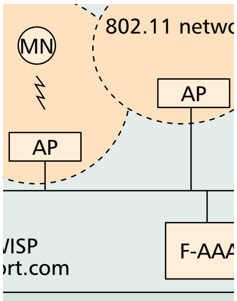
EFFICIENT AUTHENTICATION AND KEY DISTRIBUTION IN WIRELESS IP NETWORKS

LUCA SALGARELLI, MILIND BUDDHIKOT, JUAN GARAY, SARVAR PATEL, AND SCOTT MILLER,
BELL LABORATORIES, LUCENT TECHNOLOGIES



Emerging broadband access technologies such as 802.11 are enabling the introduction of wireless IP services to an increasing number of users. 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.

ABSTRACT

Emerging broadband access technologies such as 802.11 are enabling the introduction of wireless IP services to an increasing number of users. Market forecasts suggest that a new class of network providers, commonly referred to as wireless Internet service providers, will deploy public wireless networks based on these new technologies. In order to offer uninterrupted IP service combined with ubiquitous seamless mobility, these multiprovider 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 article we first describe a simple network model that accounts for heterogeneity in network service providers, and put forward the requirements any authentication and key exchange protocol that operates in such a model should satisfy, in terms of network efficiency, security, and fraud prevention. We then introduce a new authentication and key exchange protocol, 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.

INTRODUCTION

In recent years, ubiquitous access to IP networks has become increasingly important. Current trends indicate that wide-area wireless IP networks such as those based on third-generation (3G) CDMA-2000 and Universal Mobile Telecommunications System (UMTS), and local area wireless IP networks such as those based on IEEE 802.11 will compete and coexist to provide such access. In fact, 802.11 has become one of the most popular and easiest ways to provide wireless access to enterprises, homes, and public hotspots, 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, and 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 a 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 code-division multiple access (CDMA) networks use symmetric encryption based on a shared key generated by an H-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, the 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 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

1536-1284/03/$17.00 © 2003 IEEE

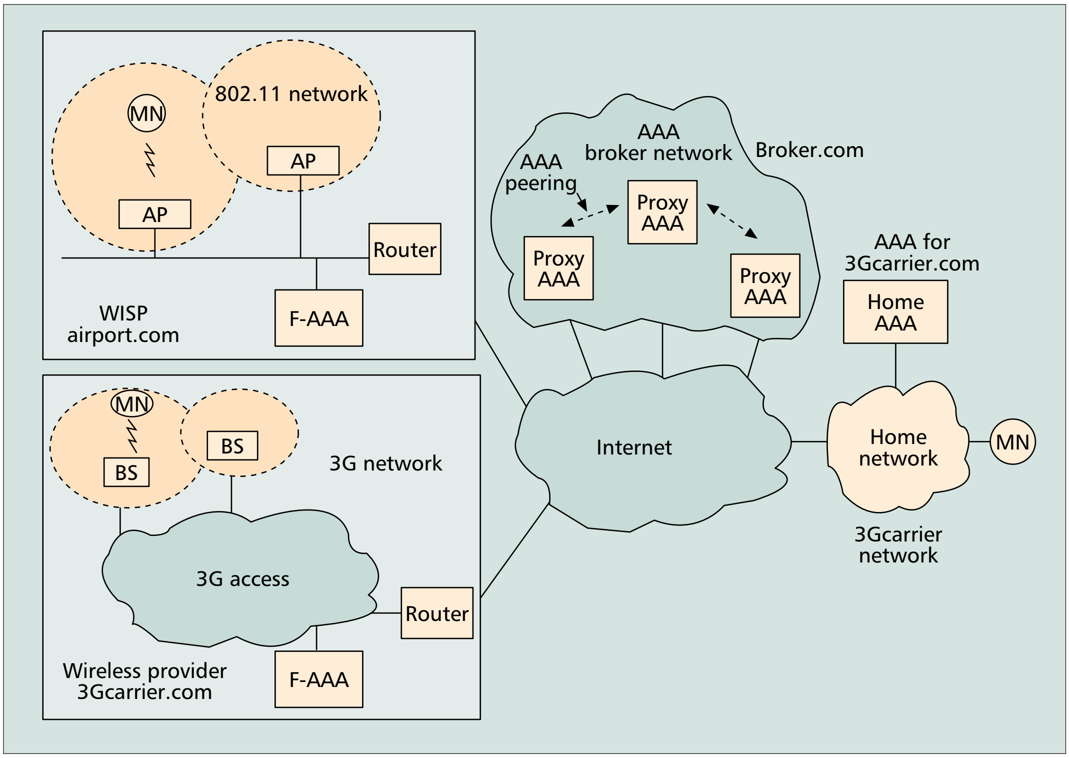


Figure 1. *A multiprovider multitechnology wireless IP network.*

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 article.

