An authentication scheme for fast handover between WiFi access points

(Invited paper)

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

ABSTRACT

In this paper, we propose an authentication scheme that is designed to reduce the authentication delay during a WiFi handover process. We observe that the largest part of the delay is due to the remote communications between the access point and the AAA Server that authorizes the access to the network. In order to eliminate remote communications, our scheme uses pre-authorization, and it pre-distributes authentication information to the access points that are the potential targets of a future handover. This ensures that only local communications (between the Mobile Station and the access point) take place during the handover itself. We describe the design of our scheme, as well as report on a proof-of-concept implementation. Our validation results show that our scheme breaks the dependency of the authentication delay on the round-trip time between the access point and the AAA Server. This makes our scheme applicable in real time applications such as telephony and video streaming for WiFi users.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

; K.6.4 [**Management of Computing and Information Systems**]: System Management—*Decentralization*

; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*

General Terms

Measurement, Security, Design

Keywords

Fast handover, IEEE 802.11i, EAP-SIM

1. INTRODUCTION

In WiFi networks that use the Infrastructure Mode of the IEEE 802.11 standard [5], each mobile station (STA) is associated with an access point (AP) that provides access to the fixed network infrastructure (e.g., the Internet). When a STA moves, it may need to change the AP it is associated with, because each AP covers only a limited geographical area. This process is called handover. The more WiFi networks are used in telephony and multimedia applications the more important such handovers are becoming. In particular, there is a need to speed up the handover process such that it does not interrupt application level sessions.

For instance, the popular Skype application opened the possibility of using WiFi enabled PDAs as cell phones in areas with WiFi coverage. This application needs a continuous connection to the Internet even if the user is moving across different WiFi networks. Another example is video streaming, although it can tolerate longer gaps in connectivity thanks to its buffering mechanism.

The handover process is composed of four main phases: (i) detecting the possible set of next APs the handover could be aimed at (also called probing phase), (ii) choosing the destination AP, (iii) associating with that AP, and finally (iv) (re-)authenticating the STA to the network. In this paper, we focus on the problem of speeding up the fourth, authentication phase.

Authentication of the mobile stations is an important security requirement in WiFi networks. In particular, due to the lack of a physical connection between the STA and the AP, authentication becomes indispensable for controlling access to the network. However, the authentication mechanisms used in WiFi were not designed to be exceptionally fast, and they are unable to guarantee low handover latencies needed by today's real-time applications. In this paper, we show that this is mainly due to the centralized nature of the authentication procedure used in WiFi. In order to remedy this situation, we propose a decentralized authentication scheme that substantially reduces the latency caused by the authentication phase of the handover process.

The organization of the paper is the following: In Section 2, we describe the WiFi handover process focusing on the authentication phase. In the same section, we also examine and measure the latencies caused by today's leading authen-

Conference name: WICON 2007, October 22-24, 2007, Austin, Texas, USA

tication mechanisms (central RADIUS authentication server with various EAP (Extensible Authentication Protocol) and PEAP (Protected EAP) methods). In Section 3, we review some of the already existing solutions for speeding up the authentication phase, and identify the advantages and the disadvantages of each method. In Section 4, we introduce the EAP-SIM authentication method in more details, because our proposed authentication scheme will be based on this method. In Section 5, we identify some design requirements, and we present our proposed approach and protocols in Section 6. We describe the proof-of-concept implementation of the proposed mechanisms in Section 7. Finally, Section 8 contains the results of the measurements that we performed for the purpose of validating our solution. The results show that the proposed decentralized approach substantially reduces the latency introduced by the authentication phase of the handover process.

2. HANDOVER IN WIFI

Figure 1 gives an overview of the handover process showing the three main participants:

- the supplicant mobile station (STA);
- the access point (AP); and
- the authentication, authorization and accounting (AAA) server.

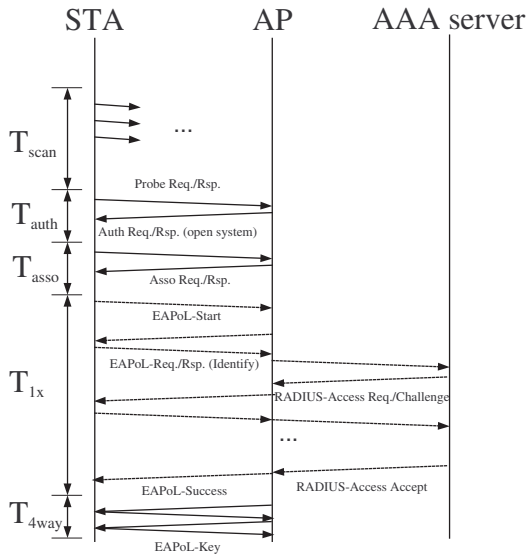

Figure 1: Overview of the handover process in WiFi networks

The latency caused by each step of the handover is marked on the left side of Figure 1, where T_{scan}, T_{auth}, T_{asso}, T_{1x}, T_{4way} correspond to the latency caused by the scanning phase, the open authentication, the association, the IEEE 802.1X [9] authentication, and the four-way handshake, respectively. The typical communication messages of the protocols involved in the process are represented by the arrows between the vertical lines.

The first phase of the handover process consists in deciding if there is a need for changing the AP, and if so, which AP the STA should be next associated with. This phase can last several seconds, but fortunately, most wireless LAN cards can do this without actually tearing down the connection with the currently used AP. The further study of this phase is beyond the scope of this paper.

