks that the ASID forwarded by the F-AAA in step 9 belongs to the AS list associated with it. (If it doesn't, then the session is terminated.) It then uses this ASID in the computation of AUTH1′. Equality of this quantity to AUTH1 implies that the ASID is MN's Intended-AS. Note that this also precludes any AS (legitimate or rogue) other than the Intended-AS from overpowering or posing as the Intended-AS.

S6– The MN does not cryptographically authenticate the AS, so it is possible that even ASs having a security association with the F-AAA could impersonate the Intended-AS. However, as argued above, this would be detected by the H-AAA. Thus, successful verification of AUTH2 assures the MN of the path (and in particular of the Intended-AS's) authenticity.

The remaining security properties now follow similarly to the case of Section 6.2. More specifically, successful computation of AUTH1 (resp., AUTH2) allows the H-AAA (resp., the MN) to authenticate the party in possession of $K_{\mathrm{MN,H-AAA}}$. With respect to service fraud, the case of corrupt insiders allows for an extended set of possibilities, since we need to consider situations such as collusions between insiders and outsiders (e.g., an MN impersonator), or insiders (ASs) trying to steal customers from other insiders. Again, the security properties render these attacks futile.

However, the case where (AS,F-AAA) pairs collude trying to overcharge the H-AAA deserves special attention. This kind of collusion would allow an AS (F-AAA) to record and re-play a session's parameters, therefore enabling the F-AAA to present multiple false accounting claims to the H-AAA for its users.

An obvious fix to this is achieved by having the random challenge N_1 generated at the H-AAA, thus guaranteeing the freshness of the new session; this however would happen at the expense of network efficiency, since it would increase the number of RTTs between the F-AAA and the H-AAA necessary to complete the procedure.

In practical terms, since the relationship between the foreign and the home network is supposedly regulated by a business agreement, it should not be necessary to adopt this fix in commercial networks. In fact, we would argue that frauds perpetrated by dishonest F-AAAs trying to overcharge the H-AAA could hardly be prevented by the authentication protocol alone. For example, even in the case where the fix discussed above were to be implemented, nothing would stop the F-AAA from falsely accounting twice the traffic that the MN actually sends or receive. Other mechanisms such as systematic audits on the network usage, or even authentication of the traffic sent or received by the MN are necessary to prevent these kind of frauds.

7 Comparison of W-SKE with the State-of-the-Art

W-SKE has been designed for Wireless IP networks, such as those based on 802.11. Recently, authentication mechanisms for such networks have begun to rely on the Extensible Authentication Protocol (EAP) [1] as a basis to transfer authentication information between the client and the network. EAP provides a basic request/response protocol framework over which to implement a specific authentication and/or key exchange algorithm. When a security algorithm gets implemented over EAP, it is referred to as an *EAP method*. As with other authentication and key distribution protocols, W-SKE is easily implementable as an EAP method [19], without diverting substantially from the general protocol outlined in Figure 3.

In this section, we briefly compare EAP-SKE, the EAP implementation of W-SKE with other approaches. Relevant to this comparison are such protocols that can be implemented over EAP, and whose objectives are comparable with those of W-SKE. In particular, we will consider protocols that, as a minimum, can provide mutual authentication between the client and its home network, and that are already published standards, or have been submitted to standard bodies for ratification. The EAP methods that we will contrast with EAP-SKE are the following: SIM [5], AKA [6], TLS [3], TTLS [7]

[10] There are various ways for the H-AAA to know the list of ASs associated with an F-AAA. For example, there could be a direct communication between F-AAA to H-AAA to convey the list, perhaps at the same time as when the security association is established. Alternatively, the AS list could be part of the F-AAAs digital certificate, communicated to and verified by the H-AAA once (or periodically), thus making the associated overhead negligible.

and SRP [11].

Scheme	Architecture	Networking properties	
		RTT F-AAA / H-AAA	Statelessness
EAP-SKE	Shared key with H-AAA	1	Yes
EAP-SIM	Subscriber Identity Module (SIM) card	3	Yes
EAP-AKA	Universal SIM (USIM) card	2+	No
EAP-TLS	Public-private key based Certificates	3	Yes
EAP-TTLS	Public-private key based Certificates + other	4+	Yes
EAP-SRP	Password	4	Yes

Table 1: *Comparison with other approaches: architecture and networking properties*

Because of space constraints, we do not describe the details of each of the methods. Instead, Table 1 reports the principal mechanism on which each method is based, and its main networking characteristics.

