Enhanced Wireless Roaming Security using Three-Party Authentication and Tunnels*

Damien Leroy[1], Mark Manulis[2], and Olivier Bonaventure[1]

[1]Universite catholique de Louvain, 1348 Louvain-la-Neuve, Belgium
[2]TU Darmstadt & CASED, Germany

{damien.leroy,olivier.bonaventure}@uclouvain.be, mark@manulis.eu

ABSTRACT

Many organizations and many home users have deployed WiFi networks permitting external users to connect to the Internet through their networks. Such WiFi sharing poses many security risks for the visited network as well as for the visiting user.

In this paper, we focus on the recently introduced concept for tunneled WiFi roaming in which the infrastructure of the visited network is considered as part of the security architecture. A secure layer-2 tunneling between the user's device and his home network is performed by the visited network only after the successful authentication of all three parties. The authentication protocol provides the mobile device and its home network with a secret key that protects their end-to-end communication. Additionally, it provides another tunnel key, shared with the visited network, that protects the actual traffic exchanged between the visited and home networks and prevents diverse resource consumption attacks against the latter. This concept encourages users to provide roaming service in a more secure and privacy-friendly way. We show how to implement this concept using the IEEE802.11i/EAP framework, based on existing infrastructures and standard tunneling protocols.

Categories and Subject Descriptors

C 2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C 2.0 [**Computer-Communication Networks**]: General—*Security and Protection*

General Terms

Security, Design

Keywords

WiFi roaming, Protocol, Tunnel, EAP, Mobility

*This work is supported by the Belgian Walloon Region under its RW-WIST Programme, ALAWN Project.

U-NET'09, December 1, 2009, Rome, Italy.

1. INTRODUCTION

Almost all laptops, netbooks and many smartphones sold today are equipped with WiFi interfaces. The different WiFi variants have enabled new types of Internet access. Ten years ago, mobile employees had to use dial-up modems to contact their enterprise's VPN or access the Internet. During the last years, usage of WiFi has increased quickly in enterprise networks, but also in private houses, restaurants or hotels. In enterprise networks, the security of the WiFi network is today a strong concern for system administrators and most enterprise networks have implemented strong authentication schemes to authenticate their wireless users. At home, users often use a WEP or WPA shared key to connect to their router or set-top box. However, more and more WiFi users expect to be able to access the Internet from any location: at their favorite restaurant, at friends' house, in their hotel room, while shopping, during any conference, etc.

Several solutions have been implemented and deployed to serve these mobile users. Some users consider that Internet access should be free and so open their Internet access via WiFi. Others, grouped in communities such as FON [1] or *wifi.com*, share their connections with other members of their communities only. Some businesses, such as restaurants, consider the WiFi-based Internet access as an added value and propose a free access or install commercial hotspots to allow their visitors access the Internet. Finally, system administrators who manage enterprise and university networks are often pushed by their users to deploy an open WiFi-based Internet access for their visitors.

However, these ways of sharing an Internet access raise many security issues for the visited network as well as for the mobile guest. When visiting another company or institution for a very short stay, users often obtain temporary credentials to access the Internet. For the visited institution, dealing with the security risks brought on is not an easy task. Other users do not support the secure authentication mechanism (e.g., EAP-TTLS [2]) used by the company and must be placed in a less secure subnet isolated from the rest of the institution network. Despite these precautions, a visitor will use IP addresses belonging to the visited network (directly or indirectly through a NAT) to access the Internet. This means that any network on the Internet will consider this visitor as an actual employee of that institution. Thus, if the visitor misbehaves, the reputation of the institution may be tarnished. Consider for example an insecure laptop that is part of a botnet, infected by a worm or sending spam. A remote network which is attacked by such a laptop may

decide to blacklist the IP address or the entire prefix of the visited network. Such blacklistings are difficult to revert.

On the visitor side, there are also security risks. First of all, since the visited network forwards all the user traffic to and from the Internet, it is able to easily sniff the whole traffic and perform all kind of man-in-the-middle attacks. Rogue access points have already been used in many places and security specialists often avoid using public WiFi networks due to the associated risks. Unfortunately, most users do not have security skills and are often unable to determine whether the access point on which it is connecting does really belong to the visited institution.

