P4

Publication P4

Jukka Ylitalo, Jan Melén, Pekka Nikander, and Vesa Torvinen, “Re-thinking Security in IP based Micro-Mobility”, in Proc. of the 7th International Conference on Information Security Conference (ISC’04) , pp. 318-329, Palo Alto, CA, USA, September 27-29, 2004, LNCS 3225, ISSN 0302-9743, ISBN 3-540-23208-7, Publisher: Springer (Eds: Kan Zhang, and Yuliang Zheng)

Re-thinking Security in IP based Micro-Mobility

Jukka Ylitalo, Jan Melén, Pekka Nikander, and Vesa Torvinen

Ericsson Research NomadicLab, 02420 Jorvas, Finland.
{first-name.surname}@ericsson.com

Abstract. Security problems in micro-mobility are mostly related to trust establishment between mobile nodes and middle-boxes, i.e. mobile anchor points. In this paper, we present a secure micro-mobility architecture that scales well between administrative domains, which are already using different kind of network access authentication techniques. The trust between the mobile nodes and middle boxes is established using one-way hash chains and a technique known as secret splitting. Our protocol protects the middle-boxes from traffic re-direction and related Denial-of-Service attacks. The hierarchical scheme supports signaling optimization and secure fast hand-offs. The implementation and simulation results are based on an enhanced version of Host Identity Protocol (HIP). To our knowledge, our micro-mobility protocol is the first one-and-half round-trip protocol that establishes simultaneously a trust relationship between a mobile node and an anchor point, and updates address bindings at the anchor point and at a peer node in a secure way.

1 Introduction

Authentication, Authorization and Accounting (AAA) and Public Key Infrastructures (PKI) can be used to establish a security association between a mobile node and middle-boxes. The operators must have cross signed certificates or common roaming agreements to support inter-domain hand-offs. However, once the mobile node changes a trust domain, AAA and PKI based systems have scalability problems that results in long hand-off times. To solve that problem, we present a secure micro-mobility architecture that can be used between different administrative domains.

Heterogenous networks and a multi-operator environment set limits to micro-mobility when a mobile node makes a hand-off between different operator networks. Current trust models are not designed to establish a trust relationship between mobile users and middle-boxes locating between two operator networks. In addition, the access networks may belong to small hot-spot providers without global roaming agreements. The hot-spot providers may want to offer signaling optimization for mobile users without having initial assurances of the users. All the issues are related to obtaining fast and secure key-sharing between mobile nodes and middle-boxes supporting signaling optimization.

In our approach, the trust between a mobile node and a middle-box is based on an initial trust relationship between the mobile node and its peer. In the basic case, the peer node may be a rendezvous server[1] that has initially shared a secret with the mobility node. The middle-box learns the identity and establishes a trust relationship with the mobile node during a macro-mobility exchange. The trust relationship is required to authenticate mobility management messages at the middle-box during micro-mobility. Our hierarchical micro-mobility scheme does not necessitate middle-boxes to access to public keys, or require them to perform computationally expensive operations.

The rest of this paper is organized as follows. Section 2 contains the problem statement. Section 3 defines the essential cryptographic techniques. Section 4 describes our mobility management protocol applying those techniques. In Section 5 we present simulation results and an instantiation of our architecture based on Host Identity Protocol (HIP). Finally, Section 6 concludes the paper.

2 Problem Statement

A middle-box supporting micro-mobility must be able to verify that address binding updates come from an authentic mobile node. However, the middle-box cannot verify the validity of the new locators without a secure binding between the IP addresses and a host. In other words, the middle-boxes and the peer nodes need evidence that an IP address belongs to the specific mobile node. Unverified address binding update messages open several security vulnerabilities. A malicious node can cause packets to be delivered to a wrong address. This can compromise secrecy and integrity of communication and cause DoS both at the communicating parties and at the address that receives the unwanted packets[1].

Mobile nodes, on the other hand, have to verify messages sent by middle-boxes to protect from Man-in-the-Middle (MitM) attackers. Mobile nodes trust anchor points, like in NAT devices, to translate the network addresses correctly. Moreover, the peer nodes cannot trust address update messages without making an end-to-end reachability test whenever a mobile node arrives to a new middle-box region.

Cellular IP[2], HAWAII [3], TIMIP[4] and HMIP[5] are examples of IP based micro-mobility schemes. They all use Mobile IP(v4)[6] / IPv6[7] for macro-mobility. Basically, the main security problems in all micro-mobility protocols are related to authenticating local address binding updates between mobile nodes and middle-boxes. Typically, security issues are mentioned only incidentally in different protocol proposals. Eardley et.al.[8] present an evaluation criteria framework for regional IP mobility protocols. However, their framework does not focus on security aspects in detail.

