A secure and performant token-based authentication for infrastructure and mesh 802.1X networks

Romano Fantacci, Leonardo Maccari, Tommaso Pecorella
Department of Electronics and Telecommunications
University of Florence
tel. : +390554796467 - fax : +390554796485 Florence, Italy
Email: {fantacci, maccari, pecorella}@lart.det.unifi.it

Federico Frosali
Telecom Italia Lab
Via G. Reiss Romoli, 274 - 10148 - Turin, Italy
tel: +39011 2285111 - fax: +39011 2285520
Email: federico.frosali@tilab.com

***Abstract*— This work deals with the design of secure handoff protocols for wireless networks using the security model introduced by IEEE 802.1X standard. The key exchange model introduced in the standard might be implemented in multiple ways each one carrying advantages and disadvantages in terms of security and performance when applied to reauthentication protocol. After the analysis of different model of reauthentication we introduce a novel scheme based on token exchange to speed up the handoff phase. This protocol variant was designed and implemented as a prototype in a joint project between University of Florence and Telecom Italia Laboratories, and proved better performances than standard protocols while maintaining a high security level.**

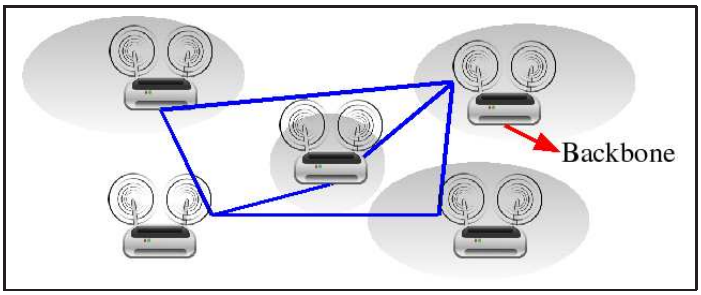

Fig. 1. A meshAP network, each meshAP serves a different WLAN

I. Introduction

Wireless local area networks reached a large commercial success in the latest years for the ease of use and deployment and for the wider possibility offered to users and administrators. Standards like *IEEE 802.11*, in all its recent versions such as *b*,*g* had a great impact on the market and produced a great change in the user perception of networking, users are getting familiar with being constantly connected and with deviating on a single medium different kind of traffic, from telephony to data traffic. One of the crucial point of the present and future success of wireless networking is mobility, that is possibly the most evident advantage to end users, but that introduces new problems to face in different aspects, such as coverage of the path followed by the user and security of the access.

To solve the problem of coverage, a solution that seems to receive great interest by service providers and telecommunication company is the use of self organizing *mesh networks*. A mesh network is an ad-hoc network made of *peers*, without hierarchy, where each node of the network participates to routing of packets; there is no fixed topology and network extension passes through the addition of a new terminal to the network border, without internal reorganization. A particular kind of mesh network is a *meshAP* network, where *IEEE 802.11 Access Points* (AP) are connected in mesh mode and each one offers networking access to its subnet, as in fig. 2. A meshAP network grants scalability and ease of setup even in extremely difficult situations, such as networks deployed in emergency or hostile environment (tactical networks) even with mobile APs. Standards like *IEEE 802.11* or *IEEE 802.16e* support a mesh mode of operation.

The other main problem of wireless networks, is due to their intrinsic vulnerability to security attacks. On one side there is a lack of a defined geographical border, that makes it difficult to distinguish legitimate traffic from external traffic, on the other side there is the ease of access to the medium for the attackers. Main consequences on security are that an attacker can freely intercept traffic and inject information into the network. To prevent this, the only solution is the use of cryptography, that can separate legitimate traffic from external traffic. Cryptography depends on algorithms and keys, that must be negotiated in advance, so that access to the network must begin with an authentication phase in which the station involved must recognize each other; that phase is normally time consuming and computationally intense. While performing an handoff from an AP to the other the re-authentication phase must be shortened as much as possible, to limit the loss of connectivity.