the end user, F-AAA, and H-AAA to achieve fast authentication and reauthentication. 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 and seamless mobility across wireless IP networks.

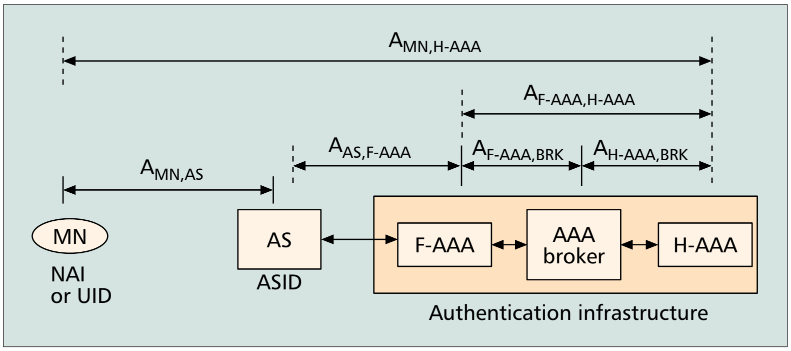
This article introduces Wireless Shared Key Exchange (W-SKE), an authentication and key exchange protocol that meets the above requirements in a simple and elegant way. Current state-of-the-art protocols that have been standardized [3] or proposed [4–7] in this area do not satisfy all the requirements briefly described above and elaborated on later in this article. In particular, none of these protocols attempt to optimize their performance in roaming scenarios, where the latency experienced by a roaming user authenticating to its remote H-AAA must be minimized. It is also equally important not to sacrifice full compliance with the security requirements of 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 objectives: while specifically designed with network efficiency in mind, W-SKE still conforms to the strictest security requirements outlined in this article. Also, W-SKE is simple to implement using current as well as emerging standard Internet Engineering Task Force (IETF) protocols, and is amenable to rigorous analysis using standard techniques (e.g., those employed in [8–10]).

Background

In this section we introduce a roaming model applicable to any wide-area wireless IP network, which 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, and UMTS. Corresponding to this high-level network architecture, we also introduce a high-level authentication model based on AAA servers. These models serve as the basis for the definition of the W-SKE protocol.

Network Architecture for Roaming Support

Figure 1 illustrates a generic multiprovider multitechnology wireless IP network. Its access infrastructure is composed of 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 of elements that support similar functionalities. In the 802.11 network, access points (APs) 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 (BSs) manage the wireless



■ Figure 2. *The high-level authentication model.*

connectivity with the MN. Although the access infrastructure that interconnects the 3G base stations uses specialized network elements not shown in Fig. 1, ultimately it connects to the rest of the Internet via a router.

Before an MN can access the network in this scenario:

- It must be authenticated by the local AAA (F-AAA) to verify its access privileges established with an H-AAA at the time service subscription was set up.
- Temporary session keys have to be generated and distributed to the interested parties in order to enable over-the-air confidentiality and facilitate reauthentication.

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 who has an account with *3Gcarrier.com*. The network operated by *3Gcarrier.com* is John's home network. As a part of a service contract with his service provider, John's MN is configured with two parameters:

- A preconfigured network access identifier (NAI, e.g., john.doe@3Gcarrier.com) or another type of user identifier such as a phone or device number
- A preconfigured security association with its H-AAA server

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 Fig. 1 simplify such peering: in this case, every network service provider, instead of peering with other providers, peers (connects) only to an AAA broker network, thus reducing the number of security associations from $O(N^2)$ to $O(N)$. The AAA broker sets up appropriate security 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.

The Basic Authentication Model

The network architecture introduced in the previous section calls for a corresponding 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 set up. 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., NAIs), some form of cryptographic keys (e.g., public keys, preshared symmetric keys), and information on cryptographic algorithms to use in order to authenticate and/or protect data in transit between X and Y.