In this paper, following the ideas from Manulis et al. [3], we describe an infrastructure and a suitable authentication and key establishment protocol that permits users to use an authenticated and possibly encrypted tunnel through their home network to access the Internet. The visited network has control over the secure tunnel between the mobile guest and his home network which is established upon successful authentication of all the parties. In contrast to existing tunneling solutions, e.g., VPNs based on IKEv2 and IPsec [4], authentication is now performed between the three parties in order to fit their trust models and security goals. The targeted type of roaming encompasses visits of short or medium duration in which it is undesirable for the visited network to issue any form of credentials to the mobile guests, i.e., enroll any registration procedure. Additionally, our scenarios do not assume high mobility of guests across different networks. Therefore, we are not concerned with the handover issues between different foreign networks. We remark that [3] provides theoretical evaluation and security analysis of the protocol. However, practical aspects such as deployment and implementation challenges based on known standards have not been investigated so far. The goal of this paper is to close this gap between theory and practice.

This paper is organized as follows. Section 2 describes our proposed infrastructure and protocols. Then, we present the architecture of our prototype implementation on the Linux platform in Section 3. Section 4 compares our solution with related work and we conclude in Section 5.

2. INFRASTRUCTURE DESCRIPTION

Our solution for wireless roaming is designed with two principles in mind. First, for legal reasons (see [5]) it is important to be able to identify the mobile guest in case it misbehaves, e.g., up- or downloads illicit content. In contrast to currently used authentication schemes that involve only the mobile user and the visited network, our solution relies on a three-party authentication protocol that involves the mobile user, the visited network and the home network of the mobile user. This home network will either be an enterprise or an ISP network while the mobile user will either be an employee of the enterprise or a subscriber of the ISP and thus identifiable. The second principle is to eliminate the need for the visited network in assigning an IP address to the mobile device and providing it with the unlimited Internet access. This is motivated by the fact that visited networks may suffer from blacklisting or other responsibility problems in case their mobile guests misbehave.

2.1 Architecture

The architecture of our wireless roaming solution is comprised of three main entities: the mobile user $\mathcal{M}$, his home network $\mathcal{H}$, and the foreign network $\mathcal{F}$ that is visited by $\mathcal{M}$.

First, we clarify the trust assumptions amongst these entities. There is a fully trusted relationship between $\mathcal{M}$ and $\mathcal{H}$; in particular they share a common high-entropy secret as a product of the registration phase. The trust level between $\mathcal{H}$ and $\mathcal{F}$ can vary depending on the roaming agreement established by both networks, either directly or through some third trusted party. In both cases their authentication is based on public key certificates. Since $\mathcal{M}$ and $\mathcal{F}$ may not be aware of each other prior to the roaming session they have to rely on the home network $\mathcal{H}$ that is supposed to authorize their connection. The connection establishment process should further prevent $\mathcal{F}$ impersonating either $\mathcal{M}$ towards $\mathcal{H}$ or vice versa since $\mathcal{F}$ mediates their communication. Besides that, the protocol should be flexible enough to support meaningful combinations of authentication and encryption amongst the participants during the roaming phase.

Our approach is composed of two subsequently executed phases. The first phase is given by a three-party authentication and key exchange protocol that involves $\mathcal{M}$, $\mathcal{H}$ and $\mathcal{F}$. We call this protocol the Roaming Authentication and Key Exchange (RAKE) protocol. The second phase is given by the layer-2 tunneling technique that is used to forward the packets between the foreign network and the home network throughout the roaming session.

The RAKE protocol allows $\mathcal{M}$ and $\mathcal{H}$ to agree on a secure end-to-end key K_{MH} and all three-parties to compute the additional tunnel key K_T. They also negotiate the parameters for authentication, encryption and tunnel establishment. K_T is used to protect the tunnel and for further parameter negotiation such as billing. It is also used to derive the WPA Master Session Key (MSK) if the user connects using a WiFi connection. K_{MH} is used to prevent $\mathcal{F}$ eavesdropping or traffic alteration, for instance encryption can be optionally enabled between $\mathcal{M}$ and $\mathcal{H}$.

