An Authentication Scheme for QoS-aware Multi-operator maintained Wireless Mesh Networks

Levente Buttyán and László Dóra
Laboratory of Cryptography and Systems Security (CrySyS)
Budapest University of Technology and Economics
Email: {buttyan, dora}@crysys.hu

Abstract

In this paper, we consider QoS aware mesh networks that are maintained by multiple operators and they cooperate in the provision of networking services to the mesh clients. In order to support mobile users and seamless handover between the access points, the authentication delay has to be reduced. Many proposed fast authentication schemes rely on trust models that are not appropriate in multi-operator environment. Here, we propose two certificate based authentication schemes such that the authentication is performed locally between the access point and the mesh client. We consider both powerful and constraint mesh clients and we propose certificate sets to decrease the authentication latency. We compare our proof-of-concept implementation to current widely used authentication methods like EAP-TLS, and we conclude that our proposed authentication scheme is considerably faster in all considered scenarios.

1. Introduction

EU-MESH networks. In the EU-MESH project (www.eu-mesh.eu), we study multi-operator maintained QoS-aware wireless mesh networks for high speed Internet access. In this paper, we refer to such networks shortly as the EU-MESH network.

The EU-MESH network consists of mesh routers that form a static wireless ad hoc network. Some of the mesh routers function as gateways to the wired Internet, and some of them function as wireless access points (AP) where mobile mesh clients can connect to the network. The sets of gateways and APs can overlap and they do not necessarily cover the entire set of mesh routers.

We envisioned that the mesh routers are potentially operated by multiple operators, and they cooperate in the provision of networking services to the mesh clients. This cooperation is based on business agreements (similar to roaming agreements in the case of cellular networks). Mesh clients (MC) are mobile computing devices (laptops, PDAs, etc.) operated by customers. Customers may be associated with one or more operators by contractual means and have the ability to roam to the rest of the cooperating operators, if necessary.

The mesh network supports QoS-based applications and mobility of the MCs. QoS services may have requirements on the length of the interruptions in the communication that they can tolerate. When a MC moves from one AP to another, it has to re-authenticate itself as part of the handover process. Thus, the re-authentication delay must be minimized in order to ensure that the interruption caused by the handover remains tolerable for the applications.

In this paper, we are focusing on the MC re-authentication process in EU-MESH networks. Furthermore, we consider the problem of setting up a connection key between the MC and the AP that is used for the continuous enforcement of some access control policy in the network.

In the rest of this section, we define the requirements on the authentication and connection key generation mechanism in such mesh networks. In Section 2, we give an overview of the related works. In Section 3, we propose two certificate based authentication methods. We evaluate the authentication delay of our proof-of-concept implementation by comparing it to some widely used solutions in different scenarios in Section 4. Finally, we conclude our paper in Section 5.

Requirements. The requirements on the authentication method and access control enforcement has been investigated in [1], already. The main requirement is that the authentication method has to support seamless handover. The whole authentication system should be scalable in terms of the number of APs and MCs. The participants have to authenticate each other

978-1-4244-4439-7/09/$25.00

mutually. Furthermore, the connection key should be controlled by each participant. The characteristic of multi-operator environment has special requirements: First of all, existing standards should remain useable. Therefore, no changing on any part of the current standard is permitted. It is also beneficial to avoid using single entity that all the operators have to trust.

2. State-of-the-art

In the literature, many authentication and access control enforcement methods have been proposed. We investigate them through a taxonomy presented in [1], where a more detailed description of the state-of-the-art can be found, too. We categorized the proposed solutions by the place of the access control enforcement and by the place of the authentication.

The access control can be enforced by 1) a central entity, 2) the gateways or 3) the APs. In the first two cases, the whole mesh network is unprotected from unauthorized access. Therefore, an attacker is able to decrease the QoS level provided by the network by flooding the mesh network with rogue traffic. Thus, in what follows, we consider only those cases when the APs are responsible for the access control enforcement.

When the access control is enforced by the APs, the authentication can be performed 1) at a remote, central authentication server, 2) at local entities (e.g. mesh routers) playing the role of authentication server, or 3) at APs.