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

Each AS maintains a preconfigured security association with a local AAA server. In the roaming case we consider, the AS has an association $A_{AS,F–AAA}$ with its F-AAA server. We also assume that a security association $A_{F–AAA,H–AAA}$ 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. Where they belong to separate providers, such an association must be set up via a AAA broker, or an 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 H-AAA; in this case we assume that a preset security association exists between any pair of adjacent nodes on the network path between the AS and the H-AAA. Also, the component associations, $A_{F–AAA,BRK}$, $A_{H–AAA,BRK}$ are set up as part of the agreement between the AAA broker and the home and visited domains.

One of the objectives of the authentication protocol is setup of a temporary per-session security association and cryptographic keys between the AS and the MN, $A_{MN,AS}$. These

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

keys are then used to encrypt and authenticate data exchanged between the AS and MN.

Finally, this authentication model assumes that as the MN moves and attaches to different ASs, it will have to reauthenticate with the H-AAA. Theoretically, this could be avoided by transferring cached authentication information between adjacent ASs and F-AAAs, therefore enabling reauthentication 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 reauthentication procedure in the types of networks considered in this article will have to involve the H-AAA.

Assumptions and Definitions

In this section we 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 (Fig. 2):

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

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

Intended-AS: The AS 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, 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 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 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 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 preset 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 WISPs; 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 by either the home network (the H-AAA) or the user (the MN). In the former model, used by 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 it selects have the appropriate levels of security, and the home network tries to ensure 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 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 hotspots (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 an ongoing valid session or overcharge an old session that has 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 later how a satisfactory degree of security can be achieved under reasonable assumptions. We refer to this case as the *reduced trust* model.[2]

Protocol Requirements

We divide the requirements an authentication and key exchange protocol used in roaming scenarios should satisfy into three categories:

- Networking and system requirements
- Security requirements
- Fraud prevention requirements

Networking Requirements

N1 — Network Efficiency. In the network model outlined earlier, roaming clients might find themselves logging on to foreign networks that are distant — in terms of number of hops — from their H-AAA and therefore may experience long authentication delays. Minimizing the number of messages the client has to exchange with its H-AAA is critical to minimizing such authentication delays. 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 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 H-AAA.

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. These keys are then used to encrypt and authenticate the data exchanged between the AS and the MN.

[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 article, we elaborate some more on this later.*

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.

Ideally, only one message exchange should take place between the F-AAA and 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 easily be realizable using current IETF standards such as EAP [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 resynchronization overheads incurred by stateful protocols such as UMTS AKA [6].

Security Requirements

The main goal of the authentication and key distribution protocol is to mutually authenticate the user and network to each other, and to guarantee that only the intended parties learn the session security association $A_{MN,AS}$, 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 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 the H-AAA to authenticate and authorize the MN with 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 $A_{MN,H\text{-}AAA}$.

S3 — Session Key Establishment. Generate the cryptographic material (specifically, the Session Master Secret, K_{SMS}) necessary to set up the temporary session security association $A_{MN,AS}$. Guarantee for 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 $A_{MN,AS}$ for one session, future (and past) session security associations (and, of course, $A_{MN,H\text{-}AAA}$) 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.

Fraud Prevention Requirements

The following requirements state the fairness conditions for both the service provider and user. Although these requirements follow from the (lower-level) security requirements of 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.

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 earlier. It follows the general techniques of the two-party shared-key model originated in [12] and further developed and analyzed in, e.g., [9, 13–15]; however, they are extended here to accommodate the scenario of relaying agents such as AS and F-AAA in Fig. 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 wireline LANs.

In W-SKE the security association between the MN and its H-AAA is formed by two parameters: the user identifier (*UID*),[3] which uniquely identifies the user to the H-AAA, and the cryptographically strong secret key $K_{MN,H\text{–}AAA}$, 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 is 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 those specified earlier [3] and in [16].

