Efficient Authenticated Wireless Roaming via Tunnels

Andreas Noack

Horst Görtz Institut für IT-Sicherheit
Ruhr University Bochum

Abstract. Wireless roaming means that a mobile device is able to switch from one network cell to another while keeping the link to active services. Recent researches [12] showed that it increases the security to establish an authenticated and confidential tunnel directly to a home network which then acts as service provider respectively proxy server for further external services. In this paper we extend the trust assumptions and formal security goals for *wireless roaming via tunnels* (WRT) that were given by Manulis et al.[7].

Additonally, we propose an efficient protocol that realizes the authentication and key agreement for establishing the secure tunnel, whereby considering the delay restrictions that are given by current multimedia services like VoIP or video streaming.

Furthermore we discuss the accounting problem and present a solution that ensures a fair accounting for the foreign network.

Keywords: Wireless networks, security, key agreement, mutual authentication, accounting.

1 Introduction

Wireless LAN is a very popular communication medium today, since it allows its users to be mobile while having access to all services they usually use in a wired LAN. Recent technologies like IEEE 802.11a/g/n also allow a very high bandwidth, so that the advantages from the wired alternative become smaller and smaller.

To let wireless LAN become even more attractive, the coverage has to be improved further on, so that everyone has everywhere access to his preferred services. Of course, it is not possible to realize a single wireless LAN that covers a whole city region. That means, it is necessary to work with several smaller wireless networks that may be operated by foreign network providers. Therefore a cooperation with foreign network providers is required.

There are three problems to solve:

1. When connecting to foreign wireless LAN providers, it is important to preserve the own security.

2. While switching between two wireless LAN cells, current running services like VoIP, video or audio streaming should not be affected.
3. The foreign wireless LAN provider clearly wants to get paid for the service he provides; that means, a fair accounting must be arranged.

Imagine a whole city covered with wireless nodes from private users. Most of them have a direct connection to the internet and are able to distribute their internet link over wireless LAN. There are several companies which want to provide seamless internet access in the whole city by using the given infrastructure. These companies offer an accounting model for all private users who share their internet connectivity, so that the companies' customers may use these internet links. The task is, to provide a network protocol that authenticates the companies' customers to the companies and offers fair accounting for the private users, that share their internet connection with the customers.

Sastry et. al [12] made a new proposal for the network structure that is needed for realizing a city-wide wireless LAN access. Shortly, they propose that a foreign network provider (in the following called $\mathcal{F}$) does only relay the traffic between the mobile node (called $\mathcal{M}$) and the home network (called $\mathcal{H}$) which then acts as a proxy server for all services, the mobile node wants to access. The communication between the mobile node and the home network is protected by a confidential and authenticated tunnel, to improve the security. The big advantage of this solution is, that the risk for the misuse of the foreign network's internet link drops to zero, because all services (including internet access) are provided by the home network. The single purpose of the foreign network $\mathcal{F}$ is to relay the tunnel data between the mobile node $\mathcal{M}$ and the home network $\mathcal{H}$.
Nevertheless, Sastry et. al did not propose a concrete implementation for this solution.

Manulis et. al [7] extended this idea with a concrete secure authentication and key establishment protocol for three parties. This protocol accomplishes mutual authentication between $\mathcal{M}$, $\mathcal{H}$ and $\mathcal{F}$, $\mathcal{H}$, which is necessary for the secure communication and can later be used for accounting purposes also. Their proposed protocol is not optimized for efficiency in terms of roaming.

We propose a new network protocol that is optimized for roaming, even when multimedia services like VoIP or video streaming are in use. This can be reached by improving the efficiency in comparison to the proposed protocol by Manulis et. al. Furthermore, we present a protocol for accounting purposes so that a commercial scenario can be realized easily.

2 Security Model

2.1 Protocol Participants and Keys

The protocol participants are namely the mobile device $\mathcal{M}$, a foreign network $\mathcal{F}$ and a home network $\mathcal{H}$. The user of the mobile device $\mathcal{M}$ has got a service contract with a home network $\mathcal{H}$, which gives him access to several provided services by $\mathcal{H}$, wherever an appropriate network infrastructure is given. An appropriate network infrastructure is realized through the nodes of the foreign network $\mathcal{F}$, that provide on the one side wireless access for all $\mathcal{M}$ and on the other side a fast link to the home network $\mathcal{H}$.

We assume, $\mathcal{M}$ and $\mathcal{H}$ are in possession of a common longterm key k_{MH} that is chosen with respect to the security parameter l.

For relaying data between $\mathcal{M}$ and $\mathcal{H}$, the foreign network wants to get paid. Therefore there is another contract between each foreign network $\mathcal{F}$ and home network $\mathcal{H}$. Because there may be a lot of different home networks and even more foreign networks, it is not efficient to provide a symmetric key between each foreign network and each home network.

For that reason, each foreign network $\mathcal{F}$ and home network $\mathcal{H}$ own a Diffie-Hellman public key pair $\{SK, PK\}$ which is chosen with regard to the security parameter l.

2.2 Instances and Protocol Sessions

The number of the mobile devices $\mathcal{M}$, foreign networks $\mathcal{F}$ and also home networks $\mathcal{H}$ may be very big, so it is likely that the same $\mathcal{F}$ *or* $\mathcal{H}$ (*or* even $\mathcal{M}$) are participants in several parallel protocol sessions. We want to extend this by saying that it is possible that there are different protocol sessions with the same $\mathcal{M}$, $\mathcal{F}$ *and* $\mathcal{H}$. The number of parallel protocol sessions is denoted as q (later used in the security analysis).

We claim that there is an unlimited number of instances of $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$, whereby denoting an instance as $\mathcal{X}_s$ with $\mathcal{X} \in \{\mathcal{M}, \mathcal{F}, \mathcal{H}\}$ and $s \in \boldsymbol{N}$. Three instances $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$ are called partnered when they have the same session id $SID := H, AID_M, F, r_H, r_M, r_F$ whereby H, AID_M, F are the identifiers of $\mathcal{H}$, $\mathcal{M}$, $\mathcal{F}$ and r_H, r_M, r_F are randomly chosen nonces of each participant.

An instance of $\mathcal{H}$, $\mathcal{M}$, $\mathcal{F}$ in a protocol session calls ACCEPT or ABORT upon the decision if the protocol execution was successful in respect to the protocol aims.

2.3 Trust Assumptions

Before protocol execution, the mobile device $\mathcal{M}$ and the home network $\mathcal{H}$ share some credentials that allow them to do a mutual authentication, which is necessary for establishing a trusted communication tunnel. Since $\mathcal{H}$ provides a service for $\mathcal{M}$, both parties must have a contract with each other, including on the one hand credentials and on the other hand rules for accounting and usage.

The foreign network $\mathcal{F}$ is responsible for the relay of the tunnel data between the mobile device $\mathcal{M}$ and the home network $\mathcal{H}$. Mutual authentication between $\mathcal{F}$ and $\mathcal{H}$ is required, because the foreign network $\mathcal{F}$ clearly wants to get paid for the forwarding service it provides and must therefore be aware of $\mathcal{H}$'s identity. Additionally the home network $\mathcal{H}$ wants to be sure about $\mathcal{F}$'s identity to realize a fair payment. Furthermore sharing credentials between $\mathcal{F}$ and $\mathcal{H}$ to support the accounting process may be necessary.

The mobile device $\mathcal{M}$ will be implicitly authenticated against the foreign network $\mathcal{F}$ due to the fact that $\mathcal{H}$ accepts in the protocol. The same applies for the foreign network $\mathcal{F}$ against $\mathcal{M}$, because the mobile device $\mathcal{M}$ is assured that $\mathcal{H}$ would not have been accepted when the authentication between $\mathcal{F}$ and $\mathcal{H}$ had failed.