From the data in the table, it is clear that EAP-SKE is characterized by the lowest latency, since it requires only one round-trip between the F-AAA and the H-AAA to perform mutual authentication and key distribution. The protocol which is closer to EAP-SKE in terms of latency is EAP-AKA, with at least 2 RTTs. However, this figure for AKA is the best case performance number. In fact, AKA being a stateful protocol[11], it potentially requires up to 5 round-trips to re-synchronize the state when the counters at the MN and H-AAA get out of sync.

The value of 4 RTTs reported for TTLS is also a minimum. In TTLS, a pre-configured authentication and key exchange mechanism is run between the client and the H-AAA over a TLS tunnel. While the TLS tunnel is established using Certificates, other mechanisms such as a shared password or a One Time Password can then be used for the end-to-end authentication. Therefore, the actual total number of exchanges is the sum of the 3 RTTs required to setup the TLS tunnel, plus the number of exchanges required to perform the tunneled algorithm, which is at least one.

The rest of the protocols in Table 1 are characterized by latencies that vary from 3 RTTs for EAP-SIM and EAP-TLS to 4 for EAP-SRP. Since these protocols are stateless, they do not suffer from the re-synchronization problem that affects EAP-AKA.

Table 2 reports the security properties of the EAP methods we considered. All of the listed protocols can basically provide for mutual authentication and session key generation, along with forward secrecy. However, only two protocols claim to have proofs of security, or to be amenable to proof. A security proof for AKA has been presented in [20]. As previously mentioned, we believe that a formal security proof for the W-SKE protocol can be derived using techniques similar to those employed in [8, 9, 10]. Given the complexity of the other protocols of Table 2, we argue that it would be much more difficult, if not impossible, to derive a formal proof of their security properties.

[11] AKA requires state, in the form of a synchronized counter, to be kept between sessions at the MN and H-AAA.

MNs using EAP-TLS would need a public key/private key pair to authenticate themselves to the server. Current implementations of SSL/TLS in web browsers allow the user to override certain failures of certificate verification which can leave uninformed users vulnerable to a security threat. EAP-TLS implementations need to be extra careful about allowing such override mechanism.

Scheme	Security properties			
	Session key establishment	Forward secrecy	Path authentication	Security proof
EAP-SKE	Yes	Yes	Yes	Amenable to proof
EAP-SIM	Yes	Yes	No	No
EAP-AKA	Yes	Yes	No	Yes
EAP-TLS	Yes	Yes	No	No
EAP-TTLS	Depending on tunneled method	Depending on tunneled method	No	No
EAP-SRP	Yes	Yes	No	No

Table 2: *Comparison with other approaches: security properties*

EAP-TTLS and EAP-SRP can be used with weak shared keys (e.g. passwords) and still resist to offline dictionary attacks. All three protocols EAP-TLS, EAP-TTLS, and EAP-SRP employ public key operations (e.g. exponentiation) which can be a order of magnitude slower than only relying on shared key based operations, as done by the other three protocols. EAP-SIM is based on GSM-triplet generation and requires the strong assumption that no GSM triplet will ever be compromised [5].

All of the protocols except EAP-SKE assume that there are no intermediaries involved in the generation of quantities used to perform authentication and/or key generation. In EAP-SKE, one of such quantities (N_1) is generated at the F-AAA. However, EAP-SKE, under reasonable assumptions, can provide meaningful security guarantees even with dishonest intermediaries as explained in section 6. Compared to other protocols, EAP-SKE offers a unique combination of efficiency (i.e. single RTT, fast shared key operations and statelessness) and security (i.e. amenability to formal proof and path authentication).

8 Conclusions

In this paper we detailed the networking and security requirements of authentication and key exchange protocols that operate in wireless IP networks with roaming clients. We introduced W-SKE, a simple and elegant authentication and key exchange protocol, and showed how it satisfies both the networking requirements and the security requirements described above. In particular, we emphasized the analysis of the security properties of W-SKE under a range of security assumptions that characterize evolving heterogeneous wireless

IP networks. We contrasted EAP-SKE, an implementation of W-SKE over the Extensible Authentication Protocol, with other approaches based on EAP.