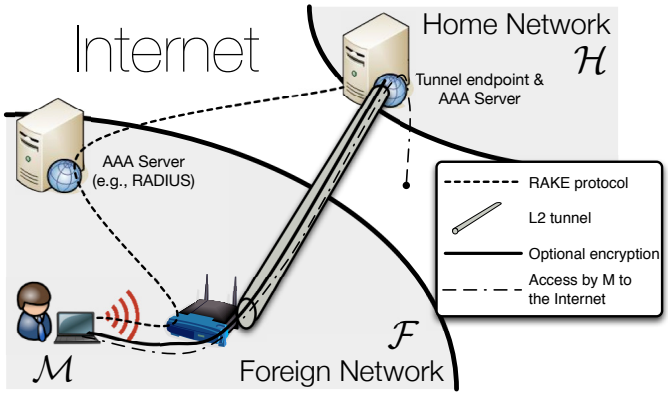

Figure 1: Simple architecture using our solution

The RAKE protocol is executed between $\mathcal{M}$, an AAA server[1] in $\mathcal{F}$ and in $\mathcal{H}$. Figure 1 shows the exchanged pro-

[1]"AAA server" is a term that defines an Authentication, Authorization and Accounting server. The server could only be used for a subset of these roles or could correspond to a more complex distributed infrastructure.

tocol messages. To simplify the presentation we consider $\mathcal{H}$ being the tunnel's endpoint and the AAA server; however, the AAA server may also run on a different machine. Once the RAKE protocol has completed, any data transmitted by $\mathcal{M}$ to the access point is encapsulated into an authenticated tunnel towards $\mathcal{H}$. At the tunnel endpoint, data is decapsulated and sent as if it was sent from a local host either to the Internet or to the local network as an Intranet request. The response packets are sent back to $\mathcal{M}$ using the same encapsulation approach. If necessary, roaming data can also be encrypted between $\mathcal{M}$ and $\mathcal{H}$ using the end-to-end key K_{MH}.

The authentication between the mobile and its home network is based on a symmetric shared secret of high-entropy. The choice of symmetric (rather than public-key) cryptography in this context allows to reduce the computation costs of the mobile device. The authentication between $\mathcal{H}$ and $\mathcal{F}$ is performed using public key certificates. For this, RAKE supports three different modes for the distribution of public keys. First, RAKE can use certificates that have been already pre-shared between the partner networks. However, this solution cannot be applied at a wide scale. Second, RAKE can obtain certificates from a centralized service that owns all the certificates and the corresponding revocation list (CRL), i.e., each time a network needs a certificate of another network it sends the corresponding request to the service. Third, RAKE allows networks to distribute their certificates during the protocol execution and provides the recipients with the ability to check the validity of the received certificates by comparing to CRLs obtained from the central service. The actual mode used in practice would depend on the trust and business models in which the protocol is deployed.

2.2 Authentication and key establishment

To allow the protocol to be used without large modifications for both the mobile device and the access point, it has been decided to include it into the WPA2/IEEE 802.11i infrastructure and more precisely into WPA with EAP, also called "WPA2-Enterprise". EAP is the Extensible Authentication Protocol [6] and so can be easily extended with a new EAP method. The advantage of this choice is that WPA with EAP is already supported by most WiFi infrastructures and clients. More details on this implementation will be provided in Section 3.

The EAP-RAKE protocol is depicted in Figure 2. The notations used to describe the protocol, in particular its messages, are summarized in Table 1. The standard EAP control messages are not shown. They consist of an EAP-start message which is sent from $\mathcal{M}$ to the access point (AP) followed by EAP identity request and response messages prior to execution of the protocol and the final EAP-success message sent from AAA server to $\mathcal{M}$ via the AP at the end.

I1 message.