2.4 Adversarial Model

The adversary $\mathcal{A}$ is modelled as a probabilistic polynomial time (PPT) machine and has full control over the communication and protocol invocations. $\mathcal{A}$ is allowed to do the following queries:

- **Invoke($\mathcal{X}$, m)**. Upon this query, a new instance $\mathcal{X}_s$ of $\mathcal{X} \in \{\mathcal{M}, \mathcal{F}, \mathcal{H}\}$ is created. Message m is sent to the new instance, whereby the answer is directed to the adversary $\mathcal{A}$.
- **Send($\mathcal{X}_s$, m)**. This query sends a message m to $\mathcal{X}_s$. When $\mathcal{X}_s$ has completed processing m, the response is sent back to $\mathcal{A}$. With the help of this query, $\mathcal{A}$'s control over the communication channel is modeled, since $\mathcal{A}$ is able to stay passive by honestly forwarding each message or to become active by modifying m or even injecting a new message.
- **Corrupt($\mathcal{X}$)**. As response to this query, $\mathcal{A}$ gets the longterm key of $\mathcal{X}$. That is k_M for $\mathcal{M}$, SK_F for $\mathcal{F}$ and $\{SK_H, k_M\ \forall \mathcal{M}\}$ for $\mathcal{H}$. When $\mathcal{X}$ becomes corrupted, all instances $\mathcal{X}_s$ of $\mathcal{X}$ become corrupted too.
- **RevealKey($\mathcal{X}_s$)**. If $\mathcal{X}_s$ has already accepted, the adversary $\mathcal{A}$ gets the session key as response to this query. The session key between $\mathcal{M}$ and $\mathcal{H}$ is k_{MH}, whereas the session key between $\mathcal{F}$ and $\mathcal{H}$ is denoted as k_{FH}.
- **TestKey($\mathcal{X}_s$)**. The adversary may query **TestKey()** to an accepted instance of a session. The instance $\mathcal{X}_s$ chooses a random bit b and answers with a random value on $b = 0$ and with the session key $\{k_{MH}, k_{FH}\}$ on $b = 1$.

2.5 Correctness

The authentication and key establishment protocol Π (Figure 1) is correct, when definition 1 holds.

Definition 1 (Correctness EAWRT). In the presence of a passive adversary, Π is correct when all parties $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$ have accepted and the key k_{MH} between $\mathcal{M}$ and $\mathcal{H}$, as well as the key k_{FH} between $\mathcal{F}$ and $\mathcal{H}$ is identical on both sides.

Further, the accounting protocol is correct when definition 2 holds.

Definition 2 (Correctness WRA). In the presence of a passive adversary, Π is correct when $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$ have accepted and are sure that the partnered instances hold the same value B, containing the transmitted data volume.

2.6 Security Goals

Now we state the security goals that have to be achieved between the mobile device $\mathcal{M}$, the foreign network $\mathcal{F}$ and the home network $\mathcal{H}$. Between $\mathcal{M}$ and $\mathcal{H}$ mutual authentication, integrity and confidentiality is required. These goals can be obtained by using symmetric cryptographic methods based on key material which is agreed on both sides. Non-repudiation is not explicitly required, which leads to the fact that no asymmetric cryptography is necessary.

Between $\mathcal{F}$ and $\mathcal{H}$ mutual authentication is required for accounting. Both sides have to be sure about the identity of the other party, so that one side can account its provided service and the other side will accept the issued bill. Integrity protection and maybe confidentiality are necessary to protect the accounting data communicated between $\mathcal{F}$ and $\mathcal{H}$.

Definition 3 (Mutual Authentication between $\mathcal{M}$ and $\mathcal{H}$). $\mathcal{A}$ wins if one of the following arises during the protocol run:

1. An uncorrupted instance of $\mathcal{M}$ accepts with a corrupted partnered instance of $\mathcal{H}$
2. An uncorrupted instance of $\mathcal{H}$ accepts with a corrupted partnered instance of $\mathcal{M}$
3. After having accepted, both uncorrupted partnered instances $\mathcal{M}$ and $\mathcal{H}$ hold a different session key k_{MH}.

Definition 4 (Authenticated Key Exchange between $\mathcal{M}$ and $\mathcal{H}$). Given a uniformly chosen bit b, a PPT adversary $\mathcal{A}$ interacts with a correct protocol Π, whereby it is not allowed for $\mathcal{A}$ to query **RevealKey()** to an accepted instance or to corrupt an instance. $\mathrm{Game}_{\Pi}^{\mathrm{ake}-\mathcal{M}-\mathcal{H}}(\mathcal{A}, \mathrm{l})$ is defined as the following interaction:

1. $\mathcal{A}$ interacts with instances of $\mathcal{M}$, $\mathcal{F}$, $\mathcal{H}$ without using the **RevealKey()** and **Corrupt()** query
2. $\mathcal{A}$ asks **TestKey()** to an instance of $\mathcal{M}$ or $\mathcal{H}$ and gets, dependent on b, a random value chosen from $\{0{,}1\}^l$ (if $b = 0$) or k_{MH} (if $b = 1$)
3. After further interaction, $\mathcal{A}$ terminates and outputs a bit b'

$\mathcal{A}$ wins $\mathrm{Game}_{\Pi}^{\mathrm{ake}-\mathcal{M}-\mathcal{H}}(\mathcal{A}, \mathrm{l})$ if $b' = b$. The maximum probability of the adversarial advantage over the random guess of b, over all adversaries (running in time l) is

$$\mathrm{Adv}_{\Pi}^{\mathrm{ake}-\mathcal{M}-\mathcal{H}}(\mathcal{A}, l) = \overset{\max}{\mathcal{A}} \; |2\mathrm{Pr}[\mathrm{Game}_{\Pi}^{\mathrm{ake}-\mathcal{M}-\mathcal{H}}(\mathcal{A}, l) = b] - 1|.$$

Definition 5 (Mutual Authentication between $\mathcal{F}$ and $\mathcal{H}$). $\mathcal{A}$ wins if one of the following arises during the protocol run:

1. An uncorrupted instance of $\mathcal{F}$ accepts with a corrupted partnered instance of $\mathcal{H}$
2. An uncorrupted instance of $\mathcal{H}$ accepts with a corrupted partnered instance of $\mathcal{F}$

3. After having accepted, both uncorrupted partnered instances $\mathcal{F}$ and $\mathcal{H}$ hold a different session key k_{FH}

Definition 6 (Authenticated Key Exchange between $\mathcal{F}$ and $\mathcal{H}$). Given a uniformly chosen bit b, a PPT adversary $\mathcal{A}$ interacts with a correct protocol $\mathit{\Pi}$, whereby it is not allowed for $\mathcal{A}$ to query **RevealKey()** to an accepted instance or to corrupt an instance. $\mathrm{Game}_{\mathit{\Pi}}^{\mathrm{ake}-\mathcal{F}-\mathcal{H}}(\mathcal{A}, \mathrm{l})$ is defined as the following interaction:

1. $\mathcal{A}$ interacts with instances of $\mathcal{M}$, $\mathcal{F}$, $\mathcal{H}$ without using the **RevealKey()** and **Corrupt()** query
2. $\mathcal{A}$ asks **TestKey()** to an instance of $\mathcal{F}$ or $\mathcal{H}$ and gets, dependent on b, a random value chosen from $\{0{,}1\}^l$ (if $b = 0$) or k_{FH} (if $b = 1$)
3. After further interaction, $\mathcal{A}$ terminates and outputs a bit b'