Applied to today's wireless IP technologies like 802.11, W-SKE offers an ideal combination of efficiency properties such as single RTT, low-overhead authentication and key distribution, and security properties such as path authentication and formal proofs.

Our on-going work includes a formal proof of the security properties of W-SKE, further optimizations to W-SKE's network efficiency, particularly to reduce the latency of re-authentication, and the study of mechanisms to allow authentication credentials other than shared keys (e.g. public keys) to work with W-SKE. The integration of W-SKE with layer-3 mobility mechanisms such as Mobile-IP [21] and its key distribution mechanisms offer other interesting research possibilities.

References

[1] L. Blunk and J. Volbrecht. PPP Extensible Authentication Protocol (EAP). RFC 2284, IETF, March 1998.

[2] C. Rigney et. al. Remote Authentication Dial In User Service (RADIUS). RFC 2865, IETF, June 2000.

[3] B. Aboba and D. Simon. PPP EAP TLS Authentication Protocol. RFC 2716, IETF, October 1999.

[4] R. Molva, D. Samfat, and G. Tsudik. Authentication of mobile users. *IEEE Network*, 8(2), 1994.

[5] H. Haverinen (Editor). EAP SIM Authentication. Work in progress - Internet Draft, IETF, June 2003. draft-haverinen-pppext-eap-sim-11.txt. This is an evolving draft standard. For its latest version, refer to the site http://www.ietf.org.

[6] J. Arkko and H. Haverinen. EAP AKA Authentication. Work in progress - Internet Draft, IETF, June 2003. draft-arkko-pppext-eap-aka-09.txt. This is an evolving draft standard. For its latest version, refer to the site http://www.ietf.org.

[7] P. Funk and S. Blake-Wilson. EAP Tunneled TLS Authentication Protocol (EAP-TTLS). Work in progress - Internet Draft, IETF, November 2002. draft-ietf-pppext-eap-ttls-02.txt. This is an evolving draft standard. For its latest version, refer to the site http://www.ietf.org.

[8] M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. *STOC'98[38]*, pages 419–428, 1998.

[9] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In *Advances in Cryptology—CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 232–249. Springer-Verlag, 22–26 August 1993.

[10] V. Shoup. On formal models for secure key exchange. In *Proc. 6th Annual ACM Conf. on Computer and Communications Security (invited talk)*, available from `http://www.shoup.net/papers/skey.ps`, 1999.

[11] J. Carlson, B. Aboba, and H. Haverinen. PPP EAP SRP-SHA1 Authentication Protocol. Work in progress - Internet Draft, IETF, July 2001. draft-ietf-pppext-eap-srp-03.txt. This is an evolving draft standard. For its latest version, refer to the site http://www.ietf.org.

[12] R. Needham and M. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21:993–999, 1978.

[13] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung. Systematic Design of a family of attack-resistant authentication protocols. *IEEE Journal on Selected Areas in Communications (special issue on Secure Communications)*, 11(5):679–693, 1993.

[14] P. Cheng, J. Garay, A. Herzberg, and H. Krawczyk. A security architecture for the Internet Protocol. *IBM Systems Journal (special issue on the Internet)*, 37(1):42–60, 1998.

[15] H. Krawczyk. SKEME: A versatile secure key exchange mechanism for the Internet. In *Proc. 1996 Internet Society Symposium on Network and Distributed System Security*, pages 114–127, Feb. 1996.

[16] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409, IETF, November 1998.

[17] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. RFC 2104, IETF, February 1997.

[18] O. Goldreich, S. Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(169):210–217, 1986.

[19] U. Blumenthal, M. Buddhikot, J. Garay, S. Miller, S. Patel, L. Salgarelli, and D. Stanley. A Scheme for Authentication and Dynamic Key Exchange in Wireless Networks. *Bell Labs Technical Journal*, 7(2):37–48, 2002.

[20] Formal Analysis of 3G Authentication Protocol. TS 33.902, ETSI, 2002.

[21] C. Perkins (Editor). IP Mobility Support for IPv4. RFC 3344, IETF, August 2002.