The Hierarchical Mobile IP (HMIP)[5] micro-mobility protocol is currently under development at IETF[9]. HMIP uses optionally IPSec to protect the local address binding updates. According to Soliman et.al. [9], the IPSec SAs can be created with any key establishment protocol, e.g., with IKE. However, in a typical case there is no initial trust relationship between the mobile node and

[1] E.g. a Mobile IP Home-Agent.

a middle-box, i.e., called Mobility Anchor Point (MAP)[2]. Thus, according to our understanding, the only realistic way to create the SAs is opportunistic authentication[3].

Many of the current problems with the micro-mobility protocols are related to making security and configuration efficient. For example, the mobile node must send twelve (12) messages when it changes securely the MAP region in HMIP using IKE. In addition, the operators have to make a careful analysis of the network topology, and configure router advertisement policies for the MAP devices. In the mobile node, the optimal selection between different MAPs requires some sophisticated algorithm.

Several security problems in micro-mobility are basically related to scalable key-sharing. Mink et. al[10] analyze different approaches to implement key management, in Mobile IP[6], between a mobile node and a middle-box in a foreign network. However, their Key Distribution Center (KDC) approach requires pre-configured security associations between networks. They have continued the work in [11] by presenting a Firewall-Aware Transparent Mobility Architecture (FATIMA).

3 Cryptographic Techniques

Middle-boxes supporting public key based authentication are typically vulnerable for CPU related Denial-of-Service attacks. In our approach, we use both Lamport *one-way hash chains* [12][13] and *secret splitting* techniques [14][15] to authenticate mobility management messages between mobile nodes and middle-boxes.

The protocol is based on an assumption that the end-points have established a mutual security association before the mobility exchange takes place. Once the mobile node wants to inform its peer about its new location, it constructs a hash chain, encrypts one of the successor hash values and sends it to the peer. As a result both end-points possess part of the same Lamport hash chain. The messages sent by both peers are protected with Hashed Message Authentication Codes (HMACs) using the hash values of the same hash chain. Basically, we protect HMACs with one-time passwords.

We apply a similar kind of authentication mechanism that is used in TESLA[16]. In our hierarchical micro-mobility model, there can be several middle-boxes on the communication path between the end-points. The middle-boxes support message buffering to implement delayed authentication for the the mobility management messages. In other words, a middle-box is able to verify a message only after it has received the successor message. Using the Lamport hash chains the middle-boxes are also able to verify that the messages belong together. A mobile node must always get a reply to its message, before sending the next message. This protects the hosts from MitM attacker trying to delay packets to learn

[2] Alternatively, the trust could be based on using the AAA infrastructure, as is planned to be done in HMIP.

[3] The nodes blindly trust each other during the initial key-exchange, e.g., like in SSH.

hash values. The basic idea of the delayed authentication is illustrated with the following protocol:

$A \Rightarrow$(Middle-box)$\Rightarrow B : IDs, Enc(H_{i+2}), H_i, HMAC(H_{i+1}, IDs||Enc(H_{i+2}))$
$A \Leftarrow$(Middle-box)$\Leftarrow B : IDs, H_{i+1}, HMAC(H_{i+2}, IDs)$
$A \Rightarrow$(Middle-box)$\Rightarrow B : IDs, H_{i+2}$

The middle-boxes learn the anchor hash value, H_i, during the end-to-end exchange. The authentication is based on an assumption that the middle-box does not need to care about the actual owner of the hash chain as long as the hash chain values are valid during the communication context lifetime (see Section 2). Thus, the only problem related to initial bootstrapping is that an attacker can establish a state using own hash chain with a spoofed identifier, e.g., a home-address of the mobile node. This results into a situation where the authentic mobile node cannot create a state with the middle-box using its identifier. The problem can be solved by hashing the identifier with a random number. The mobile node sends its identifier and the random number to the peer. If a middle-box has already a context for the specific hash, the mobile node just generates a new random number and restarts the exchange with a new hash.

Another design issue is related to hash chain bootstrapping. Basically, the bootstrapping message can be authenticated using public key cryptography. In our approach, the peers do not authenticate the bootstrapping message with signatures, but they link together two independently created one-way hash chains with HMAC computation. A value of the first one-way hash chain is used to authenticate the anchor value of the new chain. The old anchor value is replaced with the new anchor value after the exchange.