The main benefit of the central authentication server is the easy administration of the subscribers, however it is a single point of failure. In some proposals, the central authentication is only a fall back solution for the case when the responsible mesh router has no data for the MC authentication (typically for the first authentication). The main benefit of this solution that the round-trip time of the authentication messages can be reduced and the scalability can be enhanced, however, in many cases, the physical protection can not be assured for these special mesh routers which usually need to store sensitive data. The authentication can be delegated to or performed by the APs themselves. This is the most scalable solution, therefore, we investigate it in more depth.

A solution based on ID-based cryptography is proposed in [2], where the authors exploit that the public-private key pairs can be used both for authentication and for key agreement with an off-line central authority. In this solution, fast handover can not be guaranteed when the handover is performed between two APs belonging to different operators.

Another approach is presented in [3]. The authors suggest a change in the port-based network access control operation of IEEE 802.1X. Instead of restricting the MC to authentication messages through the uncontrolled port, the current AP allows MCs access to normal data traffic via a dynamically established tunnel between the current and the previous AP. The tunnel remains alive until the authentication is completed. This solution requires to change the current standard.

In [4] and [5], the authors propose a solution where the current AP issues a credential which can be used to certify MC's authenticity for the next AP. The authors propose to perform a full authentication after the lightweight credential based authorization. This mechanism requires of MCs to trust APs belonging to other operators when issuing credentials.

As none of the proposed mechanisms can fulfill all the requirements of the authentication process in EU-MESH networks, we suggest and investigate a new mechanism in the upcoming sections.

3. Our proposal

In this section, we suggest two certificate based authentication protocols for EU-MESH networks. First, we describe the architecture of the certificate based authentication protocols. Then, we investigate the speed characteristic of some classic crypto-primitives. After introducing a nonce-based and a timestamp-based authentication method, we define what public key algorithms and key sizes to use during the authentication in order to fulfill the general security requirements while still ensuring a short authentication delay during the handover.

3.1. Architecture

In our certificate based authentication and access control scheme, each operator maintains its own certificate authority service (CA). Each CA is responsible for issuing certificates for APs belonging to the operator and issuing certificates to their subscribers. The CA also maintains the certificate revocation list (CRL).

The operators which decide to cooperate issue cross certificates for each other's CA. With the cross-certificates, entities (subscribers or APs) can perform certificate based authentication and key exchange mechanisms even if they belong to different operators.

We suggest to handle the revocation in different ways depending on whether a certificate is issued to a MC or an AP. Maintaining CRL suits very well to APs because they have permanent connection to the CA. In contrast to this, MCs may not access the CRLs before

they connect to the mesh network. For this reason, MCs cannot verify CRLs efficiently, and to overcome this problem, we proposed that the AP public keys are very short-term (e.g., one day), and no CRL is maintained for them.

Regarding the certificate format, we rely on X.509, because it is a widely used standard.

3.2. Design rationale

Here, we investigate the properties of the public key based cryptographic algorithms based on publicized benchmarks [6] and own measurements. We considered the following key exchange, digital signature and encryption algorithms: Diffie-Hellman (DH), Elliptic Curve DH (ECDH), RSA, DSA, EC-DSA, EC-ElGamal.

Benchmarks showed that the elliptic curve based solutions are not beneficial because these algorithms are slower than the classical ones at similar security levels. In the case of DH key exchange algorithm, the computational complexity is balanced and it is as large as the private key operation of RSA. Furthermore, DH does not provide authenticity, and all together would cause too long delay. Therefore, in what follows, we consider only the RSA and the DSA algorithms.

In the case of RSA, the public key operations (encryption and digital signature verification) are quick operations when the exponent is relatively small (typically 65537), while the private key operations (decryption and digital signature generation) are three order of magnitude slower. In contrast to this, the digital signature generation with DSA with some precalculation can be performed very quickly, while the verification is three order of magnitude slower.