The next phase of the handover process contains an empty authentication step, which is the legacy of WEP (Wired Equivalent Privacy), the security architecture specified in the original 802.11 standard. This phase does not actually provide any security, and it takes a very short time.

The next phase is the association phase, wherein the STA establishes a logical connection to the AP. The most important task of this phase is to inform the wired network about the fact that the given STA can now be reached through the new AP. The time needed for the association is negligible, so it is unnecessary to waste any efforts to speed up this phase.

The real authentication phase starts after the association phase. In this phase, the STA authenticates itself to the AAA server, which also helps to set up a shared session key between the STA and the AP. As we will see later, this phase can take a considerable amount of time, especially if the AAA server is remote.

Finally, the STA and the AP executes a four-way handshake, whereby they confirm the knowledge of the session key to each other, and they also derive new keys from the shared session key for various purposes. The four-way handshake is a necessary process in handover. It cannot shortened, but does not take too much time.

In [11], Aliman and Aboba examined the possible latencies caused by the above described phases of the handover process (they also examined phases in upper layers that are not covered in this paper), and established a problem space showing that it is physically possible to achieve seamless handover if the STA is moving with the velocity of a pedestrian. They also showed that the authentication phase is responsible for a large (if not the largest) part of the overall latency, and it is, therefore, a good idea to speed up this phase.

We also carried out some measurements of the dominating WPA (WiFi Protected Access) authentication protocols under laboratory conditions. In our experiments, we used a Linksys WRT54GSv4 access point, a FreeRadius 1.1.5 [1] RADIUS server and a wpa_supplicant 0.5.7 [18] running on a Dell Inspiron 6000 notebook and a desktop PC with Core2Duo 6400 processor. The examined authentication mechanisms included the EAP-TTLS protocol with MD5 inner authentication, the EAP-TLS protocol (for a detailed performance analysis see e.g. [21]), the PEAP protocol with MD5 and MsChapV2 inner authentication, and the EAP-SIM protocol. For each of these methods, we measured the latency of 100 handovers. We conducted two separate measurements for each method mentioned above: one with an authentication server at the same physical location (same room), and another with a remote server (being in a different part of the city). The latter case models a possible

real-world scenario where a wireless hot-spot operator runs several hot-spots and uses a single, central AAA server.

We found that the round trip time between the AP and the AAA server has an overwhelming impact on the authentication delay, and that the latency caused by the necessary cryptographic computations is negligible. The average delay obtained in the various cases are shown in Figures 2. As we can see, the latency can be 5-10 times larger when the AAA server is remote.

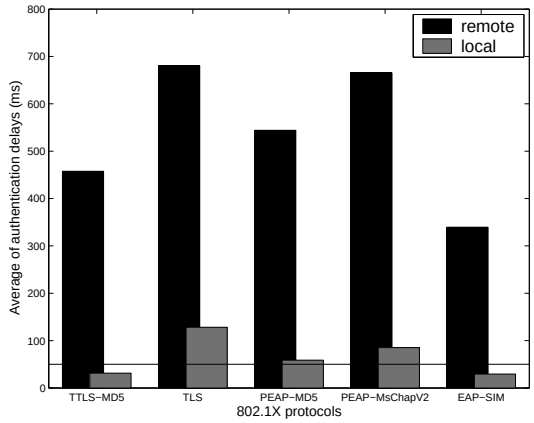

Figure 2: Average of the delays caused by the authentication phase during the handover process when using different existing authentication protocols

After taking a survey of the handover, we expect considerable decrease of latency improving the real authentication phase. In fact, the real authentication phase has two objectives: (i) corroborating the identity of the accessing STA (i.e., authentication), and (2) checking if the STA is allowed to use the network (authorization). Our approach will be to separate these two tasks, and let the AAA server perform the authorization and prepare the authentication while the STA is still associated with the currently used AP. In this way, when the handover takes place, the new AP only needs to authenticate the STA locally, which can be done very efficiently.

3. RELATED WORK

With the growing need for mobility, handover became a hot topic in the scientific literature in the past few years. Most of the proposed solutions are able to shrink the handover latency to an acceptable value (somewhere around 50 ms), but they usually give up a bit of security for the sake of fast handover, or they require special features not present in today's average networks.

IEEE itself addressed the problem in the 802.11i standard [7], where pre-authentication with key caching is proposed as the solution. The idea of the pre-authentication approach is to execute the entire authentication procedure between the STA and the new AP while the STA is still associated with the old AP (i.e., before the handover takes place). The STA and the new AP then cache the resulting Pairwise Master Key (PMK), and they run only the four-way handshake protocol when the handover actually takes place.

This solution has its advantages: the authentication can take as much time as needed (within reasonable limits), and the solution does not depend on the used authentication mechanism. Moreover, pre-authentication is part of the WPA2 standard ensuring that most new APs and mobil devices will implement this feature. The downside is that pre-authentication requires link layer communication between the participating APs. This requirement is not so easy to meet if the APs are located in different networks (possibly controlled by different operators). GRE [16], Ethernet bridging [8] or Ethernet over IP could provide a solution to this problem, but the usage of these requires serious trust between the operators (typically, they must share their LANs), and special firmware on the APs. In addition, there are scenarios where necessary connectivity is not available to support "make before break" communications [10]. In conclusion, pre-authentication is the prevailing solution when the handover happens between APs of the same network, but it is not suitable or not easily adoptable for inter-operator handovers.