This paper presents a solution for fast and secure handoff in wireless networks using the security model introduced by *IEEE 802.1X*, and reports success of implementations in a real testbed, estimating gains of performance in a meshAP environment. Moreover it introduces generic guidelines for the design of secure re-authentication schemes in *IEEE 802.1X* networks.

II. Security guidelines for reauthentication in 802.1X networks

IEEE 802.1X model introduces a high level security scheme for authentication and access control that has been successfully applied to *IEEE 802.11i* as well as to *IEEE 802.16e* networks. Its adoption follows the complete unsuccess of custom authentication techniques used in *IEEE 802.11* and *IEEE 802.16d*, as testified in [1], [2]. *IEEE 802.1X* defines three entities in the network, depicted in fig. 2:

- a Supplicant, *SA*, that is the client requesting access to the network.
- an Authenticator, *AA*, offering layer two connectivity to end clients. In *IEEE 802.11* networks, the AP.
- an Authentication Server, *AS*, a database containing the credentials needed to accept or deny access to clients.

The link between *AA* and *AS* depends on layer III connectivity and is provided by some security protocol like UDP-based RADIUS. Whenever *SA* enters the network, it phisically connects to *AA*, but *AA* only forwards authentication packages to *AS*. So initial authentication is carried on between *SA* and *AS* with *AA* acting only as a proxy. During this phase various authentication protocols can be used between *AS* and *SA*, authentication protocols are transported by EAP (Extensible Authentication Protocol) protocol as EAP *methods*. As an example *IEEE 802.11i* does not define a single method to be used but imposes that it must produce a bidirectional authentication and that it must generate some *fresh* shared secret (called *PMK* key) into the endpoints, *AS* and *SA*, that will be used as a proof of successfull authentication. At the end of authentication *AS* communicates to the *AA* that *SA* is an authorized machines, and moves into *AA* the shared secret. A wide used EAP method, based on certificate exchanges, is EAP-TLS that requires a 8-way handshake to produce the *PMK*. Note that in a mesh network the path from *SA* to *AS* might be several hops long, in [3] the average delay for different routing techniques in measured to be about vary from 0.37 to 3 seconds, so that EAP authentication phase might take several seconds to be performed.

Now a second phase begins, in which *SA* and *AA* derive from the *PMK* key a second key (*PTK* key in *IEEE 802.11i*) that is a link key and will be used to protect real data communications between these two parties. In *IEEE 802.11i* a 4-way handshake is used, but being a link layer communication it does not impact performance as much as EAP phase.

The same authentication takes place between APs whenever a new link is activated. This can happen often If we consider the possibility of temporary AP failures or signal loss. To have performances compatible with real applications, re-authentication procedure must be shortened.

A. Design Guidelines

Whenever a station *SA* is performing an handoff between AA_1 and AA_2 (see fig. 2), it appears clear that phase I described before should not be repeated. Since *PMK* key is a proof of being accredited to enter the network, it should be moved to

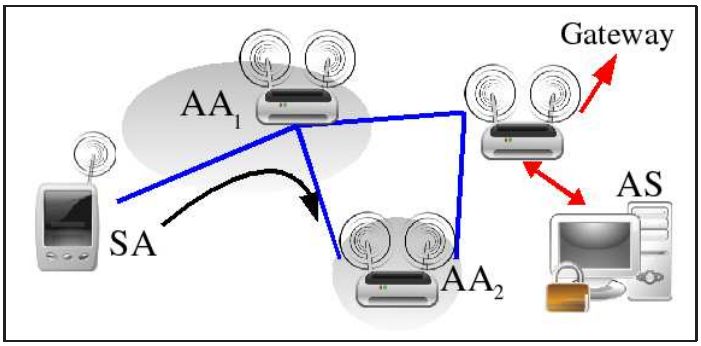

Fig. 2. *IEEE 802.1X* scheme, in blue wireless links, in red wired links

AA_2 (maybe refreshed) but not regenerated with *AS*. Only *AS* and AA_1 own *PMK* key apart from *SA*, so it might be moved between the twAPs's or received from *AS*. It must be clear that compromission of *PMK* keys means possible compromission of all future and even past traffic of the terminal using that key (this is possible if we imagine an off-line attack with multiple attackers collecting traffic that will be deciphered once obtained the corresponding key).