This message is sent by $\mathcal{F}$ and contains its id and a chosen nonce. It can also contain an optional signature and certificate request to the mobile device if the foreign network wishes to authenticate the guest based on a certificate issued by the home network. In this case the message also includes the description of signature algorithms SP_F^{SIGMF} supported by $\mathcal{F}$ and the a $\mathcal{H}$'s certificate request if it does not possess it yet. Requesting $\mathcal{M}$'s signature only prevents

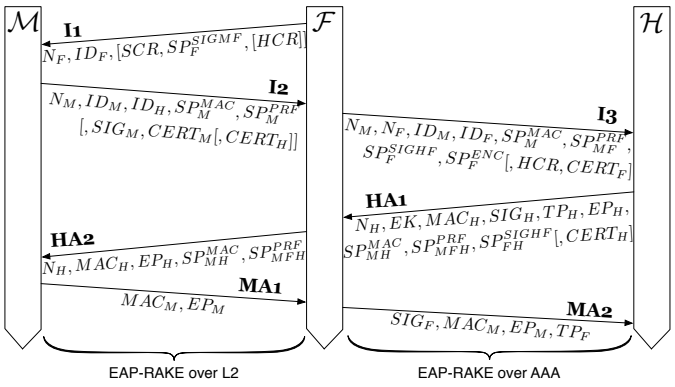

Figure 2: The RAKE protocol. Symbols used are described in Table 1. Payloads between brackets are optional.

Symbol	Description
N_X	Nonce chosen by X
ID_X	X's identity
SIG_X	Signature, computed by X, on all previous payload data
$CERT_X$	X's certificate
EK	Encrypted temporal key, it is computed by encryption, with public key of $\mathcal{F}$, of k_t
MAC_X	MAC computation by X
SP_X^Y	Security algorithm proposal, algorithm(s) for Y function supported by X
SCR	Signature and Certificate Request
HCR	Home Certificate Request
TP_X	Tunnel Parameters, generated by X
EP_X	Encryption Parameters, generated by X

Table 1: Payload type used in RAKE

$\mathcal{M}$ from trying to initiate too much new connection at a same time and should only be used if such abuses are observed.

I2 message.

$\mathcal{M}$ answers with I2 message containing its chosen nonce, two ids (one of $\mathcal{M}$ and another one of $\mathcal{H}$), and the description for the Message Authentication Code (MAC) and the Pseudo-Random Function (PRF) supported by $\mathcal{M}$. If requested in I1, $\mathcal{M}$ can also sign the previous payload with the algorithm chosen among $\mathcal{F}$'s proposals.

I3 message.

This message is sent by $\mathcal{F}$ to $\mathcal{H}$. It contains ids and nonces chosen by $\mathcal{M}$ and $\mathcal{F}$, the previous SP for MAC and the SP_{MF}^{PRF} updated as an intersection of the algorithms in SP_M^{PRF} and the algorithms supported by $\mathcal{F}$. I3 can optionally contain a certificate request for $\mathcal{H}$ and the certificate of $\mathcal{F}$.

Once it has been received, $\mathcal{H}$ can compute a session id SID as a concatenation of nonces N_M, N_F, N_H and ids ID_M, ID_F, ID_H. This concatenation ensures that SID is unique for each new session. A temporary key k_t is then generated by $\mathcal{H}$ using PRF with the pre-shared secret between $\mathcal{M}$ and $\mathcal{H}$ and SID as input. This ensures that k_t can only be computed by $\mathcal{H}$ and later by $\mathcal{M}$. The actual tunnel key K_T is then derived similarly using k_t. Finally, the end-

to-end key K_{MH} is derived from SID using PRF with the pre-shared secret between $\mathcal{M}$ and $\mathcal{H}$. In all key derivation steps SID is augmented with different public labels in order to ensure independence amongst the end-to-end and tunnel keys.

HA1 message.

This message contains the nonce of $\mathcal{H}$, the encryption EK of k_t under $\mathcal{F}$'s public key, the MAC value MAC_H serving as an authenticator of $\mathcal{H}$ towards $\mathcal{M}$ which is computed using the pre-shared secret of $\mathcal{M}$ and $\mathcal{H}$, and the signature of $\mathcal{H}$ on the previous messages serving as an authenticator of $\mathcal{H}$ towards $\mathcal{F}$. HA1 also includes several SP indicating the final choice of MAC and PRF algorithms supported by all the parties. Additionally, HA1 contains parameters TP_H needed to negotiate the tunnel establishment with $\mathcal{F}$, and EP_H indicating supported encryption mechanisms in case end-to-end encryption between $\mathcal{M}$ and $\mathcal{H}$ should be used.