$\mathcal{A}$ wins $\mathrm{Game}_{\mathit{\Pi}}^{\mathrm{ake}-\mathcal{F}-\mathcal{H}}(\mathcal{A}, \mathrm{l})$ if $b' = b$. The maximum probability of the adversarial advantage over the random guess of b, over all adversaries (running in time l) is

$$\mathrm{Adv}_{\mathit{\Pi}}^{\mathrm{ake}-\mathcal{F}-\mathcal{H}}(\mathcal{A}, l) = \overset{\max}{\mathcal{A}} \; |2\mathrm{Pr}[\mathrm{Game}_{\mathit{\Pi}}^{\mathrm{ake}-\mathcal{F}-\mathcal{H}}(\mathcal{A}, l) = b] - 1|.$$

Definition 7 (Anonymity of $\mathcal{M}$). This goal protects the anonymity of $\mathcal{M}$ by hiding the real identity of $\mathcal{M}$ towards $\mathcal{F}$ and all protocol outsiders. A PPT adversary $\mathcal{A}$ wins if one of the following occurs, after $\mathcal{M}$ and $\mathcal{H}$ have accepted:

1. $\mathcal{A}$ knows the real identity of $\mathcal{M}$
2. $\mathcal{A}$ knows if an instance of $\mathcal{M}$ has participated in a previous accepted session
3. $\mathcal{A}$ recognizes an instance of $\mathcal{M}$ when it participates in a next session

Definition 8 (Fair Accountability). In order to guarantee fair accountability, the foreign network $\mathcal{F}$ needs a non-repudiative acknowledgement over the size of the data, that was forwarded.

By demonstrating this acknowledgement, the foreign network $\mathcal{F}$ can prove, how much data was relayed (at least), whereby nor the mobile device $\mathcal{M}$ neither the home network $\mathcal{H}$ are able to deny this.
$\mathcal{A}$ wins if one of the following arises during the protocol run:

1. An uncorrupted instance of $\mathcal{F}$ or $\mathcal{M}$ accepts an acknowledgement over the transmitted bytes (COINS/SIG) that was not created by $\mathcal{H}$
2. An uncorrupted instance of $\mathcal{F}$ or $\mathcal{M}$ accepts an invalid or replayed acknowledgement over the transmitted bytes (COINS/SIG)

3 Protocol

3.1 Building Blocks

Now, we itemize the cryptographic primitves that are used by the proposed protocols EAWRT (Fig. 1) and WRA (Fig. 2).

- A *cryptographic hash function* that provides preimage, second preimage and collision resistance [10]. Hash: $\{0,1\}^* \rightarrow \{0,1\}^l$. By $\text{Succ}_{\text{Hash}}^{\text{preimage}}(l)$ we denote the success probability for a PPT adversary to find a preimage for a given output $\in \{0,1\}^l$ of the hash function. $\text{Succ}_{\text{Hash}}^{\text{2nd-preimage}}(l)$ denotes the success probability for a PPT adversary to find a second preimage $\in \{0,1\}^*$ for a given preimage-hash pair $\in \langle \{0,1\}^*, \{0,1\}^l \rangle$.
- A *message authentication code* (MAC) that suffices the weak unforgeability against chosen message attacks (WUF-CMA) [4]. $\text{Succ}_{\text{MAC}}^{\text{wuf-cma}}(l)$ denotes the success probability over all PPT adversaries to find a MAC forgery under access to the MAC oracle. A MAC is verified with $\text{ver}_{\text{key}}(\text{value})$.
- A *pseudo random function* PRF: $\{0,1\}^l \times \{0,1\}^* \rightarrow \{0,1\}^*$ for key derivation. We denote the maximum advantage over all PPT adversaries (running within time l) in distinguishing the outputs of PRF from the outputs of a random oracle better than $\text{Pr}=\frac{1}{2}$ by $\text{Adv}_{\text{PRF}}^{\text{prf}}(l)$.
- A *symmetric encryption scheme with integrity protection* that suffices the indistinguishability property under adaptive chosen ciphertext attacks (IND-CCA2) [2]. We denote the advantage that an adversary is able to decrypt (dec) at least one bit without knowing the used key as $\text{Adv}_{\text{DEC}}^{\text{ind-cca2}}(l)$. Furthermore, the symmetric encryption scheme satisfies weak unforgeability against chosen message attacks. The adversary's success probability to encrypt (enc) without the right key and gaining a valid ciphertext is $\text{Succ}_{\text{ENC}}^{\text{wuf-cma}}(l)$.
- A static *diffie-hellman key agreement* over a finite cyclic group, where the decisional diffie hellman (DDH) problem is strong. By $\text{Adv}_{\text{DH}}^{\text{ddh}}(l)$ we denote the advantage over all PPT adversaries to recognize a valid DH tuple.
- A *digital signature scheme* that provides existential unforgeability under chosen message attacks (EUF-CMA). The signing operation is denoted by $\text{sig}_{SK_?}$ and the according verification operation by $\text{ver}_{PK_?}$. The maximum success probability over all PPT adversaries of finding a forgery is represented by $\text{Succ}_{\text{SIG/VER}}^{\text{euf-cma}}(l)$.
- A set of *database operations*: **lookup(**AID_M**)** searches for the given index AID_M and returns the corresponding identity ($\mathcal{M}$). **add()** inserts a new assignment: $AID_M \rightarrow \mathcal{M}$.
- A set of *verification functions*: **validate** and **verify**. **validate** checks, if a value is within a logical range. The range may be of length one (an expected value). **verify** is used, when the expected value must be cryptographically computed, e.g. when the expected value must be hashed.

3.2 Roaming Protocol (EAWRT)

In the following, we propose a new protocol for the wireless roaming via tunnels scenario. We introduce a more efficient protocol than Manulis et al. by abandoning on digital signatures and asymmetric encryption. Due to this, we have smaller messages and we need less computation time. Additionally we support anonymity of the mobile device.

The EAWRT protocol is shown in Figure 1. $\mathcal{M}$, $\mathcal{F}$, $\mathcal{H}$ are the identities of the participants and AID_M is the anonymous identity of $\mathcal{M}$.

$SK_i = i$, $PK_i = g^i \mod p$ are the private respectively public diffie-hellman parameter for $i \in \{F, H\}$. In detail (but not shown in the figure), there is also a big prime p that conforms to the security level l and a base g that generates $\boldsymbol{Z}_p^*$.