$A \Rightarrow$(Middle-box)$\Rightarrow B : H_0^{new}, H_i^{old}, HMAC(H_{i+1}^{old}, H_0^{new}))$
$A \Leftarrow$(Middle-box)$\Leftarrow B : H_{i+1}^{old}$

4 Mobility Management Protocol

In our context, a logical *end-point* is a participant in an end-to-end communication[17]. Each end-point is identified with a global *End-point Identifier (EID).* A location name, i.e. *a locator*, defines the topological point-of-attachment of an end-point in the network. When the locators are separated from end-point identifiers, an end-point may change its location without breaking the transport layer connection. The binding, between EIDs and locators, may be simultaneously dynamic and one-to-many, providing mobility and multi-homing, respectively [6][7][18].

When we separate the location names from the end-point identifiers, we obtain a new name space that can be used as static identifiers. An EID multiplexed NAT device associates a connection state with the EIDs. It is able to multiplex several connections on a single IP address based on the end-point identifiers.

$$\text{REA: } \underbrace{\text{EIDs}, \text{SHA1}(K_{1xor2}), \text{Enc}_{K_{e2e}}(K_2 \| H_{i+2}), \text{Enc}_{K^{old}_{1xor2}}(K_2 \| H_{i+2}), H_i, [H^{new}_0]}_{\text{HMAC}(H_{i+1}, \text{SHA1}(...))}, K_1, \text{SPIs}$$

$$\text{AC: } \text{EIDs}, K_2, H_{i+1}, \text{SPIs}, \text{HMAC}(H_{i+2}, \text{SHA1}(\text{EIDs} \| K_2 \| H_{i+1}))$$

$$\text{ACR: } \text{EIDs}, H_{i+2}, \text{HMAC}(K_{1xor2}, \text{EIDs} \| H_{i+2}), \text{HMAC}(K_{e2e}, \text{EIDs} \| H_{i+2})$$

Fig. 1. Micro and macro-mobility exchanges use common packet structures. $K_1 = 1^{st}$ key piece, $K_2 = 2^{nd}$ key piece, $K_{1xor2} = K_1 \oplus K_2, K^{old}_{1xor2} = K^{old}_1 \oplus K^{old}_2, K_{e2e} =$ shared end-to-end key,$Enc_K =$ encrypted with K,$H_k = k^{th}$ hash value.

This makes network address translation similar to routing, since now IP address translation can be logically based on the static end-point identifiers. An EID multiplexed NAT device supporting dynamic address bindings is called a *regional anchor point*. An anchor point maintains a state for each mobile node inside its region.

When a mobile node arrives to a new anchor point region and the anchor point does not have a state for the mobile node, it forwards mobility management messages to the peer. During this macro-mobility exchange the anchor points in the hierarchy learns the anchor hash value. In addition, the nearest anchor point in the hierarchy learns a symmetric key. It is the only anchor point that knows the symmetric key and can reply to messages sent by the mobile node during micro-mobility. Other anchor points in the hierarchy authenticate the address binding update messages with the hash chain values before forwarding packets to the peer. After the initial macro-mobility exchange, the anchor point hides mobility signaling from the peer node. The micro-mobility approach is transparent to the mobile nodes.

The trust model between the mobile node and the anchor points is based on a flow of trust. The flow start from the initial security association (SA) established between the mobile node and the other end-point (e.g. a rendezvous server). During the macro-mobility exchange the other end-point plays the peer role, but when the micro-mobility takes place the old anchor point, knowing the shared secret, becomes the peer. Basically, the mobile node trusts its peer to faithfully decrypt and reply to all packets sent to it.

Both the micro- and macro-mobility exchange use similar kind of three-way handshake. The protocol consists of re-address (REA), address check (AC) and address check reply (ACR) packets (See Figure 1). The mobile node informs its peer about its IP addresses using the REA packet. The peer responds with AC packet, verifying that the mobile node is indeed at the claimed location. The ACR message contains the answer to the challenge. The purpose of the AC/ACR message pair is to prevent legitimate mobile nodes from inducing flooding attacks [1]. [4]

[4] It corresponds the Mobile IPv6 Return Routability (RR) test[7].

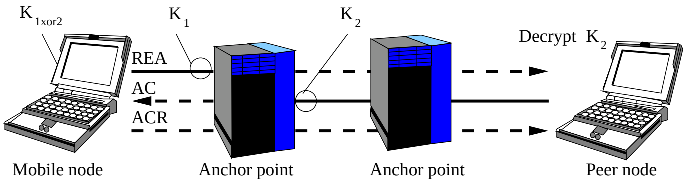

Fig. 2. Using secret splitting to share a key between a mobile node and an anchor point.