The latency of a public-key cryptography operation on one block mainly depends on the key size of the algorithm and on the performance of the device which performs the algorithm. There is always a trade-off between the speed of the algorithms and the level of the security. Nowadays, e.g. RSA with 512 bit key size is secure for 1 hour [7], and with 1024 bit for 1 year. In our proposals, we consider only these two key-sizes because the operations with 256 bit long keys are insecure and with 2048 bit long keys cause intolerable delays in the authentication process.

3.3. Certification based authentication protocol using nonces

In order to minimize the authentication delay, we choose the Blake-Wilson and Menezes Provably Secure Key Transport Protocol [8] (BWM). This protocol has the minimal number of public key based computations as 1) one signature per each participants is a minimum to prove that each one is online and 2) another public key based crypto primitive is required to provide a secure key for the upcoming communication. Another reason to choose this protocol is that it was proven to be secure.

Here, we extend the original proposal only with the certificates:

1. $MC \rightarrow AP : MC, N_{MC}, Certs_{MC}$
2. $AP \rightarrow MC : Msg_1 = [AP, MC, N_{AP}, N_{MC}, E_{MC}(ID_{AP}, K)], S_{AP}(Msg_1), Certs_{AP}$
3. $MC \rightarrow AP : Msg_2 = [MC, AP, N_{AP}], S_{AP}(Msg_2)$

MC first sends its ID, a fresh nonce (N_{MC}), and relevant certificates ($Certs_{MC}$). After verifying the certificates, AP generates a key K and encrypts it with its ID using the encryption key of MC. This encrypted data is concatenated to the ID's of the participants and two nonces: one received from MC and one generated freshly (N_{AP}). The signature ($S_{AP}(Msg_1)$) is calculated over these data using AP's private key. The relevant certificates ($Cert_{AP}$) are included in the message. On the other side, MC verifies the signature and the certificates and compare if N_{MC} is the nonce that it sent in the first message. MC obtains the key K by decrypting the encrypted message. The encrypted message contains the ID of AP, MC has to check if it corresponds to the ID appearing in the certificates. MC, in the third message, sends the ID of MC and AP, N_{AP} and the signature of these data. Finally, AP verifies the signature and compares if the obtained nonce is the same that AP sent in the second message.

This protocol provides key authenticity and key freshness both for MC and AP. The protocol itself does not provide key confirmation, but our implementation will rely on standard IEEE 802.11i [9] which provides key confirmation through the 4-way handshake. The key is controlled only by AP in BWM. However, we propose an extension to BWM that ensures that no party can control the value of connection key:

$$K_{conn} = Hash(K, N_{MC}) \tag{1}$$

where $Hash()$ is a one-way function, therefore, AP is not able to choose K such that K_{conn} takes a requested value.

3.4. Certification based authentication protocol using timestamps

The verification of the certificates requires loosely synchronized clocks. Therefore, when timestamp based solution is used, no new requirement has to be fulfilled. Furthermore, it needs fewer random bits and the signed timestamps can be used as a basis of the accounting.

Similarly to the BWM Protocol, we propose a timestamp based scheme which uses two digital signatures and a random number encryption:

1. $MC \rightarrow AP : Msg_1 = [MC, AP, t_{MC}],$
   $S_{MC}(Msg_1), Certs_{MC}$
2. $AP \rightarrow MC : Msg_2 = [MC, AP, t_{AP}, E_{MC}(K)],$
   $S_{AP}(Hash(\text{First message}), Msg_2),$
   $Certs_{AP}$

First, MC sends its timestamp t_{MC} signed with its private key. The first message contains the IDs of the participants and the relevant certificates ($Certs_{MC}$). After AP has checked if the difference between t_{MC} and t_{AP} (AP's currently generated timestamp) is below a threshold and the IDs are correct, it verifies the signature and the certificates. AP creates a message containing the IDs, t_{AP} and an encryption of a securely generated key K using MC's public key. The message sent back to MC contains a signature over these data and the hash value of the message received from MC. The relevant certificates ($Certs_{AP}$) are also included. MC verifies the signature and the certificates, and checks the difference between the clocks. If the IDs agree with the value sent in the first message, MC decrypts key K.