The Inter Access Point Protocol (IAPP) [6] is another possible solution, proposed by the IEEE too. The protocol came to life as an IEEE recommended practice 802.11f in 2003 but was revoked in 2006. However the main idea behind its design is still considerable: instead of re-authentication, the current AP sends a so called security context (basically the PMK) to the new AP. The solution is weaker than pre-authentication since it requires secure communication between the APs, and it is impossible to use for inter-operator handovers where the authentication servers are not the same.

Yet another approach to the same problem is proposed by Aura and Roe in [15]. Their solution uses a very fast but somewhat weaker authentication algorithm during the handover that initiates a strong, and potentially slow authentication protocol run (e.g., EAP-TLS) right after its successful execution. If the strong second protocol fails (meaning that the STA cheated in the first authentication), then the AP denies further access to the network. This allows unauthorized access to the network if the weak protocol is broken, but only for a few seconds as the strong protocol supposedly cannot be broken. This scheme does not require inter AP communication, it is easy to implement, and it can handle inter operator handovers too. On the other hand, the optimistic approach means at least some loss of security, which might be unacceptable for some business applications (it is unlikely that any major company would allow a possible one second access to its intranet).

Finally, a pro-active key distribution mechanism is proposed by Arbaugh *et al.* in [12]. The idea is to use a neighbor graph to determine which APs are possible targets of a handover, and to distribute keying material to those APs from the AAA server. The scheme's only disadvantage is that the moving station is required to use the PMK shared with the current AP to construct the new PMK meaning that if an adversary can somehow break an old PMK, then she will be able to follow the STAs move in the network and easily calculate the new PMKs in use.

4. EAP-SIM

In this subsection, we present the EAP-SIM protocol in more detail, because our proposed solution is based on that. The detailed description of the EAP protocol itself and its use in a RADIUS environment is beyond the scope of this paper; good documentation of these can be found in [13, 14, 22].

EAP-SIM is described in RFC 4186 [17]. Its authentication mechanism is based on the scheme used in GSM networks for subscriber authentication. GSM networks are handling millions of seamless handovers every day which makes EAP-SIM a good candidate for a fast handover mechanism in WiFi networks.

In the case of GSM, the subscriber and the home network are sharing a key K_i stored in a smart card called Subscriber Identity Module (SIM) on the subscriber's side and on a central server called Authentication Center (AuC) on the network side. Subscribers are identified by a number called IMSI which is also stored in the SIM card.

As the subscriber moves from base-station to base-station, she has to re-authenticate herself to the network. This re-authentication is based on a simple challenge-response protocol and the subscriber's key K_i. However, this process should be very fast so there is no time to communicate with the (possibly remote) AuC server. Furthermore, it is not feasible to send out all the K_i keys for all base-stations, therefore, the base-station should request some information needed to authenticate the subscriber from the AuC of the home network of the subscriber. More specifically, the AuC sends so called triplets to the base-station, where each triplet contains a random challenge value $RAND$, a response $SRES$, and a session key K_c. The values of the 64 bit long K_c and 32 bit long $SRES$ are calculated from $RAND$ and K_i using the confidential A3 and A8 algorithms as follows:

$$K_c = A8(RAND, K_i) \quad (1)$$

$$SRES = A3(RAND, K_i) \quad (2)$$

Then, the foreign base-station challenges the subscriber with $RAND$. The SIM card of the subscriber computes the response $SRES$ and sends it to the base-station. This is compared to $SRES$ of the corresponding triplet, and if the values match, then the subscriber is authenticated. Further communication over the wireless interface is encrypted with K_c.

The EAP-SIM protocol is an adoption of the above described GSM subscriber authentication protocol for WiFi networks, where the role of the subscriber is played by the STA and the role of the foreign base-station is played by the AAA server. The APs (as always) are blindly forwarding the communication between the STA and the AAA server.

The typical architecture of the EAP-SIM authentication is shown in Figure 3, while the details of the protocol with all the messages can be seen in Figure 4. Note that as the short keying material provided by one run of the GSM challenge-response protocol is not enough to serve as the PMK, the protocol is run three times (with different $RAND$ values), and the resulting K_c, K_c', K_c'' values are used to generate the

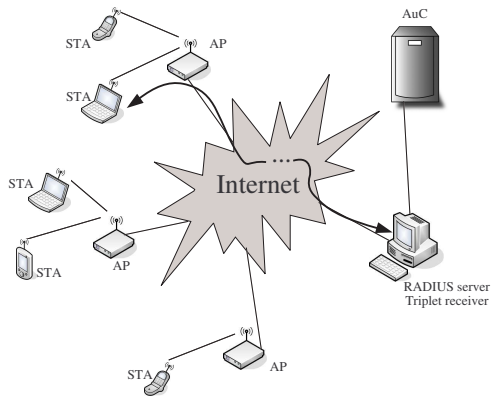

Figure 3: Architecture for EAP-SIM authentication

PMK as follows:

$$PMK = SHA1(Id|K_c|K_c'|K_c''|Nonce|Ver\_list|Sel\_ver) \quad (3)$$

Where $Nonce$ is a random number generated by the STA in the EAP-SIM process ensuring the freshness of the PMK, Ver_list is the supported EAP-SIM versions by the AAA server, and Sel_ver is the selected EAP-SIM version by the STA.