Here 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 an F-AAA and H-AAA. (We omit for simplicity any proxy AAA servers from the description.) Note that parameters optional in the exchange are identified by square brackets [·]. The way optional parameters influence the properties of W-SKE will be discussed in detail later. The steps of the exchange are as follows:

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

[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.*

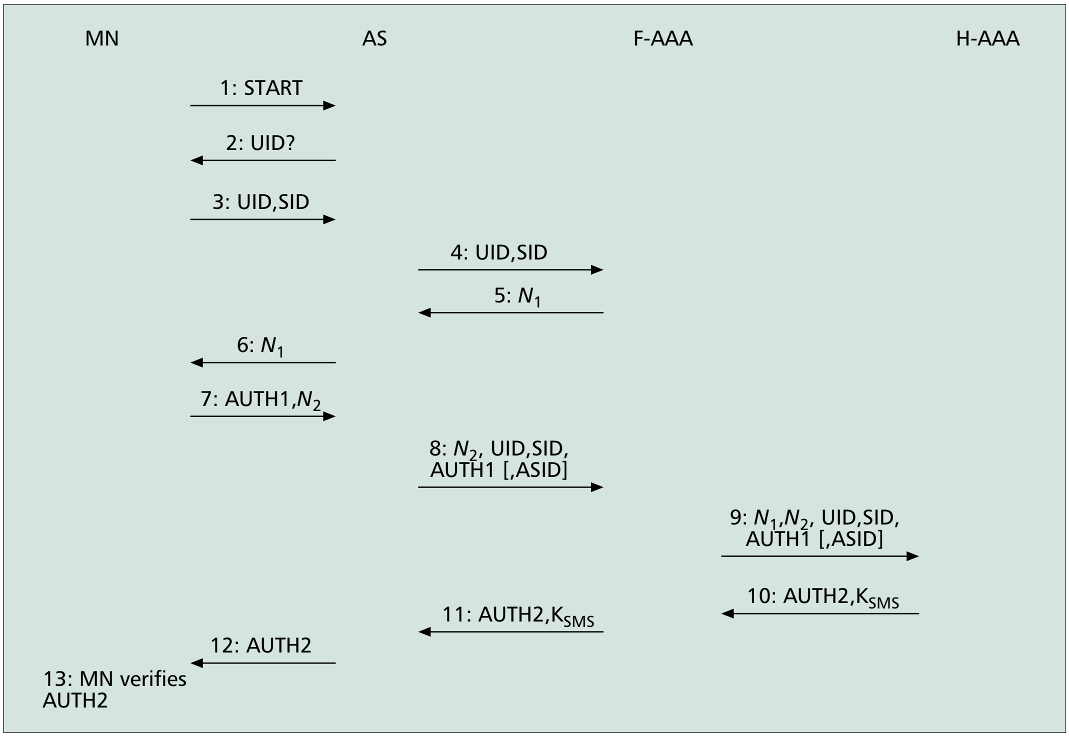


Figure 3. *The W-SKE protocol.*

5. **F-AAA presents a challenge:** F-AAA generates a nonce[5] N1 for this session with MN and forwards it to the AS.
6. AS relays the F-AAA challenge to the MN.
7. **MN responds to challenge.** MN generates nonce N_2 and computes AUTH1 as follows[6]:

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

   MN sends (AUTH1, N_2) to AS.

8. AS forwards the MN's response to the F-AAA.
9. **F-AAA processing:** F-AAA uses (UID, SID) to verify that the MN is a visiting MN. Using a pre-established secure channel via a AAA broker network, the F-AAA forwards the MN's response to the appropriate H-AAA for the MN.
10. **H-AAA processing:** The H-AAA performs the following steps:
    - It uses UID to look up the user credentials and access the shared secret $K_{\mathrm{MN,H\text{–}AAA}}$
    - Using this, and the values received from F-AAA in step 9, it computes AUTH1′ as in Eq. 1. If AUTH1 = AUTH1′, the authentication of the MN is successful. If 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.
    - In case of successful authentication, it computes AUTH2 as follows:

    $$\mathrm{AUTH2} = \mathrm{MAC}_{K_{\mathrm{MN,H\text{–}AAA}}}(N_2|N_1|\mathrm{UID}|\mathrm{SID}|[\mathrm{ASID}]) \quad (2)$$

    - Generates the K_{SMS} as follows:[7]

    $$K_{SMS} = \mathrm{PRF}_{K_{\mathrm{MN,H\text{–}AAA}}}(\mathrm{AUTH2}) \quad (3)$$

    to be used by the MN and AS during this session.
    - Finally, it sends a AAA message containing AUTH2 and K_{SMS} to the F-AAA.[8]
11. **F-AAA processing of H-AAA response:** F-AAA relays the AAA message from the H-AAA to the AS.
12. **AS processing of H-AAA message:** AS extracts K_{SMS} from the AAA message and forwards AUTH2 to the MN.
13. **MN processing of AUTH2:** MN verifies AUTH2 as per Eq. 2, which should successfully prove the H-AAA's (and AS's) valid authentication. It then generates K_{SMS} locally using Eq. 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. The MN and AS can start their exchange using the ciphersuite the specific wireless technology requires, deriving the necessary keys and initialization vectors from K_{SMS}.