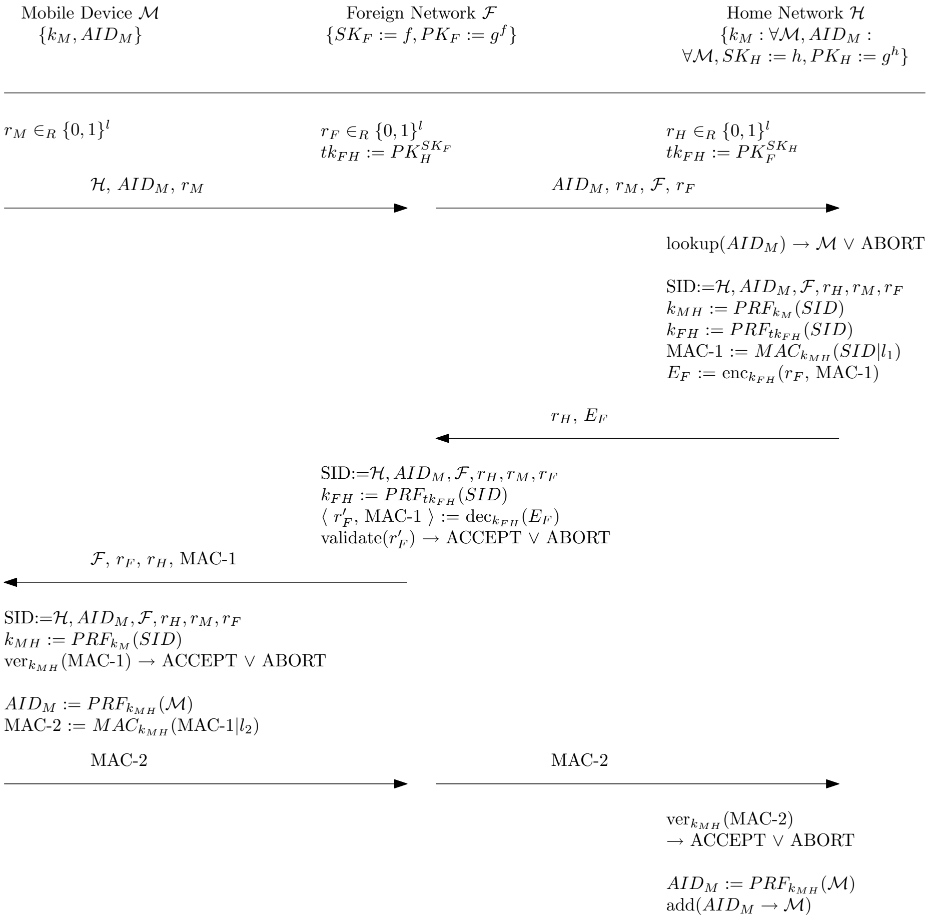
Mobile Device $\mathcal{M}$
$\{k_M, AID_M\}$
Foreign Network $\mathcal{F}$
$\{SK_F := f, PK_F := g^f\}$
Home Network $\mathcal{H}$
$\{k_M : \forall\mathcal{M}, AID_M : \forall\mathcal{M}, SK_H := h, PK_H := g^h\}$
$r_M \in_R \{0,1\}^l$
$r_F \in_R \{0,1\}^l$
$tk_{FH} := PK_H^{SK_F}$
$r_H \in_R \{0,1\}^l$
$tk_{FH} := PK_F^{SK_H}$
$\mathcal{H}$, AID_M, r_M
AID_M, r_M, $\mathcal{F}$, r_F
lookup(AID_M) → $\mathcal{M}$ ∨ ABORT
SID:=$\mathcal{H}, AID_M, \mathcal{F}, r_H, r_M, r_F$
$k_{MH} := PRF_{k_M}(SID)$
$k_{FH} := PRF_{tk_{FH}}(SID)$
MAC-1 := $MAC_{k_{MH}}(SID|l_1)$
$E_F :=$ enc$_{k_{FH}}$(r_F, MAC-1)
r_H, E_F
SID:=$\mathcal{H}, AID_M, \mathcal{F}, r_H, r_M, r_F$
$k_{FH} := PRF_{tk_{FH}}(SID)$
⟨ r'_F, MAC-1 ⟩ := dec$_{k_{FH}}$(E_F)
validate(r'_F) → ACCEPT ∨ ABORT
$\mathcal{F}$, r_F, r_H, MAC-1
SID:=$\mathcal{H}, AID_M, \mathcal{F}, r_H, r_M, r_F$
$k_{MH} := PRF_{k_M}(SID)$
ver$_{k_{MH}}$(MAC-1) → ACCEPT ∨ ABORT
$AID_M := PRF_{k_{MH}}(\mathcal{M})$
MAC-2 := $MAC_{k_{MH}}$(MAC-1$|l_2$)
MAC-2
MAC-2
ver$_{k_{MH}}$(MAC-2)
→ ACCEPT ∨ ABORT
$AID_M := PRF_{k_{MH}}(\mathcal{M})$
add($AID_M \to \mathcal{M}$)

Fig. 1. Efficient Authenticated Wireless Roaming via Tunnels (EAWRT)

Correctness of EAWRT. According to definition 1, EAWRT is correct, if all parties $\mathcal{M}$, $\mathcal{F}$, $\mathcal{H}$ have accepted and the keys k_{MH} and k_{FH} are identical on both sides.

k_{MH} is computed as $PRF_{k_M}(SID)$, whereby k_M is a shared key between $\mathcal{M}$, $\mathcal{H}$ and SID is the session identifier (consisting of all participant identifiers and all participant nonces). As proof statement we state that k_{MH} is identical on both sides, if both parties are partnered in the protocol session and share the same key k_M.

k_{FH} is computed as $PRF_{tk_{FH}}(SID)$, whereby tk_{FH} is a static Diffie-Hellman key between $\mathcal{F}$, $\mathcal{H}$ and SID is the session identifier. If both instances are partnered in the protocol session, the public key of the other party is known and $PK_H^{SK_F} \equiv PK_F^{SK_H}$, then k_{FH} is identical on both sides.

The combination of both statements gives an idea for the correctness proof of EAWRT.

Security of EAWRT. The security proof is given in appendix A.

3.3 Accounting Protocol (WRA)

We extended the model of Manulis et al. by the need for a fair accounting. To realize that, we propose the WRA protocol, which is an extension to the normal tunnel communication between the mobile device $\mathcal{M}$ and the home network $\mathcal{H}$. Additionally to the tunnel data, which is represented by MSG and MSG2, we have added some cryptographic measures to ensure that the foreign network $\mathcal{F}$ is able to proof, how many data was relayed. As consequence, the foreign network $\mathcal{F}$ is able to bring this size of transmitted data to account, whereby neither $\mathcal{H}$ nor $\mathcal{F}$ is able to cheat.

The home network $\mathcal{H}$ acknowledges the size of the transmitted data to F via two mechanisms. Firstly as an absolute value of the transmitted bytes in a digital signature. Secondly as a n element of a hash chain, representing a value relative to the last digitally signed value.

Figure 2 shows the WRA protocol. The size of the used hash chain is denoted by n. B is the number of transmitted bytes, whereby $Base$ is the last digitally signed value of B.

Correctness of WRA. We give an idea for the correctness proof of WRA in the following. If all parties have accepted, it is left to show that all parties have the knowledge of the same value B in the presence of a passive adversary. B can be represented by several values: B, COINS and SIG. $\mathcal{H}$ sends COINS with a corresponding MAC in the third message, $\mathcal{F}$ forwards these values in the fourth message. If these values have arrived at $\mathcal{M}$ and $\mathcal{M}$ accepts, it is obvious that all parties hold the same value for B.
The correctness of WRA can be proven with these considerations.

Security of WRA. The security proof is given in appendix A.

4 Efficiency Improvements

In comparison with the WRT protocol from Manulis et al. [7], we have some obvious advantages in respect to performance, since we abandon digital signatures and asymmetric encryption. Due to this, we have smaller sized messages and