5. REQUIREMENTS

After reviewing the already existing solutions, we will now set up the goals that we are trying to achieve with our new authentication scheme. As we saw in Section 2, methods like EAP-TLS use a lot of messages to communicate with the AAA server, that is why the growth in the latency is so immense once we use a remote server. One solution could be to switch to a protocol using fewer messages, but that would not break the linear dependency of the authentication delay on the round trip time between the AP and the AAA server. Our main goal is *to break this linear dependency entirely* by eliminating any need of communication between the AAA server during the handover itself.

We have already seen that the various solutions proposed to reduce the authentication delay fall in two classes: one approach is to get as much work as possible done before the actual handover begins, whereas another approach is to transfer an already established security context (i.e., cryptographic keys) from the current AP to the new one.

We want to insist on having a new authentication of the STA in case of a handover because the scheme should work in a multi-operator environment where the necessary secure communication links between the participating APs might not exist. Therefore, we must reject the idea of a context-transfer.

Finally, an important requirement that our solution must meet is that it should not be any weaker than an original EAP-based challenge-response protocol like EAP-SIM. In other words, it should provide a sufficiently large PMK, and it should guarantee its freshness. It is also unacceptable to allow an adversary to get access to the network for even a short period of time.

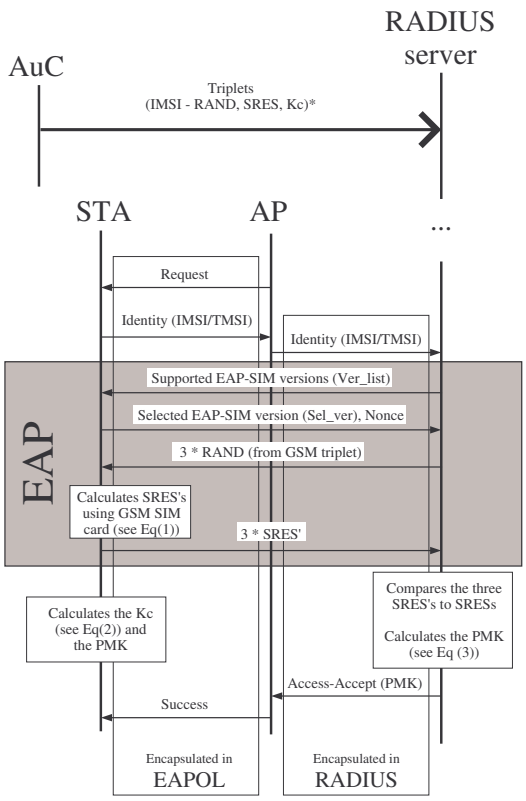

Figure 4: Process of EAP-SIM authentication

6. PROTOCOL DESCRIPTION

In order to fulfil the most important requirement (i.e., constant time re-authentication, regardless of the accessibility of the remote server), our scheme uses pre-authorization and it eliminates the need for communication with the remote server when the handover actually takes place. This solution will lead to a latency which is no longer in linear dependency with the round-trip time between the AP and the remote AAA server, which – as shown by our measurements – is responsible for the largest part of the overall handover latency.

The idea itself is simple: We saw in Section 4 that in GSM networks, the AuC server sends out triplets to foreign base stations, which can then authenticate the STA without any further need for remote communications (i.e., the AuC server performs pre-authorization). The implementation of EAP-SIM in the WiFi world, however, lost this advantage, because the triplets are sent out to the still remote RADIUS server instead of the APs themselves. Therefore, our main idea is to send out the triplets to the possible next APs before the handover takes place. The resulting architecture is illustrated in Figure 5 (compare this to Figure 3).

If a mobile device that is already connected to the network connects to another AP, the triplets required for its authentication will already be available at the new AP, and there is no need for remote communications. This distribution is the main point of our solution, because it ensures that the latency of the handover process stays under a threshold, even in an overloaded network, or with a remote central authentication server residing in another network.