Analysis

Network Efficiency

N1 — The protocol does minimize the number of messages 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.

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

[6] *MACK(·) 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.*

[7] *PRFK(·) 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.*

[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.*

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.

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 between sessions.

Security: The Honest Insiders Case

In this section we argue how the security and fraud prevention requirements discussed earlier are satisfied; a formal proof of these requirements is beyond the scope of this article. 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 a challenge to the MN to "prove" to the H-AAA in step 10 that it possesses the preshared key $K_{\mathrm{MN,H\text{-}AAA}}$. 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 $A_{\mathrm{H\text{-}AAA,F\text{-}AAA}}$, under the assumption of *full trust* (honest insiders) 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.

S2 — Consider authenticator AUTH2 (note change in order of arguments with respect to AUTH1). Similar to the previous case, the MN gets convinced in step 13 of possession of the preshared key $K_{\mathrm{MN,H\text{-}AAA}}$ by the generating party, and of the authenticator's freshness, given the inclusion of N2.

S3 — The freshness and randomness of the session master secret, generated according to Eq. 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 $K_{\mathrm{MN,H–AAA}}$, 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 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 the arguments 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 [8–10]. Given the space constraints and context of this article, 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 the 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 for the user that no unauthorized user will be able to hijack an existing session (**F2**).

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 so. 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 we are considering, the session key needs to be delivered to the AS, since all 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 Fig. 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 earlier that the current model captures situations where the user 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 Fig. 3 by making the optional ASID parameter mandatory in the AUTH1 and AUTH2 calculations.

Furthermore, we make the additional assumptions that:

- The H-AAA "knows" the list of ASs associated with a particular F-AAA.[10]
- The entities *on the intended path* are trustworthy.

[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.*

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 H-AAA; if there were, the H-AAA would discover the inconsistencies created by a rogue AS when verifying AUTH1 (thus making the first assumption unnecessary), and no rogue F-AAA on the path would have access to the session key (solving the second assumption).

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, 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 the 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's (in particular the Intended-AS's) authenticity.

The remaining security properties now follow similar to the case of the honest insiders. More specifically, successful computation of AUTH1 (resp. AUTH2) allows the H-AAA (resp. the MN) to authenticate the party in possession of $K_{\mathrm{MN,H\text{-}AAA}}$. With respect to service fraud, the case of corrupt insiders allows 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 replay 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; but this would happen at the expense of network efficiency, since it would increase the number of RTTs between the F-AAA and H-AAA necessary to complete the procedure.

In practical terms, since the relationship between the foreign and home networks 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 if the fix discussed above were to be implemented, nothing would stop the F-AAA from falsely accounting twice the traffic the MN actually sends or receives. Other mechanisms such as systematic audits on network usage, or even authentication of the traffic sent or received by the MN are necessary to prevent these kinds of fraud.

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 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 Fig. 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 consider protocols that, as a minimum, can provide mutual authentication between the client and its home network, and are already published standards, or have been submitted to standard bodies for ratification. The EAP methods we will contrast with EAP-SKE are the following: SIM [5], AKA [6], TLS [3], TTLS [7], and SRP [11].

Because of space constraints, we do not describe the details of each method. 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 closest to EAP-SKE in terms of latency is EAP-AKA, with at least two RTTs. However, this figure for AKA is the best case performance number. In fact, AKA, being a stateful protocol,[11] potentially requires up to five round-trips to resynchronize the state when the counters at the MN and H-AAA get out of sync.

The value of four RTTs reported for TTLS is also a minimum. In TTLS, a preconfigured 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 end-to-end authentication. Therefore, the actual total number of exchanges is the sum of the three RTTs required to set up 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 char-

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 EAP as a basis to transfer authentication information between the client and the network.