HA2 message.

This message consists of payloads received by $\mathcal{F}$ within the HA1 message and forwarded to $\mathcal{M}$. Upon receiving the HA2 message $\mathcal{M}$ is able to authenticate $\mathcal{H}$.

MA1 message.

This message contains MAC_M which is computed by $\mathcal{M}$ on SID using the pre-shared key with $\mathcal{H}$ and is used to authenticate $\mathcal{M}$ towards $\mathcal{H}$. Additionally, MA1 contains encryption parameters EP_M as response to EP_H.

MA2 message.

This message consists of MAC_M, the signature of $\mathcal{F}$ on SID and MAC_M that authenticates $\mathcal{F}$ to $\mathcal{H}$, and encryption parameters EP_M as received from $\mathcal{M}$. Additionally, MA2 contains tunnel parameters TP_F specified by $\mathcal{F}$ in response to TP_H.

2.3 Encapsulation mechanism

As mentioned earlier, our WiFi roaming solution uses a tunnel between $\mathcal{F}$ and $\mathcal{H}$. This tunnel is an answer to $\mathcal{F}$'s and $\mathcal{H}$'s security issues. First, thanks to the utilization of the tunnel, $\mathcal{F}$ does not need to allocate an IP address to $\mathcal{M}$ nor grant it an unlimited Internet access. This implies that $\mathcal{M}$ cannot misuse an IP address owned by $\mathcal{F}$ to misbehave neither inside $\mathcal{F}$'s network nor on the global Internet. This is particularly important for enterprise networks that need to protect their infrastructure. Second, $\mathcal{H}$ can enforce its own policies on $\mathcal{M}$ as all packets sent by $\mathcal{M}$ are tunneled to $\mathcal{H}$. Since our tunnel is a layer-2 tunnel, this implies that some home networks can use IPv4 even if others have already migrated to IPv6. Some home networks may protect their mobile devices by using firewalls and intrusion detection systems while others can grant unrestricted Internet access. Furthermore, the optional encryption of packets transmitted through the tunnel allows $\mathcal{H}$ to protect against malicious $\mathcal{F}$ and so enforce confidential communication with mobile users.

The tunneling protocol implemented between $\mathcal{F}$ and $\mathcal{H}$ is L2TP [7]. L2TP is a protocol that allows to encapsulate a layer-2, i.e. MAC-layer, frame into a L2TP over UDP packet. In order to be able to encapsulate all the layer-2 frames sent by $\mathcal{M}$, this encapsulation is performed directly from the access point on which the client is connected to the tunnel endpoint in his home network.

Our implementation allows for the optional encryption of data exchanged between $\mathcal{M}$ and $\mathcal{H}$. While enterprise networks will probably require it, some ISP networks may prefer to support a larger number of users without encryption rather than fewer with encryption. Our prototype currently uses IP Encapsulating Security Payload [8] (ESP) from IPSec. ESP is used in tunnel mode which means that it encapsulates the whole IP packet in an encrypted ESP payload over IP.

3. IMPLEMENTATION

We have implemented the proposed WiFi roaming solution by extending the EAP supplicant and authenticator that are part of the *hostap* distribution [2]. Due to space limitations, we only describe our implementation of the RAKE protocol. For L2TP and IPSec, we reuse respectively the *xl2tpd* and *Openswan* implementation, both maintained by Xelerance Corporation.

One of the main advantages of EAP is that it is easily extendable. In practice, this means that new authentication scheme, denoted in general as an EAP method, can be added easily as a new module to an existing EAP implementation. Some EAP methods performing only simple authentication (e.g., EAP-MD5) or channel protection (e.g., EAP-TTLS) can be combined to securely authenticate a mobile node at an access point (or a delegate AAA server) and to eventually derive the WPA master key used for encrypting the wireless channel. EAP-RAKE is a standalone EAP method, it performs both authentication and key generation. It also provides other features that cannot be performed by other existing EAP methods, namely authentication and key exchange with a third party. The RAKE packets exchanged between $\mathcal{F}$ and $\mathcal{H}$ are sent using EAP over RADIUS.