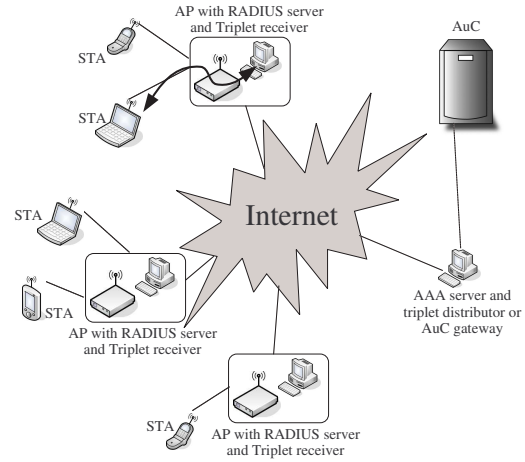

Figure 5: Architecture for our modified EAP-SIM authentication

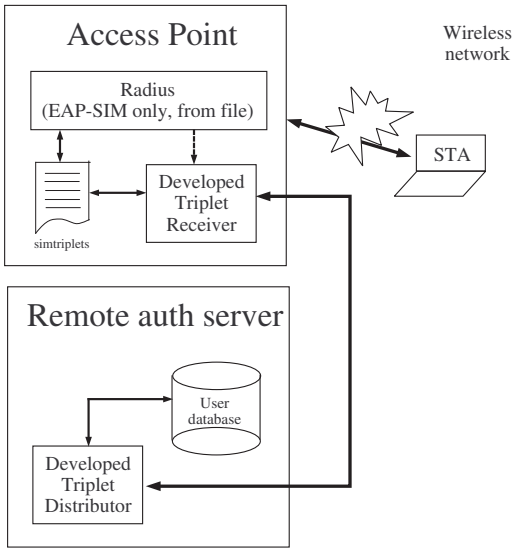

Figure 6: System model

Our system model, is shown in Figure 6. The main point is that we run a small footprint RADIUS server on the AP, which operates reliably in a resource constrained environment. This server is responsible for the SIM-based authentication. When a new authentication request arrives, this server checks if there are triplets corresponding to the requesting STA in the "simtriplets" database. In the normal case, when a handover is taking place, such triplets must already be available, as they are pre-loaded during pre-authorization when the STA is still associated with the old AP. If the triplets are found in the database, then authentication can take place locally. Otherwise, the RADIUS server running on the AP informs the Triplet Receiver (TR) application, which is responsible for managing the triplets. Triplet Receiver then connects to Triplet Distributor (TD) running on the remote authentication server and acquires

the required information. In this case, obviously, the scheme is not faster than the normal EAP-SIM protocol described in Section 4.

If the STA authenticates itself successfully, then the RADIUS server informs TR application, which, in turn, informs TD. Then, TD selects the access points which are potential roaming targets, and sends them the triplets required for the future authentication of the STA. This ensures that when a handover to one of these APs occurs, no communication with the remote server is needed.

One can argue that the usage of the RADIUS protocol is quite unnecessary in this situation, as the RADIUS server running on the AP does not hold the actual user database, and the RADIUS communication takes place inside the AP. We still stuck to the usage of RADIUS, because it ensures that existing AP firmware does not have to be altered in order to use our solution (apart from running a RADIUS server on the AP).

Another important issue is the secure communication link needed between the AP and the central AAA server. However, such a secure link is easy to implement as all communications between the AP and the remote AAA server go through Triplet Receiver and Triplet Distributor, and therefore, they can set up a long-term security association.

There should also be an algorithm in place responsible for deciding which APs in the network are possible handover targets. Many solutions to this problem can be found in the literature (see e.g., [20]). For simplicity, in our implementation, the server sends out triplets to all APs.

Finally, one may wonder why we did not use the AKA (Authentication and Key Agreement) protocol, which was developed for next generation UMTS networks. AKA brought two major novelties with respect to GSM authentication: integrity protection, and sequence numbering in order to prevent the re-use of challenge-response pairs. However, integrity protection has already been implemented in the WiFi environment (during the four-way handshake an integrity protection key is derived from the PMK). Thus, sequence numbering is the only feature we have to implement. This is needed because if an adversary somehow intercepts a valid triplet for a STA, it will be able to impersonate an AP to the STA who will have no means to know that it is not connecting to a legitim network. However, adding sequence numbering to GSM triplets does not require to adopt the entire UMTS AKA protocol.

7. IMPLEMENTATION

Our intention with implementing our decentralized authentication scheme was to provide a proof-of-concept implementation. In particular, we wanted to verify if our solution indeed eliminates the linear dependency of the latency on the round-trip time. We did not want to produce a production-ready solution, so we omitted some necessary features (for example we did not implement the secure communication link between the AP and AAA server).

We chose the open source FreeRADIUS and wpa supplicant software to be the basis of our implementation. Both have good developer documentation and a modular design which makes them easily adaptable to our special needs. Moreover, FreeRADIUS is known to run smoothly on some APs under the popular OpenWRT [3] operating system.

After choosing FreeRADIUS and wpa supplicant, we had to find a way to avoid the usage of real SIM cards in our proof-of-concept implementation. This means that we had to replace A3 and A8 algorithm to provide the $SRES$ and K_c values from $RAND$. We chose the following easily implementable solution for this purpose:

$$K_c = BITS_{0-63}(HMAC(RAND, K_i)) \quad (4)$$

$$SRES = BITS_{64-95}(HMAC(RAND, K_i)) \quad (5)$$

Note that the calculation of the PMK (see formula (3)) remains unchanged since it is not affected by the way the K_c values are generated.

The fact that we eliminated the usage of SIM card implies that the K_i key has to be stored on the STA device. We adopted the widely used technique storing the pre-shared K_i key in the configuration file of the wpa supplicant where the PIN was stored in the original version of the EAP-SIM.

As we mentioned in Section 6 the implementation of UMTS AKA's sequence numbering is essential in ensuring the security of our scheme. On the other hand we had to stick to the original EAP-SIM algorithm as much as possible so that already existing and well tested pieces of software like FreeRadius can be used with only limited changes. To meet these two conflicting requirements we decided to extend the length of the $RAND$ value to 256 bits (it is 128 bits long in the original version) and treat it's first 64 bits as sequence number. In this way, we will still have 192 random bits per triplet which should be enough for EAP-SIM to work properly. The last accepted sequence number will be stored in the STA by wpa supplicant and the STA will reject any triplet whose sequence number (the first 64 bits of the $RAND$) is less than the stored value. On the other side Triplet Distributor will increment the sequence number with every triplet issued. This means that the three triplets used for one run of the EAP-SIM protocol will have three different sequence numbers, each higher than the one stored in the STA. We emphasize one advantage of this solution: the RADIUS server does not need to be aware of the presence of the sequence number at all, the only change needed is setting the size of $RAND$.