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

		Networking properties	
Scheme	Architecture	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.*

acterized by latencies that vary from three RTTs for EAP-SIM and EAP-TLS to four for EAP-SRP. Since these protocols are stateless, they do not suffer from the resynchronization 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 mutual authentication and session key generation, along with forward secrecy. However, only two protocols claim to have proofs of security or be amenable to proof. A security proof for AKA was presented in [20]. As previously mentioned, we believe a formal security proof for W-SKE can be derived using techniques similar to those employed in [8–10]. Given the complexity of the other protocols in Table 2, we argue that it would be much more difficult, if not impossible, to derive a formal proof of their security properties.

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 an override mechanism.

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) that can be an 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 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 such quantity (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 earlier. 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).

Conclusions

In this article we detail the networking and security requirements of authentication and key exchange protocols that operate in wireless IP networks with roaming clients. We introduce W-SKE, a simple and elegant authentication and key exchange protocol, and show how it satisfies both the networking and security requirements described above. In particular, we emphasize the analysis of the security properties of W-SKE under a range of security assumptions that characterize evolving heterogeneous wireless IP networks. We contrast EAP-SKE, an implementation of W-SKE over EAP, 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 ongoing 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

	Security properties			
Scheme	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.*

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)," IETF RFC 2284, Mar. 1998.

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

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

[4] R. Molva, D. Samfat, and G. Tsudik, "Authentication of Mobile Users," *IEEE Network*, vol. 8, no. 2, 1994.

[5] H. Haverinen, Ed., "EAP SIM Authentication," Internet draft, IETF, June 2003, draft-haverinen- pppext-eap-sim-11.txt; for the latest version, see http://www.ietf.org

[6] J. Arkko and H. Haverinen, "EAP AKA Authentication," Internet draft, IETF, June 2003, draft-arkko-pppext-eap-aka-09.txt; for the latest version see http://www.ietf.org

[7] P. Funk and S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol (EAP-TTLS)," Internet draft, IETF, Nov. 2002; draft-ietf-pppexteap-ttls-02.txt; for the latest version see 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*, 1998, pp. 419–28.

[9] M. Bellare and P. Rogaway, "Entity Authentication and Key Distribution," *Advances in Cryptology— CRYPTO '93, LNCS*, vol. 773, 22–26 Aug. 1993, pp. 232–49.

[10] V. Shoup, "On Formal Models for Secure Key Exchange," *Proc. 6th Annual ACM Conf. Comp. and Commun. Security* (invited talk); http://www.shoup.net/papers/skey.ps, 1999.

[11] J. Carlson, B. Aboba, and H. Haverinen, "PPP EAP SRP-SHA1 Authentication Protocol," Internet draft, IETF, July 2001, draft-ietf-pppext-eapsrp-03.txt; for the latest version see http://www.ietf.org

[12] R. Needham and M. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Commun. ACM*, vol. 21, 1978, pp. 993–99.

[13] R. Bird *et al.*, "Systematic Design of a Family of Attack-resistant Authentication Protocols," *IEEE JSAC*, Special Issue on Secure Communications, vol. 11, no. 5, 1993, pp. 679–93.

[14] P. Cheng *et al.*, "A Security Architecture for the Internet Protocol," *IBM Sys. J., Special Issue on the Internet*, vol. 37, no. 1, 1998, pp. 42–60.

[15] H. Krawczyk, "SKEME: A Versatile Secure Key Exchange Mechanism for the Internet," *Proc. 1996 Internet Soc. Symp. Net. and Distrib. Sys. Sec.*, Feb. 1996, pp. 114–27.

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

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

[18] O. Goldreich, S. Goldwasser, and Silvio Micali, "How to Construct Random Functions," *J. ACM*, vol. 33, no. 169, 1986, pp. 210–17.

[19] U. Blumenthal *et al.*, "A Scheme for Authentication and Dynamic Key Exchange in Wireless Networks," *Bell Labs Tech. J.*, vol. 7, no. 2, 2002, pp. 37–48.

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

[21] C. Perkins, Ed., "IP Mobility Support for IPv4," IETF RFC 3344, Aug. 2002.