In practice, in *hostap*, the EAP process interacts with the upper method through a simple API. It calls the EAP method hooks upon creation of the state at method startup, each time an EAP message is received or has to be sent, to know whether the method has succeeded and if so, to obtain the WPA master session key. The implementation of this new EAP method required, in addition to the support for the new configuration options, specific modules containing about 2700 lines of C. It runs on both Linux desktops and *OpenWRT*, a Linux distribution embedded in several WiFi access points. Cryptographic functions proposed for the RAKE protocol in [3] are also supported in our implementation. The source code has been written in a modular way so that new cryptographic algorithms can be added easily.

To evaluate the performance of our RAKE implementation, we have built a small testbed composed of one Pentium 4 2.6Ghz with 1 GBytes RAM running Linux acting as the mobile device, an ASUS 500 GP access point running OpenWRT 8.09.1 (Broadcom CPU BCM47XX 266 Mhz, 32 MBytes RAM) and a Linux server with a dual CPU E2180 2.00GHz and 2 GBytes of RAM.

Within this testbed we have measured the processing time required for the different messages of the RAKE protocol:

[2]Hostap is composed of *hostapd* (the authenticator, i.e, the access point), *WPA supplicant* (i.e., $\mathcal{M}$) and some utilities. The main contributor is Jouni Malinen. Project website: `http://hostap.epitest.fi/`

the low-end access point took 18 msec to perform the initialization and to generate I1. The mobile device replied directly with I2. The processing of the I2 message and the generation of the I3 by the access point took slightly less than 6 msec and the home network took 8 msec to initiate and reply with the HA1 message. The longest processing times were the processing of the HA1 message (46 msec) and the processing of the MA1 message (57 msec) by the access point.

In comparison when using PEAP+MSCHAPv2, the processing on the supplicant requires 14 msec while the access point takes more than 176 msec to process the authentication messages.

4. RELATED WORK

In an enterprise, a typical way to permit short-term visitors to connect to its network is to give them temporary credentials. In addition to the administrative cost of managing the creation of these accounts, the identity of the visitors should be checked and they should be informed about the acceptable usage practices. However, a user coming from another company has already credentials in his home network, so why do not use these? The *Eduroam* project [9] follows this idea and allows any user, from a partner educational institution, to connect in any other institution by using its "home" credentials. *Eduroam* uses IEEE802.1X with methods such as EAP-TTLS and a hierarchy of RADIUS servers to authenticate the user in his home institution. Other propositions suggested to provide authentication in roaming situations have already been discussed in [3]. A common issue of these solutions is that they are mainly authentication schemes and so do not allow key derivation for a tunnel-based approach. All these do not suggest solutions that would prevent the misuse of the IP address belonging to the visited institution by the mobile user.

For a user wishing to share its WiFi connection at home, some other solutions such as *wifi.com* or FON [1] exist. *wifi.com* acts more as a social network in which people share their home WEP/WPA personal secret key with their friends using the provided application. FON allows users from their community to connect with their own FON account credentials to a specific hardware placed by another user. With the latter, you can have people you absolutely do not know using your network to access the Internet. For both *wifi.com* and FON, if the visitor misbehaves, the owner of the access point will first be considered responsible. Moreover, unlike in enterprise networks, home routers use NAT to connect several users that will use the same public IP address. As a consequence, if several people are connected at the same time, it will be difficult, depending on whether the NAT maintains precise logs or not, to determine a posteriori who made a reprehensible action.