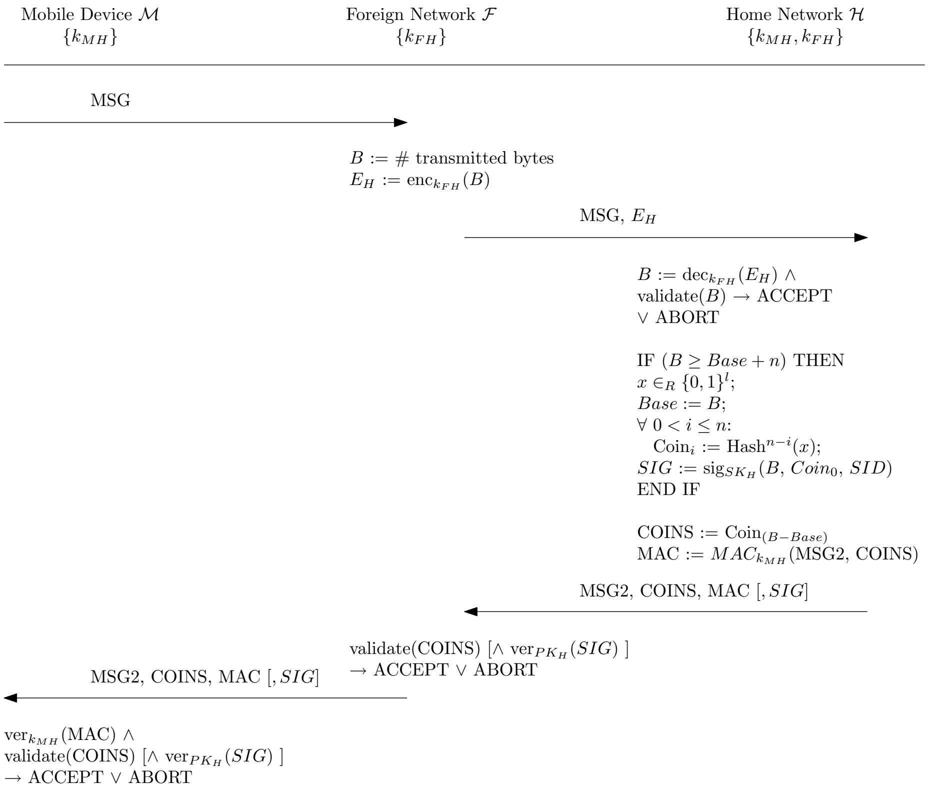
Mobile Device $\mathcal{M}$
$\{k_{MH}\}$
Foreign Network $\mathcal{F}$
$\{k_{FH}\}$
Home Network $\mathcal{H}$
$\{k_{MH}, k_{FH}\}$
MSG
B := # transmitted bytes
$E_H := \text{enc}_{k_{FH}}(B)$
MSG, E_H
$B := \text{dec}_{k_{FH}}(E_H) \wedge$
validate(B) → ACCEPT
∨ ABORT
IF ($B \geq Base + n$) THEN
$x \in_R \{0,1\}^l$;
$Base := B$;
$\forall\ 0 < i \leq n$:
Coin$_i$:= Hash$^{n-i}(x)$;
$SIG := \text{sig}_{SK_H}(B,\ Coin_0,\ SID)$
END IF
COINS := Coin$_{(B-Base)}$
MAC := $MAC_{k_{MH}}$(MSG2, COINS)
MSG2, COINS, MAC [$, SIG$]
validate(COINS) [∧ ver$_{PK_H}(SIG)$]
→ ACCEPT ∨ ABORT
MSG2, COINS, MAC [$, SIG$]
ver$_{k_{MH}}$(MAC) ∧
validate(COINS) [∧ ver$_{PK_H}(SIG)$]
→ ACCEPT ∨ ABORT

Fig. 2. Wireless Roaming Accounting protocol (WRA)

less computation time needed. Particulary for mobile devices this approach fits good, because their computation power respectively battery power is limited.

Moreover, we are able to improve the performance from EAWRT even more by applying some precomputations. The computation of tk_{FH}, the static diffie-hellman key, is computational expensive but has to be done only one time for all protocol instances with the same $\mathcal{F}$ and $\mathcal{H}$. So, this key can be computed at the first contact between $\mathcal{F}$ and $\mathcal{H}$ and then stored for later use.

After the last message of the EAWRT protocol, $\mathcal{H}$ verifies MAC-2 by comparison with a self-computed MAC-2. This computation can be done earlier to save time. The verification MAC-2 can be computed by $\mathcal{H}$ right after sending out his message $\langle r_H, E_F\rangle$, while waiting for the last message of the protocol.

5 Conclusion

In this paper, we introduced two new properties for the wireless roaming via tunnels scenario. At first, the anonymity property, which allows the user of the mobile device to stay anonymous for outsiders (including the foreign network) while roaming. This includes the unlinkability of two different sessions.

The second property is named fair accounting, which has a special meaning for this scenario. It is necessary for the foreign network, which forwards the tunnel data between the mobile device and the home network, that the home network approves the size of the transmitted data. Since the foreign network wants to get paid for relaying, the home network's confirmation of the size of the transmitted data must be non-repudiative, in other words: signed. In dispute, the foreign network can present the signatures and demand the payment.

We have presented an optimized AWRT protocol (named EAWRT), that fulfills the requirements propsed by Manulis et al. [7]. Additionally, our protocol has the anonymity property *and* is designed to be more efficient. Noteably the efficiency of our protocol is important, since we want to allow near realtime services like VoIP or video chats even in roaming cases.

Moreover we showed up a solution for the accounting problem by introducing another protocol named WRA. This protocol attaches some cryptographic values to the normal communication flow and can thereby enforce fair accounting without too much overhead.

For both introduced protocols there is a security proof in appendix A.

References

1. B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP). RFC 3748 (Proposed Standard), June 2004. Updated by RFC 5247.
2. Rackoff C. and Simon D. R. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In CRYPTO'91, LNCS 576, pp. 433-444, 1992.
3. Sha Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17:281–308, 1988.
4. Bellare M. and Namprempre C. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In ASIACRYPT'00, LNCS 1976, pp. 531-545, 2000.
5. Bellare M., Kilian J., and Rogaway P. Security of the cipher block chaining message authentication code. Journal of Computer and System Sciences, 61 (3), pp. 362-399., 2000.
6. Bellare M., Canetti R., and Krawczyk H. Keying hash functions for message authentication.
7. Mark Manulis, Damien Leroy, Francois Koeune, Olivier Bonaventure, and Jean-Jacques Quisquater. Authenticated wireless roaming via tunnels: Making mobile guests feel at home. Cryptology ePrint Archive, Report 2008/382, 2008. `http://eprint.iacr.org/`.
8. Mark Manulis, Ahmad-Reza Sadeghi, and Jörg Schwenk. Linkable democratic group signatures.
9. R. C. Merkle. A certified digital signature. Advances in Cryptology - CRYPTO'89, pp. 241-250., 1989.
10. Rogaway P. and Shrimpton T. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. FSE 2004, LNCS 3017, pp. 371-388, 2004.
11. David Pointcheval and Jacques Stern. Provably secure blind signature schemes. pages 252–265. Springer-Verlag, 1996.
12. N. Sastry, K.Sollins, and J. Crowcroft. Architecting citywide ubiquitous wi-fi access. HotNets-VI, 2007. `http://conferences.sigcomm.org/hotnets/2007/papers/hotnets6-final88.pdf`.
13. Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. `http://eprint.iacr.org/`.

A Security Analysis

The following security analysis is based on the sequences of games technique by Shoup [13].

Theorem 1 (Mutual authentication between $\mathcal{M}$ and $\mathcal{H}$) With a WUF-CMA secure MAC, the protocol Π of EAWRT provides mutual authentication in the sense of definition 3 and

$$\mathrm{Succ}_{\mathrm{EAWRT}}^{MA-\mathcal{M}-\mathcal{H}}(l) \leq \frac{3q^2}{2^l} + 2\mathrm{Succ}_{\mathrm{MAC}}^{\text{wuf-cma}}(l).$$

The event that an adversary $\mathcal{A}$ breaks the mutual authentication between $\mathcal{M}$ and $\mathcal{H}$ is denoted by $\mathrm{Win}_i^{MA-\mathcal{M}-\mathcal{H}}$.

Game G_0. [*Real protocol*] The real $\mathrm{Game}_{EAWRT}^{MA-\mathcal{M}-\mathcal{H}}(l)$ played between a PPT adversary $\mathcal{A}$ and a simulator Δ. Δ simulates all protocol queries from $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$ according to the protocol specification.