Biographies

Luca Salgarelli (salga@bell-labs.com) received his Laurea (Dr.Eng. degree) in electronic engineering from Milan Polytechnic University in 1995, and his M.Phil. in computer science from CEFRIEL, Milan, in the same year. He was a researcher in the Networking Department of CEFRIEL from 1995, where he worked in the areas of broadband IP networks and QoS provisioning. Since 1998 he has been with Lucent Technologies, first in the Wireless R&D Department in the United Kingdom, and then in the Networking Laboratory of Bell Laboratories Research, Holmdel, New Jersey. His activities cover design, development, and evaluation of systems and protocols for data networks, in particular when they involve mobility, QoS provisioning, and security. He is currently on a leave of absence from Bell Laboratories, teaching both graduate and undergraduate courses at the University of Brescia, Italy.

Milind M. Buddhikot (mbuddhikot@bell-labs.com) is a member of technical staff in the Center for Networking Research at Bell Laboratories, Lucent Technologies (Bell Labs-Lucent), Holmdel, New Jersey. He holds a D.Sc. in computer science (July 1998) from Washington University, St. Louis, Missouri, and an M.Tech. in communication engineering (December 1988) from the Indian Institute of Technology (IIT), Bombay. His current research interests are in the areas of systems and protocols for public wireless networks, authentication and dynamic key exchange, and multimedia messaging and caching. He has authored over 20 research papers and eight patent submissions on the design of multimedia systems and protocols, layer 4 packet classification, MPLS path routing, and authentication and dynamic key exchange. He served as a co-guest editor of *IEEE Network*'s March 2001 Special Issue on Fast IP Packet Forwarding and Classification for Next-Generation Internet Services. He has served in the capacity of tutorial chair for IEEE LCN '94, and '95, as a publicity chair for NOSSDAV '97, and as a program committee member for MMCN 2001, 2003, IEEE ICNP 2002 and 2003, and IEEE LCN 1993–2000 conferences.

Juan A. Garay (garay@bell-labs.com) received his Ph.D. in computer science from Pennsylvania State University in 1989. He also holds a degree of electrical engineer from the Universidad Nacional de Rosario, Argentina, and an M.E.E. from the Philips International Institute for Technological Studies/Netherlands' Universities Foundation, Eindhoven, The Netherlands. He has been a member of the Computing Sciences Center of Bell Labs since 1998. From 1990 until 1998 he was a research staff member at IBM T. J. Watson Research Center. In 1992 he was a postdoctoral fellow at the Weizmann Institute of Science, Rehovot, Israel, and in 1996 a visiting scientist at the Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands. His current research interests include theoretical and practical aspects of cryptographic protocols. He has been involved in the design, analysis, and implementation of a variety of secure systems over open networks, including cryptographic key management for secure communication, electronic payments, and distributed storage. At the foundational level, his achievements include showing that efficient (i.e., polynomial-time) optimal Byzantine agreement is possible, a well-studied problem that had been open since its formulation in 1980 (STOC '93); the notion of batch verification, with applications to modular exponentiation and digital signatures (Eurocrypt '98); and the construction of the first efficient zero-knowledge protocols satisfying the strong notions of nonmalleability and universal composability (Eurocrypt '03). He has published extensively in the areas of cryptography, distributed computing, algorithms, and fault tolerance, and served on the program committees of many conferences and international panels.

Sarvar Patel (sarvar@bell-labs.com) is a member of technical staff in the Wireless Security Group at Lucent Technologies, Whippany, New Jersey. His current research involves the development of cryptographic algorithms and protocols. Prior to joining Lucent, he was a research scientist in the Math and Cryptography research group and the Speech and Signal Processing group at Bellcore. He has B.S. and M.S. degrees in electrical engineering from Columbia University, New York.

Scott C. Miller (scm@bell-labs.com) is director of the High Speed Mobile Data Research Department at Bell Labs, Holmdel, New Jersey. He has B.S. and M.S. degrees in electrical engineering from Cooper Union, New York. His current research involves the integration of 802.11 and 3G wireless data service and related mobile networking issues concerning seamless mobility, authentication, security, roaming, and accounting. Prior to his work on 802.11/3G integration, he led several systems research efforts in wireless applications, implementing novel systems for wireless messaging, speech-driven directory services, wireless instant messaging, carrier-based content billing, and multimedia content adaptation.

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.