This schemes provides key authenticity and key freshness both for MC and AP, but no key confirmation. The key is controlled by both parties if it is calculated in the following way:

$$K_{conn} = Hash(K, t_{MC}) \tag{2}$$

3.5. Public key algorithms and key parameters

So far, we did not investigate the parameters of the public key algorithms and the certificates. In both protocols, a MC needs a public-private key pair for the encryption (Q_{MC}) and another one for the digital signature (P_{MC}). APs only require a public-private key pair for digital signature (P_{AP}).

The RSA algorithm suits very well to the digital signature of certificates because even though the signing operation needs a lot of time, it is not performed in a time critical period. While the verification is very fast and it is performed during the time critical handover.

As we have already stated, we use X.509 certificates because of compatibility reasons. Basically, it does not permit certifying with two different public keys in one certificate, however that would decrease the latency of the certificate verification on the AP side as the MC uses two different keys. Hence, we define a special extension for one of the certificates. There, the hash value of the other certificate can be added. With this mechanism, the two certificate verification can be reduced to one verification and one hash value computation.

Regarding to the AP's and MC's public key parameters, we differentiate between two cases: 1) when the MC is more powerful than the AP and 2) when the difference is less significant.

When the MC has more power than the AP (which is a typical case if we consider laptop computers as MCs), the MC uses RSA both for digital signature and for encryption, while the AP generates digital signature with DSA. In that case all the computationally intensive operations (private key operations with RSA and digital signature verification with DSA) are shifted to the powerful MC, whereas, the lightweight operations are performed by the AP.

The public keys of the MCs, as we defined earlier, are long term keys. Therefore, we chose 1024 bit long public-private keys. The size of APs' public key are mid-term as they can change them frequently. We also chose 1024 bit long keys for mid-term keys.

The $Certs_{AP}$ consists of the certificate of P_{AP} and optionally a cross certificate if the AP is not maintained by the MC's operator. The $Certs_{MC}$ consists of the certificate of P_{MC} and Q_{MC} and optionally a cross certificate.

Note that a less powerful MC is not able to perform all the computing intensive operations. Therefore, we propose another technique to reduce the delay of the whole protocol instead maybe at the cost of some precomputation by both participants.

The idea is based on speeding up the digital signature operations by using short keys. These short keys are weak, but they have a very short lifetime, such that they surely expire by the time they can be broken.

The weak keys are generated by the participants before the handover happens. In fact, MCs and APs issue certificates themselves (for $P_{MC}^{(w)}$ and $P_{AP}^{(w)}$, respectively). We have to emphasize that these certificates are not self signed certificates but new elements of certificate chains generated by a MC or an AP. The validity of the certificates are short-term, therefore, maintaining of CRL is not required for implementing this mechanism. Note that in this mechanism, the target AP and the MC which will perform the handover

do not need to communicate with each other or to obtain some information about each other, because the certificates are issuer specific. The certificates of the weak keys are signed with RSA so they can be verified very quickly.

Note that we cannot use weak keys for encryption, because the encryption hides information from attackers and after revealing the private part of the weak key, the hidden information can be revealed, too.

We suggest to use 512 bit weak long keys as short-term keys which seams to be the best tradeoff between the validity time and the computational overhead.

The time synchronization needs to be performed in a secure way, otherwise an attacker can make a MC or AP to accept an already expired certificate of an already broken public-private key pair. However, this is out of scope of this paper.

The $Certs_{AP}$ consists of the certificate of P_{AP}, $P_{MC}^{(w)}$ and optionally a cross certificate. The $Certs_{MC}$ consists of the certificate of Q_{MC}, P_{MC}, $P_{MC}^{(w)}$ and optionally a cross certificate.

4. Evaluation

Implementation. We created a proof-of-concept implementation. We embedded the authentication messages into EAP (Extensible Authentication Protocol) frames [10]. EAP messages are embedded into EAPOL messages in IEEE 802.1X [11] which is referred by IEEE 802.11i, the current standard solution for the Wi-fi authentication. K_{conn} is used as a Pairwise Master Key defined in IEEE 802.11i.