Game G_1. [*Collisions for nonces r_M, r_F and r_H*] The simulation aborts, when the same random nonces r_M, r_F or r_H are chosen by the simulator Δ in different protocol sessions.

$$|Pr[\mathrm{Win}_1^{MA-\mathcal{M}-\mathcal{H}}] - Pr[\mathrm{Win}_0^{MA-\mathcal{M}-\mathcal{H}}]| \leq \frac{3q^2}{2^l}$$

This game implies that the session identifier SID is unique for each session. It is needless to say that SID would stay unique as long as not all nonces show up collisions.

Game G_2. [*MAC forgeries for MAC-1 and MAC-2*] This Game differs from Game G_1 in the fact that the simulator Δ aborts, when the adversary $\mathcal{A}$ sends a message with a valid MAC, that was not previously computed by $\mathcal{M}$ or $\mathcal{H}$.

$$|Pr[\mathrm{Win}_2^{MA-\mathcal{M}-\mathcal{H}}] - Pr[\mathrm{Win}_1^{MA-\mathcal{M}-\mathcal{H}}]| \leq 2\mathrm{Succ}_{\mathrm{MAC}}^{\text{wuf-cma}}(l)$$

Since we can exclude MAC forgeries for MAC-1 and MAC-2, we can also exclude replay attacks for the values of MAC-1 and MAC-2. This is because the MACs are computed over the session identifier SID, which is unique for each session according game G_1. The mandatory verification of the MACs (if successful) leads to the fact, that both parties $\mathcal{M}$ and $\mathcal{H}$ share the same session identifier SID and are therefore partnered.

Furthermore we can conclude that both parties $\mathcal{M}$ and $\mathcal{H}$ have the same session key k_{MH}, because this key is necessary for the successful verification of MAC-1 and MAC-2.

Finally this game cannot be won by the win conditions 1, 2 and 3 from definition 3 in section 2.6. The probability to win game G_2 is therefore

$$Pr[\mathrm{Win}_2^{MA-\mathcal{M}-\mathcal{H}}] = 0.$$

Combining the previous equations, we conclude this proof.

Theorem 2 (Authenticated Key Exchange between $\mathcal{M}$ and $\mathcal{H}$) With a pseudo random function and a WUF-CMA secure MAC, the protocol Π of EAWRT provides authenticated key exchange in the sense of definition 4 and

$$\mathrm{Succ}_{\mathrm{EAWRT}}^{AKE-\mathcal{M}-\mathcal{H}}(l) \leq \frac{3q^2}{2^l} + 2\mathrm{Succ}_{\mathrm{MAC}}^{\text{wuf-cma}}(l) + 2q\mathrm{Adv}_{\mathrm{PRF}}^{\mathrm{prf}}(l).$$

The event that an adversary $\mathcal{A}$ breaks the mutual authentication between $\mathcal{M}$ and $\mathcal{H}$ is denoted by $\mathrm{Win}_i^{AKE-\mathcal{M}-\mathcal{H}}$.

Game G_0. [*Real protocol*] The real $\mathrm{Game}_{EAWRT}^{AKE-\mathcal{M}-\mathcal{H}}(l)$ played between a PPT adversary $\mathcal{A}$ and a simulator Δ. Δ simulates all protocol queries from $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$ according to the protocol specification. The adversary $\mathcal{A}$ may query **TestKey()** after an instance has accepted.

Game G_1. [*Collisions for nonces* r_M, r_F *and* r_H] The simulation aborts, when the same random nonces r_M, r_F or r_H are chosen by the simulator Δ in different protocol sessions.

$$|Pr[\mathrm{Win}_1^{AKE-\mathcal{M}-\mathcal{H}}] - Pr[\mathrm{Win}_0^{AKE-\mathcal{M}-\mathcal{H}}]| \leq \frac{3q^2}{2^l}$$

This game implies that the session identifier SID is unique for each session. It is needless to say that SID would stay unique as long as not all nonces show up collisions.

Game G_2. [*MAC forgeries for MAC-1 and MAC-2*] This Game differs from Game G_1 in the fact that the simulator Δ aborts, when the adversary $\mathcal{A}$ sends a message with a valid MAC, that was not previously computed by $\mathcal{M}$ or $\mathcal{H}$.

$$|Pr[\mathrm{Win}_2^{AKE-\mathcal{M}-\mathcal{H}}] - Pr[\mathrm{Win}_1^{AKE-\mathcal{M}-\mathcal{H}}]| \leq 2\mathrm{Succ}_{\mathrm{MAC}}^{\text{wuf-cma}}(l)$$

Since we can exclude MAC forgeries for MAC-1 and MAC-2 now, we can exclude replay attacks as well. We are sure, that both partnered instances have the same session identifier SID and use the same session key k_{MH}, since this key was used to create MAC-1 and MAC-2.

Game G_3. [*Pseudo-randomness of* k_{MH}] In this game, the simulator Δ chooses k_{MH} fully random in each session instead of computing it via a PRF. To conceive consistency, the same random value is chosen at the partnered instance.

$$|Pr[\mathrm{Win}_3^{AKE-\mathcal{M}-\mathcal{H}}] - Pr[\mathrm{Win}_2^{AKE-\mathcal{M}-\mathcal{H}}]| \leq q\mathrm{Adv}_{\mathrm{PRF}}^{\mathrm{prf}}(l)$$

Since in this game k_{MH} is not dependent on any known data, $\mathcal{A}$ queries testkey() and has to decide between two fully random values. Since that, $\mathcal{A}$ cannot make a better guess than

$$Pr[\mathrm{Win}_3^{AKE-\mathcal{M}-\mathcal{H}}] = \frac{1}{2}.$$

Combining the previous equations, we conclude this proof.

Theorem 3 (Mutual authentication between $\mathcal{F}$ and $\mathcal{H}$) With a WUF-CMA secure MAC, the protocol $\mathit{\Pi}$ of EAWRT provides mutual authentication in the sense of definition 5 and

$$\mathrm{Succ}_{\mathrm{EAWRT}}^{MA-\mathcal{F}-\mathcal{H}}(l) \leq \frac{3q^2}{2^l} + \mathrm{Succ}_{\mathrm{ENC}}^{\text{wuf-cma}}(l) + q\mathrm{Adv}_{\mathrm{DEC}}^{\text{ind-cca2}}(l).$$

The event that an adversary $\mathcal{A}$ breaks the mutual authentication between $\mathcal{F}$ and $\mathcal{H}$ is denoted by $\mathrm{Win}_i^{MA-\mathcal{F}-\mathcal{H}}$.

Game G_0. [*Real protocol*] The real $\mathrm{Game}_{EAWRT}^{MA-\mathcal{F}-\mathcal{H}}(l)$ played between a PPT adversary $\mathcal{A}$ and a simulator $\mathit{\Delta}$. $\mathit{\Delta}$ simulates all protocol queries from $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$ according to the protocol specification.

Game G_1. [*Collisions for nonces* r_M, r_F *and* r_H] The simulation aborts, when the same random nonces r_M, r_F or r_H are chosen by the simulator $\mathit{\Delta}$ in different protocol sessions.

$$|Pr[\mathrm{Win}_1^{MA-\mathcal{F}-\mathcal{H}}] - Pr[\mathrm{Win}_0^{MA-\mathcal{F}-\mathcal{H}}]| \leq \frac{3q^2}{2^l}$$

This game implies that the session identifier SID is unique for each session. It is needless to say that SID would stay unique as long as not all nonces show up collisions.