Moreover, while in infrastructure networks trust relationships are easy to understand, in a mesh network they are not so easy to define. This is due to several factors: first is that terminals are constantly exposed to external attacks, so that they might be under control of an attacker, then, due to the dynamical nature of these networks, in certain circumstances it might just not be possible to identify users, even authenticated users can be malicious entities, third, introducing mobility and signall loss for APs a fast way to rebuild trust between neighbors is required. Considering the possibility of the insider enemy makes the design of security protocols extremely challenging, we outlined the following guidelines to perform a secure, performant handoff even in these situations:

1) APs's should not exchange *PMK* between each other, otherwise *AS* would loose control over access control. Only *AS* should move *PMK* keys.
2) A compromised terminal, should not be able to generate valid credentials to let other unauthorized terminals access the network (if not directly attached to the compromised one).
3) A compromised machine should not be able to freely gather *PMK* keys that it doesn't need. This prevents an attacker from being able to decrypt all the traffic of the network, apart from the traffic it is directly involved into, even with an off-line attack.

Designing protocols resistant to insider attackers means realizing a system that should not be completely under the attacker control even after compromission of a single machine. These guidelines enforce a policy that tries to speed down the attack and limitate the consequences over the network, waiting for other components of the security mechanisms to identify the attack, and maybe react, in a multi-fence security project, as described in [4]. Besides, it is advisable that an insider attacker should not be able to produce stronger denial

of service attacks then an outsider attack, and, as a last issue *AS* should never lose control over authentications of the network.

III. Proposed Solution

Respecting the general guidelines outlined we designed and realized an handoff mechanism that is used to avoid the repetition of TLS phase of *IEEE 802.1X* networks. Our starting point is that AA_2 must ask and obtain the *PMK* key relative to *SA* from *AS* only and not from its neighbor APs. This means that whenever a handoff is performed, AA_2 is forced to realize a multi-hop communication with *AS* that we try to shorten as much as possible, maintaining a high level of security. Still we have to limit the possibility of AA_2 to request *PMK* keys to *AS*; if we let AA_2 ask and receive any *PMK* key from *AS* we break the third guideline we stated. Our mechanisms is based on the use of authentication tokens that are dynamically generated during an handoff by the requesting station. Whenever performing an handoff *SA* generates a token and sends it to AA_2, the token is forwarded to *AS* and *AS* decides its validity. Once verified the token *AS* moves the *PMK* key into AA_2, and phase II can take place.

The basic idea behind this protocol is that AA_2 must provide to *AS* the proof that it is in contact with a station that wants to perform an handoff to receive a *PMK*, that proof is the token, without a token AA_2 won't receive a *PMK*, preserving our second guideline. Since the token is designed to be impossible to be replicated or generated by any other station, even if compromised, AA_2 cannot gather keys from the *AS*, and it cannot create custom tokens to be distributed to other unauthorized machines. The computational power needed to generate and verify the token is limited (no public key algorithms used) so that there is no overload for the terminals and no new denial of service attacks are introduced. *Fast authentication* has been realized with a 2-way only exchange, that is, the EAP phase is reduced by 75% of the packets needed by EAP-TLS.

We realized this *fast authentication* protocol in a *IEEE 802.11i* network modifying hostap and Freeradius suite of application for GNU/Linux and we measured the results. In table I we report inter-arrival times of a packets constituting a reauthentication, packages marked as $SA \rightarrow AA$ are one link packets, while packed marked as $AA \rightarrow AS$ represent packages that traverse the whole network. Our implementation was done over a simple network where the access points where directly connected to the authentication server, so also the multi-hop path is actually a direct link with static routing. Based on the results of [3] we used the factor F to amplify the latency of the multi-hop packets to simulate a mesh environment. Total time of the handoff is reported but since it is dependant from multiple issues (firmware/driver implementation) and our solution impacts only EAP phase we focus attention only on EAP time and its percentage over the whole reauthentication process.