The hostapd on the AP side and wpa_supplicant [12] on the MC side gave an extensible framework for our proof-of-concept EAP implementation. We caught the events sents by wpa_supplicant when authentication starts and successfully ends. We measured the elapsed time between these two events getting the authentication delay. Note that we did not consider the delay of 4-way handshake, because it is independent of the authentication method and its delay has been already investigated in other papers.

Testbed. We investigated the authentication delay in different scenarios. In each cases, the AP was a MikroTik Routerboard 133 (175 MHz MIPS32 CPU, 32 MB memory) with OpenWRT (r11349, kernel v2.6.28.6) installed on it. In order to analyze how the MC performance affects the authentication delay, we used three different MCs: 1) high performance (Dell Inspiron 6000 laptop with 1.86 GHz 32 bit CPU), 2) moderate performance (Same laptop with the CPU running at 800 MHz), and 3) low performance (another MikroTik router with same parameters as the AP has).

We compared our proposal to classical widely used solutions (e.g. EAP-TLS, EAP-TTLS) with authentication servers (AS). For these cases, we installed hostapd as a stand alone Radius server on a PC (with Core2Duo 6400 2.13 GHz CPU, 1 Gb RAM, 32 bit Linux distribution, and kernel v2.6.28). In these scenarios, we connect the AS to the AP with direct link, thus, the roundtrip time between the AS and the MC is minimized.

The type of the wireless card was Atheros AR5414 and Intel 2915 in the case of MikroTik Routerboard and Dell laptop, respectively. The AP and MC communicated through 11g link.

Authentication delay. In this paper, we proposed a nonce based (NONCE) and a timestamp (TIME) based authentication scheme with two different certificate sets: one for powerful MCs and another one for constraint MCs (denoted by p and c in the index of the protocol name). We compared all these four authentication proposals to the 1) centralized EAP-TTLS with EAP-MD5 inside (TTLS-md5), 2) centralized EAP-TLS (TLS_{as}), and 3) distributed EAP-TLS (TLS_{ap}). Note that EAP-TLS does not require central subscriber manegement, because it uses only certificates for the authentication and key exchange. Therefore, the TLS connection establishment can be performed at the APs themselves. This is why we differentiated between the centralized and distributed EAP-TLS. In these methods, we used the same certificates and RSA public-private keys as we did in our proposed methods, with pre-generated 1024 bit Diffie-Hellman.

We compared the seven authentication scenarios with three different MC devices. We measured each case 100 times and calculated the average and the standard deviation. The results can be seen in Figure 1. On the horizontal axis, different protocols in different scenarios can be seen, while on the vertical axis, the authentication delay is shown. In each scenario, the different bars correspond to the measurements made with the different MC devices. Note that in some cases, the authentication delay was such long that we do not show with bars the value, instead we write explicitly the average value on the top of the reduced bar.

As one can see, each of our mechanisms significantly reduced the authentication delay compared to the centralized authentication methods (TTLS-md5 and TLS_{as}) where the AS is a powerful entity in contrast to our mechanism where the AP has limited performance. Furthermore, in the case of the considered centralized methods, the roundtrip time is minimized which, in a real application, may increase with the latency caused by some wireless hops in the mesh network and with the latency caused by the wired network. The authen-

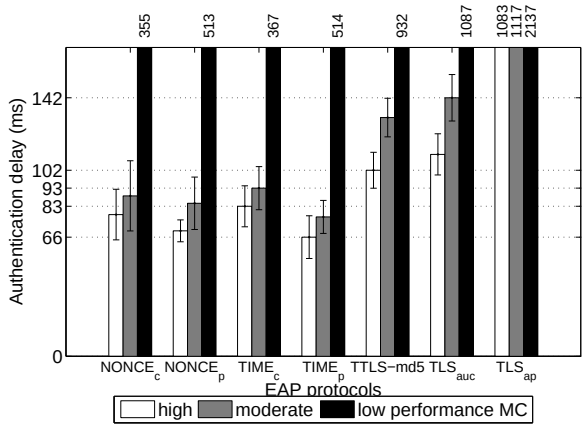

Figure 1. Average authentication delay with the estimated standard deviation

tication delay in the case of TLS_{ap} is even larger, because the TLS was not designed for fast connection establishment on constraint devices.