Game G_2. [*Encryption forgery for* E_F] This Game differs from Game G_1 in the fact that the simulator $\mathit{\Delta}$ aborts, when the adversary $\mathcal{A}$ sends a message with a valid encryption of r_F, that was not previously computed by $\mathcal{H}$.

$$|Pr[\mathrm{Win}_2^{MA-\mathcal{F}-\mathcal{H}}] - Pr[\mathrm{Win}_1^{MA-\mathcal{F}-\mathcal{H}}]| \leq \mathrm{Succ}_{\mathrm{ENC}}^{\text{wuf-cma}}(l)$$

We conclude that $\mathcal{H}$ is authenticated towards $\mathcal{F}$ with the used key k_{FH}.

Game G_3. [*Security* E_F] To proof the encryption strength of E_F, consider that $\mathit{\Delta}$ chooses a random bit b and encrypts a randomly chosen value (if $b = 0$) or MAC-1 (if $b = 1$). If $\mathcal{A}$ makes the right guess for b with a better probability than $\frac{1}{2}$, a distinguisher that breaks the IND-CCA2 security with the use of $\mathcal{A}$ can be created.

$$|Pr[\mathrm{Win}_3^{MA-\mathcal{F}-\mathcal{H}}] - Pr[\mathrm{Win}_2^{MA-\mathcal{F}-\mathcal{H}}]| \leq q\mathrm{Adv}_{\mathrm{DEC}}^{\text{ind-cca2}}(l)$$

Only if E_F is decrypted successfully by $\mathcal{F}$ and the valid MAC-1 is transmitted to $\mathcal{M}$, a valid MAC-2 can be computed by $\mathcal{M}$. Consequently, $\mathcal{F}$ is successfully authenticated towards $\mathcal{H}$ if a valid MAC-2 was received by $\mathcal{H}$, because MAC-1 must have been valid also. Moreover, k_{FH} must be identical on both sides, since the decryption/verification of E_F would fail with a different k_{FH}.

The probability that one of the partnered instances accepts with a wrong session key k_{FH} is therefore

$$Pr[\mathrm{Win}_3^{MA-\mathcal{F}-\mathcal{H}}] = 0.$$

Combining the previous equations, we conclude this proof.

Theorem 4 (Authenticated Key Exchange between $\mathcal{F}$ and $\mathcal{H}$) With a static Diffie-Hellman and a IND-CCA2 secure symmetric encryption, the protocol $\mathit{\Pi}$ of EAWRT provides authenticated key exchange in the sense of definition 6 and

$$\mathrm{Succ}_{\mathrm{EAWRT}}^{AKE-\mathcal{F}-\mathcal{H}}(l) \leq \frac{3q^2}{2^l} + q\mathrm{Adv}_{\mathrm{DH}}^{\mathrm{ddh}}(l) + q\mathrm{Adv}_{\mathrm{DEC}}^{\text{ind-cca2}}(l) + 2q\mathrm{Adv}_{\mathrm{PRF}}^{\mathrm{prf}}(l).$$

The event that an adversary $\mathcal{A}$ breaks the mutual authentication between $\mathcal{F}$ and $\mathcal{H}$ is denoted by $\mathrm{Win}_i^{AKE-\mathcal{F}-\mathcal{H}}$.

Game G_0. [*Real protocol*] The real $\mathrm{Game}_{EAWRT}^{AKE-\mathcal{F}-\mathcal{H}}(l)$ played between a PPT adversary $\mathcal{A}$ and a simulator $\mathit{\Delta}$. $\mathit{\Delta}$ simulates all protocol queries from $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$ according to the protocol specification. The adversary $\mathcal{A}$ may query **TestKey()** after an instance has accepted.

Game G_1. [*Collisions for nonces r_M, r_F and r_H*] The simulation aborts, when the same random nonces r_M, r_F or r_H are chosen by the simulator $\mathit{\Delta}$ in different protocol sessions.

$$|Pr[\mathrm{Win}_1^{AKE-\mathcal{F}-\mathcal{H}}] - Pr[\mathrm{Win}_0^{AKE-\mathcal{F}-\mathcal{H}}]| \leq \frac{3q^2}{2^l}$$

This game implies that the session identifier SID is unique for each session. It is needless to say that SID would stay unique as long as not all nonces show up collisions.

Game G_2. [*Secrecy of tk_{FH}*] In this game, the simulator $\mathit{\Delta}$ chooses tk_{FH} at random instead of computing it via static diffie-hellman key agreement. For consistency, tk_{FH} is replaced by the same random value at both partnered instances of $\mathcal{F}$ and $\mathcal{H}$.

The simulator $\mathit{\Delta}$ chooses a random value x and a random bit b. A distinguisher based on $\mathcal{A}$ can be created that decides if $[g, PK_F = g^f, PK_H = g^h, PRF, SID, \{k_{FH} = PRF_{g^x}(SID) \text{ (if } b = 0) \vee k_{FH} = PRF_{g^{fh}}(SID) \text{ (if } b = 1)\}]$ is a valid tuple. If the probability of the distinguisher is non-negligible higher than $\frac{1}{2}$, $\mathcal{A}$ can break the DDH-problem in this group.

$$|Pr[\mathrm{Win}_2^{AKE-\mathcal{F}-\mathcal{H}}] - Pr[\mathrm{Win}_1^{AKE-\mathcal{F}-\mathcal{H}}]| \leq q\mathrm{Adv}_{\mathrm{DH}}^{\mathrm{ddh}}(l)$$

Because the DDH-problem is (by definition in section 3.1) strong in this group, the adversary $\mathcal{A}$ is not able to gain any information about the common key tk_{FH}.

Game G_3. [*Security of E_F*] The simulator $\mathit{\Delta}$ chooses a random bit b. E_F will be encrypted with a random value if $b = 0$ and with k_{FH} if $b = 1$. A distinguisher that can make use of $\mathcal{A}$ decides whether E_F was encrypted with a random value or k_{FH}.

$$|Pr[\mathrm{Win}_3^{AKE-\mathcal{F}-\mathcal{H}}] - Pr[\mathrm{Win}_2^{AKE-\mathcal{F}-\mathcal{H}}]| \leq q\mathrm{Adv}_{\mathrm{DEC}}^{\text{ind-cca2}}(l)$$

This means that the encryption E_F leaks no information about k_{FH}, since the distinguisher cannot decide between k_{FH} and a random key with a probability higher than $\frac{1}{2} + q\mathrm{Adv}_{\mathrm{DEC}}^{\text{ind-cca2}}(l)$.

Game G_4. [*Pseudo-randomness of* k_{FH}] In this game, the simulator Δ chooses k_{FH} fully random in each session instead of computing it via a PRF over SID. To conceive consistency, the same random value is chosen at the partnered instance.

$$|Pr[\mathrm{Win}_4^{AKE-\mathcal{F}-\mathcal{H}}] - Pr[\mathrm{Win}_3^{AKE-\mathcal{F}-\mathcal{H}}]| \leq q\mathrm{Adv}_{\mathrm{PRF}}^{\mathrm{prf}}(l)$$

Since in this game k_{FH} is exchanged by a randomly chosen value, $\mathcal{A}$ is not able to win the **TestKey()**-game.

$$Pr[\mathrm{Win}_4^{AKE-\mathcal{F}-\mathcal{H}}] = \frac{1}{2}.$$

Combining the previous equations, we conclude this proof.

Theorem 5 (Anonymity of $\mathcal{M}$) With a pseudo random function PRF, the protocol Π of EAWRT provides anonymity of $\mathcal{M}$ in the sense of definition 7 and

$$\mathrm{Succ}_{\mathrm{EAWRT}}^{\mathrm{anonymity}}(l) \leq \frac{3q^2}{2^l} + q\mathrm{Adv}_{\mathrm{PRF}}^{\mathrm{prf}}(l).$$

The event that an adversary $\mathcal{A}$ breaks the anonymity of $\mathcal{M}$ is denoted by $\mathrm{Win}_i^{\mathrm{anonymity}}$.