Another important point is the communication protocols used by the Triplet manager applications. In a business environment this communication should be secure and as fast as possible. In our implementation we used a simple insecure character based TCP/IP protocol which can be easily replaced by any acceptable method in a production-ready version. The messages of our protocol are shown in Table 1.

As it can be seen in the system model, we had to choose some kind of database management for the storage of users and the triplets. The FreeRADIUS server was able to read the triplets only from text files, which has many disadvantages, the most important being that after the use of the triplets, the server does not delete them from the file. Moreover, searching, inserting, and deleting triplets are rather difficult

Table 1: Messages of the Triplet Receiver (TR)– Triplet Distributor (TD) communication

Direction	Message format and description
$TR \rightarrow TD$:	A<IMSI_15bytes> TR (running on an AP) notifies the central server that the user identified by IMSI connected successfully to the network. TD should not respond.
$TD \rightarrow TR$:	N<IMSI_15bytes & TRIPLETS_3*56bytes> TD sends 3 triplets to TR allowing it to authenticate a user identified by IMSI. TR should use the triplets immediately or store them in its internal database.
$TD \rightarrow TR$:	R<IMSI_15bytes> TD notifies TR that the user identified by IMSI is not authorized to use the network therefore its request should be rejected.
$TR \rightarrow TD$:	G<IMSI_15bytes> The message is sent by TR if a user identified by IMSI tried to connect to the network, but TR was unable to authenticate her due to the lack of necessary authentication information in the AP. TD should reply to this message either with an "N" or with an "R" message according to the user's authorization status.

with plain text files.

In the course of the implementation we have decided to use the SQLite function library [4] because of its small resource requirements. SQLite is a free software and it basically implements the whole SQL standard. The advantages of SQLite are that it is serverless and that the databases can be stored in the file system. This enables the usage of SQLite on the very limited platform of an AP.

We started the development with the study of the code of FreeRADIUS. According to the system model, we had to add the following functions:

- Searching triplets in SQLite database;
- Communicating with the local Triplet Receiver application;
- Informing the Triplet Receiver application in case of a successful authentication.

The first two functions were developed in a new module, based on the existing `rlm_sim_files` module. The solution to the third function was given by the `rlm_exec` module of FreeRADIUS, which is able (with appropriate configuration) to run any scripts after different events (e.g., a successful authentication).

After finishing the server, we worked on the wpa supplicant. Fortunately, there was not much to do, only to implement an algorithm to create the $SRES$ and K_c values from the $RAND$ value in order to replace the original SIM card handling functions. We have chosen the HMAC algorithm [19] from the openssl library [2] for implementing the computation in accordance with formula 4 and 5.

Finally, we have developed the Triplet manager applications. For database we used SQLite again, and the communication was done over TCP connections using the character based protocol described in Table 1.

8. VALIDATION

Our validation environment consisted of a Linksys WRT54GS v4.0 wireless AP (200MHz MIPS processor, 4MB Flash, 16MB RAM), a local and a remote FreeRADIUS server (running on a desktop computer with Core2Duo 6400 CPU and 1Gb memory), and a Dell Inspiron 6000 (Pentium M 1.86GHz with 1Gb memory) laptop as the mobile station. We simulated the handover by giving reassociation commands to the laptop's wireless client software (wpa supplicant). We conducted 100 simulated handovers in 4 different schemes: the original EAP-SIM protocol with a local (i) and a remote (ii) RADIUS server, (iii) our modified EAP-SIM protocol with the FreeRADIUS component running on a PC in the same LAN as the AP and finally (iv) our modified EAP-SIM protocol with the FreeRADIUS component running on the AP itself. Note that the location of the authentication server (local or remote) is not important in our scheme because there is no communication with the authentication server in the handover process.

The average, minimum and maximum values of our simulations are shown in Figures 7(a), 7(b), and 7(c), respectively. On the x-axis we listed the 4 simulation environments (as described above) and we put the measured values in milliseconds on the y-axis.

Figures 8(a), 8(b), 8(c) and 8(d) show the frequency of the simulation runs as a function of delay for the 4 different setups. On these Figures we put the authentication delay in milliseconds in the x-axis and the number of simulation runs with the given delay on the y-axis.

The results clearly show that our solution is capable of reducing the authentication delay below 55 ms even if the FreeRADIUS server runs on the resource limited AP. Our measurements with the FreeRADIUS running on a PC also prove that this latency will be even shorter with the growth of the AP's computational power. The linear dependency on the RTT between the AP and the authentication server is also broken, however this comes from the design of our protocol (if we do simulations with remote and local triplet-distributing servers we get the exact same results).