Anyway, connecting to a foreign WiFi network, whether owned by an enterprise or by a private user, is often a risky operation from the security point of view. Technically, the visited network is able to sniff and alter all unencrypted data sent or received by the user. Additionally, in most of the previously described solutions, the user is even not able to authenticate the access point. A classical solution used by enterprises is to force the employees to use a Virtual Private Network (VPN) to connect to the Internet outside the enterprise. VPN is an encrypted and eventually authenticated tunnel from the user's laptop to his home institution VPN server. Most VPNs use protocols such PPTP [10] and L2TP [7] over IPSec. They rely on specific authentication schemes. These authentications are only performed between the mobile user and his institution, disregarding the visited network. In other words, if used alone, VPN does not provide information to the visited network about who is connecting to the network. Another existing technology, worth to be mentioned, is Mobile IP [11]. Using encapsulation it allows a user to use an IP address from his home network while outside. However, Mobile IP provides neither authentication with the visited network nor tunnel encryption. It also forces the visited network to offer a public IP address to each visitor.

All the aforementioned solutions only address one concern at a time but do not enable to build a global solution tackling all different security threats. The only currently available tunnel-based solution in which the visited institution is aware that the mobile user is doing roaming to its home network has been proposed by Sastry et al. [12]. In their scheme the visited network accepts every device without any authentication and grants it access to its home network over the Internet. The mobile device can thus initiate a VPN connection (using NAT traversal techniques if necessary) to its home network. However, this solution has several weaknesses. First, the Internet access granted by the visited network, even a restricted one, may bear intrusion risks to its infrastructure. Second, the mobile device must comply with the network layer infrastructure of the foreign network, e.g., IPv4/IPv6, IP assignment via DHCP, etc. Third, VPN tunnels do not provide any proof to the visited network that the mobile device is connecting to its genuine home network as a VPN connection can be established to any server on the Internet. Fourth, visited and home networks do not authenticate each other and, as a consequence, neither accounting mechanisms nor quality-of-service contracts can be securely implemented. We fairly remark that Sastry et al. were focusing on the actual architecture for a city-wide WiFi roaming rather than dealing with the related authentication and key establishment goals.

5. CONCLUSION

More and more users expect to be able to use their WiFi device to access the Internet everywhere. Besides commercial hotspots, many universities, enterprises and home WiFi networks have implemented techniques to share their WiFi networks with visitors. However, sharing a WiFi network induces security risks if the visitor misbehaves.

To counter these risks, we have proposed a safer WiFi roaming technique that combines three-party authentication with tunnels. We have described the encapsulation mechanisms, the RAKE protocol and its implementation on top of EAP. Our prototype, running on the Linux platform, shows that it is possible to achieve three-party authentication without causing higher computation delays than existing EAP schemes. Our further work is to deploy RAKE on a wider scale to gain operational experience with it.

6. REFERENCES

[1] FON, http://www.fon.com.

[2] P. Funk and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)," IETF, RFC 5281, Aug. 2008.

[3] M. Manulis, D. Leroy, F. Koeune, O. Bonaventure, and J.-J. Quisquater, "Authenticated Wireless Roaming via Tunnels: Making Mobile Guests Feel at Home," in *Proc. ACM Symp. on Information, Computer and Communication Security (ASIACCS)*, Mar. 2009, pp. 92–103.

[4] S. Kent and K. Seo, "Security Architecture for the Internet Protocol," IETF, RFC 4301, Dec. 2005.

[5] R. Robert, M. Manulis, F. de Villenfagne, D. Leroy, J. Jost, F. Koeune, C. Ker, J.-M. Dinant, Y. Poullet, O. Bonaventure, and J.-J. Quisquater, "WiFi Roaming: Legal Implications and Security Constraints," *Int. J. of Law and Information Technology*, vol. 16, no. 3, pp. 205–241, 2008.

[6] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible Authentication Protocol (EAP)," IETF, RFC 3748, Jun. 2004.

[7] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter, "Layer Two Tunneling Protocol L2TP," IETF, RFC 2661, Aug. 1999.

[8] S. Kent, "IP Encapsulating Security Payload (ESP)," IETF, RFC 4303, Dec. 2005.

[9] Eduroam, http://www.eduroam.org.

[10] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn, "Point-to-Point Tunneling Protocol (PPTP)," IETF, RFC 2637, Jul. 1999.

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

[12] N. Sastry, K. Sollins, and J. Crowcroft, "Architecting Citywide Ubiquitous Wi-Fi Access," in *HotNets-VI*, 2007.