Game G_0. [*Real protocol*] The real $\mathrm{Game}_{EAWRT}^{\mathrm{anonymity}}(l)$ played between a PPT adversary $\mathcal{A}$ and a simulator Δ. Δ simulates all protocol queries from $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$ according to the protocol specification.

Game G_1. [*Collisions for nonces* r_M*,* r_F *and* r_H] The simulation aborts, when the same random nonces r_M, r_F or r_H are chosen by the simulator Δ in different protocol sessions.

$$|Pr[\mathrm{Win}_1^{\mathrm{anonymity}}] - Pr[\mathrm{Win}_0^{\mathrm{anonymity}}]| \leq \frac{3q^2}{2^l}$$

This game implies that the session identifier SID and therefore the key k_{MH} is unique for each session (As long as not all nonces show up collisions). This again means that there is a distinct AID_M computed in each session, because AID_M is computed as $AID_M := PRF_{k_{MH}}(\mathcal{M})$.

This excepts the win conditions 2 and 3 from definition 7 (section 2.6), because the only identifier that is used by $\mathcal{M}$ is AID_M, which is different after each accepted session. Furthermore no other static element that could give a reference to $\mathcal{M}$ (there is only k_M left) is sent in the protocol, neither plain nor encrypted.

Game G_2. [*Pseudo-randomness of* AID_M] The simulator Δ chooses AID_M fully random instead of computing it via PRF. For consistency reasons, AID_M is chosen identically on both sides.

$$|Pr[\mathrm{Win}_2^{\mathrm{anonymity}}] - Pr[\mathrm{Win}_1^{\mathrm{anonymity}}]| \leq q\mathrm{Adv}_{\mathrm{PRF}}^{\mathrm{prf}}(l)$$

Since we have chosen AID_M fully random, no advice to the identity of $\mathcal{M}$ is present. The adversary is not able to win this game by win condition 1 from definition 7. After all, the adversary's probability to win the game by win condition 1, 2 or 3 is

$$Pr[\mathrm{Win}_2^{\mathrm{anonymity}}] = 0.$$

Combining the previous equations, we conclude this proof.

Theorem 6 (Fair Accountability of WRA) Given a EUF-CMA secure digital signature scheme and a cryptographic hash function, the fair accountability property of WRA (definition in section 2.6) can be broken with a probability of

$$Succ_{\mathrm{WRA}}^{\mathrm{FA}}(l) \leq \frac{1}{m} Succ_{\mathrm{SIG/VER}}^{\mathrm{euf\text{-}cma}}(l) + nSucc_{\mathrm{Hash}}^{\mathrm{preimage}}(l) + \mathrm{Succ}_{\mathrm{MAC}}^{\mathrm{wuf\text{-}cma}}(l).$$

The event that an adversary $\mathcal{A}$ breaks the fair accountability between $\mathcal{F}$ and $\mathcal{H}$ is denoted by Win_i^{FA}.

Game G_0. [*Real protocol*] The real $\mathrm{Game}_{WRA}^{FA}(l)$ played between a PPT adversary $\mathcal{A}$ and a simulator Δ. Δ simulates all protocol queries from $\mathcal{M}$, $\mathcal{F}$ and $\mathcal{H}$ according to the protocol specification.

To prove the fair accountability property, we have to show, that COINS as well as SIG cannot be forged. Since SIG only occurs, when the end of the hash chain is reached, we begin with that.

Game G_1. [*Forgery of SIG*] In this game, the simulator fails, if $\mathcal{A}$ sends a query containing a valid signature SIG that was not previously sent by $\mathcal{H}$. The appearance of SIG depends on the number of bytes that have to be acknowledged and the length of the hash chain n. We define the probability of appearance of SIG as $Pr[SIG \text{ occurs}] := \frac{1}{m}$, whereby $1 < m \leq n$.

$$|Pr[\mathrm{Win}_1^{FA}] - Pr[\mathrm{Win}_0^{FA}]| \leq \frac{1}{m} Succ_{\mathrm{SIG/VER}}^{\mathrm{euf\text{-}cma}}(l)$$

After this game we are sure that the adversary $\mathcal{A}$ cannot win by win condition 1 regarding SIG. Now we prove, that COINS cannot be forged, too. Lower values for COINS (i.e. Hash(COINS)) are detected instantly by the foreign network $\mathcal{F}$, since $\mathcal{F}$ expects a value that represents at least B, the number of transmitted bytes. It is open to show, that also $\mathcal{M}$ detects lower values for COINS and that an adversary, i.e. a malicious $\mathcal{F}$, is not able to create a value for COINS that represents a higher value. We begin with the second open problem.

Game G_2. [*Forgery of COINS to a higher value*] The simulator Δ fails, if the adversary $\mathcal{A}$ sends a query containing a valid preimage of COINS that was not formerly sent by $\mathcal{H}$.

$$|Pr[\mathrm{Win}_2^{FA}] - Pr[\mathrm{Win}_1^{FA}]| \leq nSucc_{\mathrm{Hash}}^{\mathrm{preimage}}(l)$$

The probability is n times as high as a normal preimage-attack on a cryptographic hash function, since a hash chain with length n is used. We are sure, COINS cannot be forged to a higher value and proceed to the first open problem. To enable the mobile device $\mathcal{M}$ to detect changes of COINS to lower values, a MAC was added. The MAC is computed over COINS *and* MSG2. This MAC has to be forged, if an andersary $\mathcal{A}$ wants to decrease the value of COINS (by hashing) for $\mathcal{M}$.

Game G_3. [*Forgery of MAC*] The simulator Δ fails, if the adversary $\mathcal{A}$ sends a query containing a valid MAC that was not sent by $\mathcal{H}$ before.

$$|Pr[\mathrm{Win}_3^{FA}] - Pr[\mathrm{Win}_2^{FA}]| \leq \mathrm{Succ}_{\mathrm{MAC}}^{\text{wuf-cma}}(l)$$

After having excluded forgeries of COINS to lower values for both relevant parties $\mathcal{M}$ and $\mathcal{F}$, the adversary $\mathcal{A}$ cannot win by win condition 1 any more. SIG, as well as COINS, can only be forged with a negligible probability.

Furthermore, invalid or replayed acknowledgements (SIG/COINS) are not possible neither. Invalid acknowledgements become obvious when validating SIG/ COINS, because $\mathcal{M}$ and $\mathcal{F}$ expect a certain value. In example, $\mathcal{F}$ would deny further collaboration with $\mathcal{H}$, if $\mathcal{H}$ would respond with a value for B that is lower than expected.

Replayed acknowledgements are not possible because of the following fact. SIG includes a signature over the fresh session ID (SID), which excludes deployment of SIG in parallel oder later sessions. If SIG is sent twice within one session, $\mathcal{M}$ and $\mathcal{F}$ recognize this because of the lower value for B.

COINS cannot be replayed to $\mathcal{F}$, since it represents the value B that was sent in the last message of $\mathcal{F}$ (freshness). Further, an adersary $\mathcal{A}$ cannot replay COINS to $\mathcal{M}$, because it is secured with a MAC over the fresh message MSG2. If an adversary $\mathcal{A}$ querys a message with a former value of COINS (and the corresponding MAC), $\mathcal{M}$ recognizes this because MAC would not match to the current MSG2.

As consequence, the adversary $\mathcal{A}$ cannot win by win condition 2.

$$Pr[\mathrm{Win}_3^{FA}] = 0.$$

Combining the previous equations, we conclude this proof.