9. CONCLUSIONS

In this paper, we proposed an authentication scheme that is designed to reduce the authentication delay during a WiFi handover process. In particular, our scheme breaks the dependency of the authentication delay on the round-trip time between the access point and the AAA Server. We achieve this by eliminating the need for remote communications between the access point and the AAA Server through pre-authorization and pre-distribution of authentication information to the access points that are the potential targets of

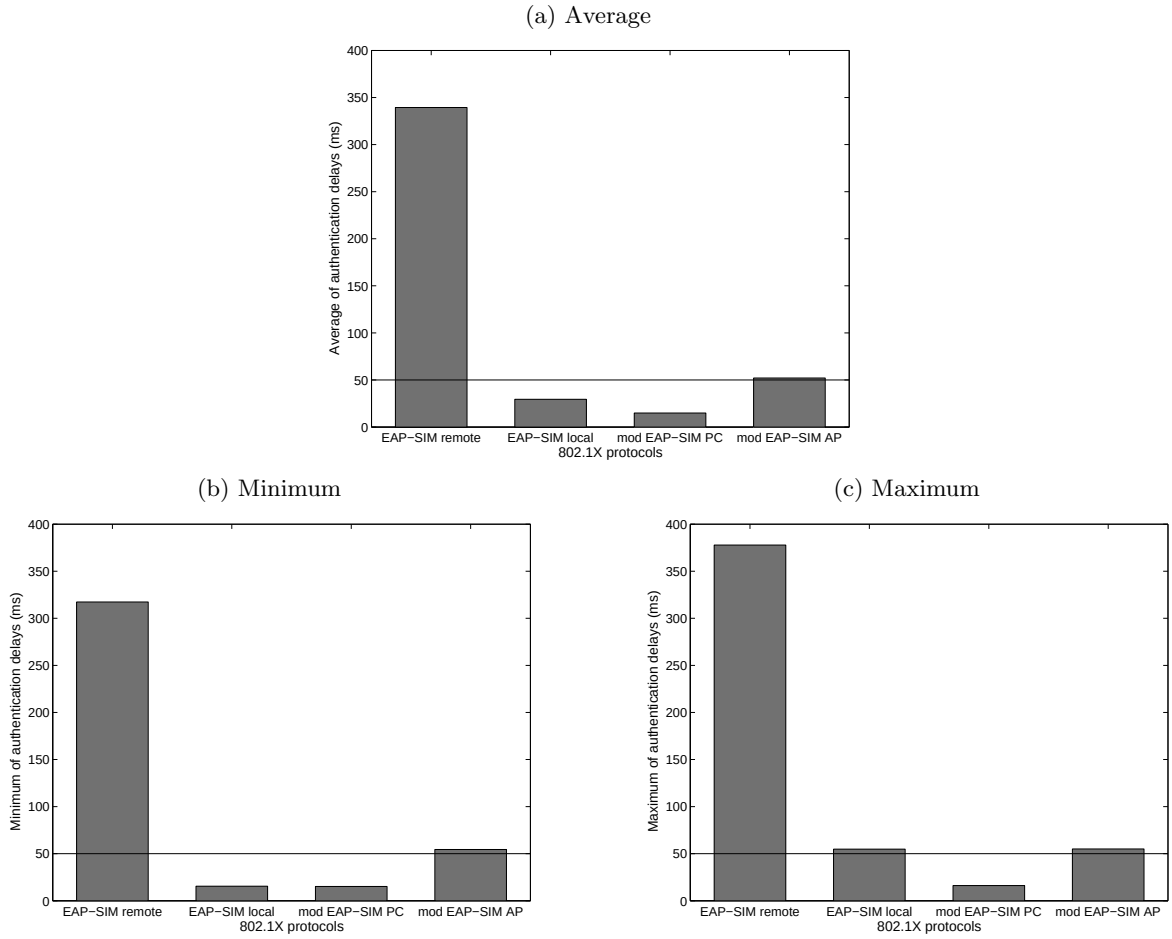

Figure 7: Authentication delay in the original and modified EAP-SIM protocol

a future handover.

Our solution is based on the EAP-SIM authentication protocol and it requires to run a small footprint RADIUS server on the access points. We developed a proof-of-concept implementation of the proposed scheme in order to demonstrate that it can be realized with a reasonable effort and without major changes to already available software.

By ensuring a very short re-authentication delay, our scheme can be a step forward from best-effort to QoS in the WiFi world. In particular, the primary application area of our scheme is related to real time systems such as telephony and video streaming for WiFi users.

10. ACKNOWLEDGMENTS

The work presented in this paper has been partially supported by the Mobile Innovation Center (www.mik.bme.hu) and the NKFP Messenger Project (www.messenger.mcl.hu). The authors are thankful to Zoltán Faigl, Győző Gódor, András Méhes, and Máté Szalay for his useful comments on an early version of this paper.

11. REFERENCES

[1] freeRADIUS: The world's most popular RADIUS Server. `http://www.freeradius.org`.

[2] OpenSSL. `http://www.openssl.org`.

[3] OpenWrt: Wireless Freedom. `http://openwrt.org/`.

[4] SQLite. `http://www.sqlite.org`.

[5] IEEE 802.11: Information Technology Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std. 802.11, ISO/IEC 8802-11. First edn., 1999.

[6] IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11. IEEE Std 802.11F, 2003.