All of the four proposed mechanisms have very similar delays. What we can read from Figure 1 is that the weak key mechanism increases the authentication delay when the MC has high or moderate performance. The reason is that the verification of the added certificates and digital signature generation with RSA at the AP side lasts longer than the DSA verification and RSA digital signature generation with larger key at the MC side because the MC has more computing capacity. But e.g., comparing $TIME_c$ and $TIME_p$ bars in Figure 1, it shows that the weak key mechanism has significant benefit when the MC has low performance, and it reduced the authentication delay by 30% on average in the considered scenario.

5. Conclusion

In this paper, we proposed two authentication protocols that support fast handover in multi-operator maintained wireless mesh networks. For both schemes, we proposed two public key sets: one for a powerful mesh client and one for a constraint mesh client. In the former set, the computationally intensive operations are shifted to the mesh client, while in the latter certificate set, we proposed the usage of weak keys and short-term certificates for digital signature. We implemented a proof-of-concept, integrated it into the EAP framework, and measured the authentication delay comparing with current widely used centralized authentication mechanisms such as EAP-TLS and EAP-TTLS. Our solutions were considerably faster in all considered scenarios. Furthermore, our mechanisms satisfy special requirements relating to the multi-operator environment.

Acknowledgement. This work was supported in part by the European Commission in the context of the 7th Framework Programme through the EU-MESH Project (www.eu-mesh.eu) and in part by the Mobile Innovation Center (www.mik.bme.hu).

References

[1] I. Askoxylakis, B. Bencsáth, L. Buttyán, L. Dóra, V. Siris, D. Szili, and I. Vajda, "Securing Multi-operator Based QoS-aware Mesh Networks: Requirements and Design Options," *Wireless Communications and Mobile Computing (Special Issue on QoS and Security in Wireless Networks)*, 2009.

[2] Y. Zhang and Y. Fang, "A secure authentication and billing architecture for wireless mesh networks," *Wireless Networks*, vol. 13, no. 5, pp. 663–678, 2007.

[3] J.-J. Chen, Y.-C. Tseng, and H.-W. Lee, "A Seamless Handoff Mechanism for IEEE 802.11 WLANs Supporting IEEE 802.11i Security Enhancements," in *Proc. of IEEE APWCS*, Hsinchu, Taiwan, 2007.

[4] T. Chen, G. Schäfer, C. Fan, S. Adams, M. Sortais, and A. Wolisz, "Denial of Service Protection for Optimized and QoS-aware Handover Based on Localized Cookies," in *Proc. of EW'04*, Barcelona, Spain, 2004.

[5] T. Aura and M. Roe, "Reducing Reauthentication Delay in Wireless Networks," in *Proc. of IEEE SecureComm'05*, Athens, Greece, 2005.

[6] "D.VAM.14, Performing Benchmarks," IST-2002-507932, ECRYPT, European Network of Excellence in Cryptology, 2008.

[7] "D.SPA.28, ECRYPT Yearly Report on Algorithms and Keysizes (2007-2008)," IST-2002-507932, ECRYPT, European Network of Excellence in Cryptology, 2008.

[8] S. Blake-Wilson and A. Menezes, "Entity authentication and authenticated key transport protocols employing asymmetric techniques," in *Proc. of the 5th International Workshop on Security Protocols*, London, UK, 1998, pp. 137–158.

[9] IEEE Std 802.11i™, "Medium Access Control (MAC) security enhancements, amendment 6 to IEEE Standard for local and metropolitan area networks part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications." July 2004.

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

[11] IEEE Std 802.1X-2001, "IEEE Standard for Local and Metropolitan Area Networks - Port-Based Network Access Control," June 2001.

[12] J. Malinen, "WPA/RSN Supplicant (wpa_supplicant) and WPA/RSN/EAP Authenticator (hostapd) v0.6.7," http://hostap.epitest.fi/, 2009.