In table II we report a comparison of the reauthentication time necessary for *fast authentication* and EAP-TLS; measured in the same testbed. It can be noted that our solution produces a gain between 85-90% of the EAP time, that is the main part of a complete reauthentication, thus greatly reducing the time needed even in a mesh environment with multi-hop paths between *AA* and *AS*.

		IAT (s)	%EAP	IAT2 (s)	F
1	SA -> AA IEEE 802.11 Authentication				
2	AA -> SA IEEE 802.11 Authentication	0.0392		0.0392	1
3	SA -> AA IEEE 802.11 Association	0.0017		0.0017	1
4	AA -> SA IEEE 802.11 Association	0.0807		0.0807	1
5	SA -> AA EAPOL Start	0.0020		0.0020	1
6	AA -> SA EAP Request, Identity	0.4417		0.4417	1
7	SA -> AA EAP Response, Identity	0.0064	35.97	0.0064	1
8	AA -> AS RADIUS Access Request	0.0017	9.64	0.1727	100
9	AS -> AA RADIUS Access Accept	0.0063	35.42	0.0063	1
10	AA -> SA EAP Response	0.0034	18.97	0.3400	100
11	AA -> SA EAPOL Key	0.0009		0.0009	1
12	SA -> AA EAPOL Key	0.0250		0.0250	1
13	AA -> SA EAPOL Key	0.0089		0.0089	1
14	SA -> AA EAPOL Key	0.0273		0.0273	1
	Total time (s)	0.6454		1.1530	
	EAP exchange time (s)	0.0179		0.5255	
	100*(EAP time)/(total time)	2.7770		45.5813	

TABLE I

fast authentication INTER ARRIVAL TIME MEASUREMENTS, AND PERCENTAGE OVER WHOLE REAUTHENTICATION

	EAP TIME	EAP TIME (2)
EAP-TLS (s)	0.1777	3.6739
Fast Auth. (s)	0.0179	0.5255
Gain (s)	0.1598	3.1484
Gain (%)	89.91	85.7

TABLE II

EAP TIME: TOTAL TIME NEEDED FOR EAP EXCHANGE IN HANDOFF PROCEDURE, EAP TIME (2): EAP TIME AMPLIFIED BY FACTOR F

IV. Conclusions

Our solution is applicable to every *IEEE 802.1X* network, such as *IEEE 802.11i* or *IEEE 802.16e*, and offers better performances then standard protocols like EAP-TLS, being it high level (EAP) solution it might be used also in mixed networks, if both support *IEEE 802.1X*. For the low complexity required the basic structure could be replicated even in non-802.1X low power networks like sensor networks. Finally, the guidelines we defined are to be used as a reference for the the design of reauthentication protocols in any mesh-like situation.

References

[1] N. Shankar, W. A. Arbaugh, and Y. C. J. Wan, "Your 802.11 wireless network has no clothes," May 15 2001. [Online]. Available: http://citeseer.ist.psu.edu/472552.html;http://www.drizzle.com/~aboba/IEEE/wireless.pdf

[2] D. Johnston and J. Walker, "Overview of IEEE 802.16 security," *IEEE Security & Privacy*, vol. 2, no. 3, pp. 40–48, May/June 2004.

[3] R. S. Gray, D. Kotz, C. Newport, N. Dubrovsky, A. Fiske, J. Liu, C. Masone, S. McGrath, and Y. Yuan, "Outdoor experimental comparison of four ad hoc routing algorithms," in *Proceedings of the ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, October 2004.

[4] Y. Hao, L. Haiyun, Y. Fan, L. Songwu, and Z. Lixia, "Security in mobile ad hoc networks: Challenges and solutions," *Wireless Communications, IEEE*, vol. 11, pp. 38 – 47, 2004.