[7] IEEE Standard for Information technology - Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 6: Medium

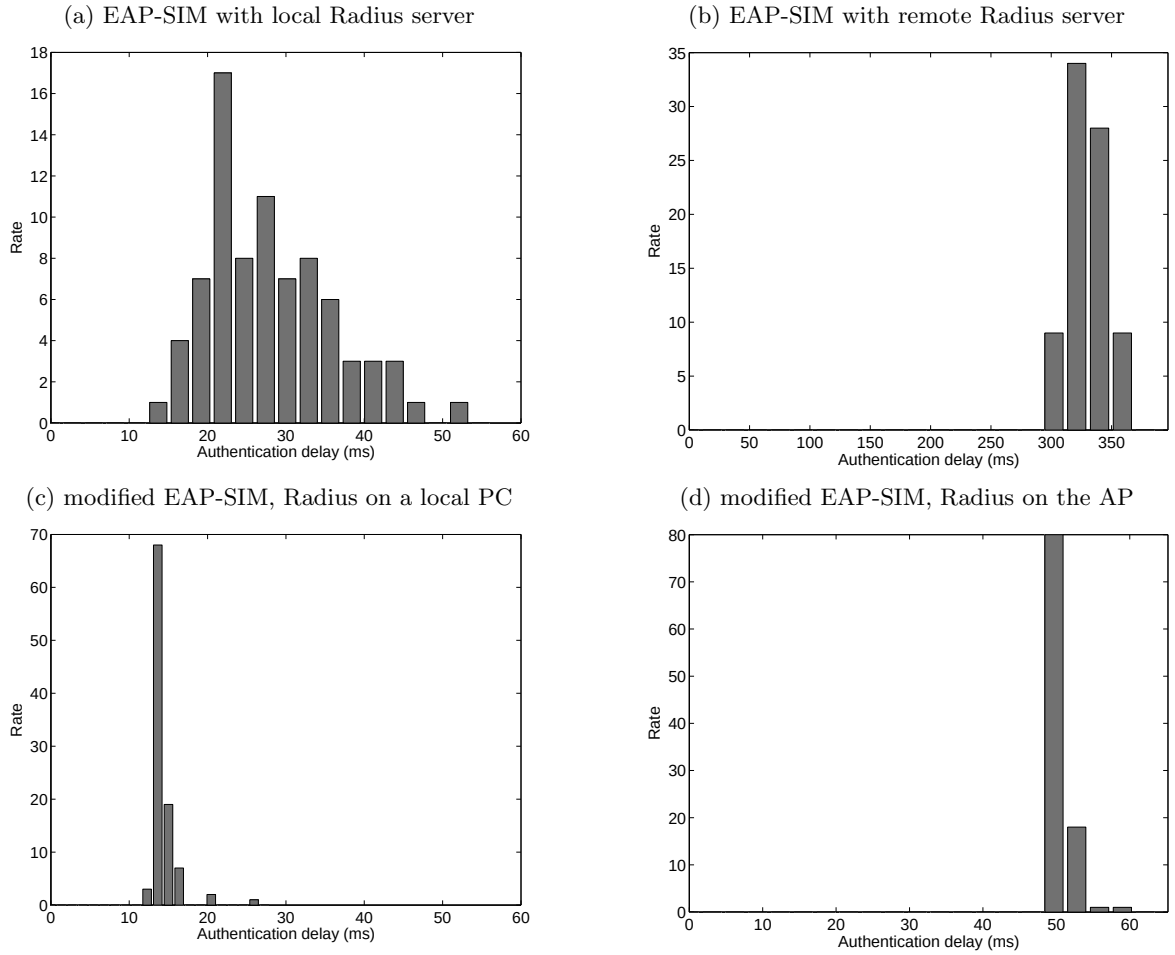

Figure 8: Distribution of authentication delays in the original and in the modified EAP-SIM protocol

Access Control (MAC) Security Enhancements. IEEE Std. 802.11i, 2004.

[8] IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges. IEEE Std. 802.1d-2004, 2004.

[9] IEEE Standard for Local and Metropolitan Area Networks: Port-Based Network Access Control. IEEE Std. 802.1X-2004, 2004.

[10] Description of the Handover Keying (HOKEY) IETF Working Group. `http://www.ietf.org/html.charters/hokey-charter.html`, 2007.

[11] B. Aboba A. Alimian. *Analysis of Roaming Techniques*. IEEE 802.11-04/0377r1, 2004.

[12] W. Arbaugh A. Mishra, M. Shin. Context Caching using Neighbor Graphs for Fast Hand-offs in a Wireless Network. *IEEE INFOCOM*, March 2004.

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

[14] B. Aboba and P. Calhoun. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). RFC 3579 (Informational), September 2003.

[15] Tuomas Aura and Michael Roe. Reducing Reauthentication Delay in Wireless Networks. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, pages 139–148, Washington, DC, USA, 2005. IEEE Computer Society.

[16] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina. Generic Routing Encapsulation (GRE). RFC 2784 (Proposed Standard), March 2000.

[17] H. Haverinen and J. Salowey. Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM). RFC 4186 (Informational), January 2006.

[18] Jouni Malinen. WPA Supplicant. `http://hostap.epitest.fi/wpa_supplicant/`.

[19] Ran Canetti Mihir Bellare and Hugo Krawczyk. Keying hash functions for message authentication. 1996.

[20] Sangheon Pack, Hakyung Jung, Taekyoung Kwon, and Yanghee Choi. SNC: a selective neighbor caching scheme for fast handoff in IEEE 802.11 wireless networks. *SIGMOBILE Mob. Comput. Commun.*

Rev., 9(4):39–49, 2005.

[21] Z. Faigl S. Lindskog, A. Brunstrom and K. Tóth. Providing Tunable Security Services: An IEEE 802.11i Example. In *1rst Workshop on Enterprise Network Security (WENS 2006*, August 28 2006.

[22] D. Stanley, J. Walker, and B. Aboba. Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs. RFC 4017 (Informational), March 2005.