4.1 Message authentication at an anchor point

The mobile node generates a hash chain containing n items to be used in several subsequent protocol runs. During the initial macro-mobility exchange the mobile node bootstraps a hash chains by revealing H_i (i=0) in the REA message. Each exchange consumes three hash values, and each hand-off inside an anchor region triggers a new exchange. The first micro-mobility REA message reveals the hash value, H_{i+3}, and so on. The hash chain binds the subsequent re-addressing exchanges to each other. Finally, when the mobile node is running out of hash items (n = 3) it must bootstrap a new hash chain by revealing a new anchor value H_0^{new}. The bootstrapping and authentication follows the procedures presented in Section 3. An anchor point must always forward a packet containing a new anchor value to the peer node. In other words, each bootstrapping results in a macro-mobility exchange. In this way, the other anchor points in the hierarchy keep in synchrony with the hash chains. During the bootstrapping procedure the anchor points between the peers must verify that H_{i+k} in the received REA is a successor value of already known anchor value H_i.

As described in Section 3, the hash value, H_{i+2}, is encrypted with a symmetric key in the REA message. The protocol uses the existing security association between the peers to encrypt the value. Once the peer receives the REA packet it decrypts the H_{i+2} value and constructs the same hash chain with the mobile node. As a result, both peers know the values of the same hash chain. Each regional anchor point in the hierarchy verifies the first message after the second packet arrives, and the second packet after the third packet arrives. If a MitM sends a spoofed AC message, the mobile node just drops the message based on the invalid HMAC and resends a new REA packet. On the other hand, the anchor points drop ACR packets with incorrect H_{i+2} values, until they receive a valid one. The REA packets are protected from reply attacks with an increasing message counter.

4.2 Sharing a key between a mobile node and an anchor point

The hash chains are used to update address binding and to bootstrap hash chains at anchor points in the hierarchy. The key splitting technique [14][15], in turn, is used to protect the communication between the mobile and the nearest anchor point in which region the mobile node is. A shared secret, K_{1xor2}, is divided into two pieces, K_1 and K_2. [5] The REA packet contains a plaintext key piece and an encrypted piece (Figure 1). The nearest anchor point zeros the plain key piece, K_1, before forwarding the packet to the peer (Figure 2). The HMAC cannot be computed over the K_1, because otherwise the peer could not verify the REA message. Instead, a hash of the full shared key, $SHA1(K_{1xor2})$, is covered by the HMAC.

Once the peer receives a re-addressing packet it decrypts the second key piece, K_2, and sends it back as plaintext in the reply message. The nearest anchor point to the mobile mobile validates the second key piece using the hash of the shared key. Before forwarding the AC reply packet to the mobile node, the anchor point zeros also the second key piece. As a result, it remains the only Man-in-the-Middle knowing the both key pieces. After the initial macro-mobility exchange, the anchor point runs the reachability test by replying to incoming REA messages with AC message. Both the challenge (AC) and challenge reply (ACR) are authenticated with hash chain values known only by the mobile node and the anchor point.

The protocol is vulnerable for two kind of MitM attacks. In the first case, the MitM pretends to be the nearest anchor point towards the mobile node. The operator may moderate the attack by tagging its leaf anchor points. If the authentic leaf anchor point does not find the K_1 value in the packet it knows somebody is pretending to be the leaf anchor point. In such a case, the authentic leaf anchor point must zero the K_2 when it arrives in the AC message. As result, the attacker will not find the K_{1xor2}, but each following hand-off results in macro-mobility exchange. In the other attack, two MitM attackers must locate on both sides of the anchor point to learn the key pieces and replace the packets containing the hash chain values to successfully implement an attack. On the other hand, the current Mobile IPv6 Return Routability (RR) test is also vulnerable for this kind of attack.

Each subsequent micro-mobility exchange updates the security association between the mobile node and the anchor point. The old shared key, K_{1xor2}^{old}, is replaced by the new K_{1xor2}. The anchor point uses the old shared key to decrypt the second key piece of the new shared key. This is an important property during the regional hand-off (Section 4.3). The anchor point uses the H_{i+2} hash value instead of K_{1xor2} to protect the AC message. The reason for this is that an anchor point knowing the shared secret can play the peer role in the hierarchical model during hand-off. The other anchor points on the path are able to authenticate messages as described in the next Section.

4.3 Micro-mobility management inside hierarchy

Basically, it is easier to implement security in micro-mobility using soft hand-offs than hard hand-offs. However, several wireless technologies do not currently support soft hand-offs. We have focused on solving fast hand-off problems related

[5] $\exists(K_1, K_{1xor2} \in nonce); (K_2 = K_{1xor2} \oplus K_1) \Rightarrow (K_{1xor2} = K_1 \oplus K_2)$

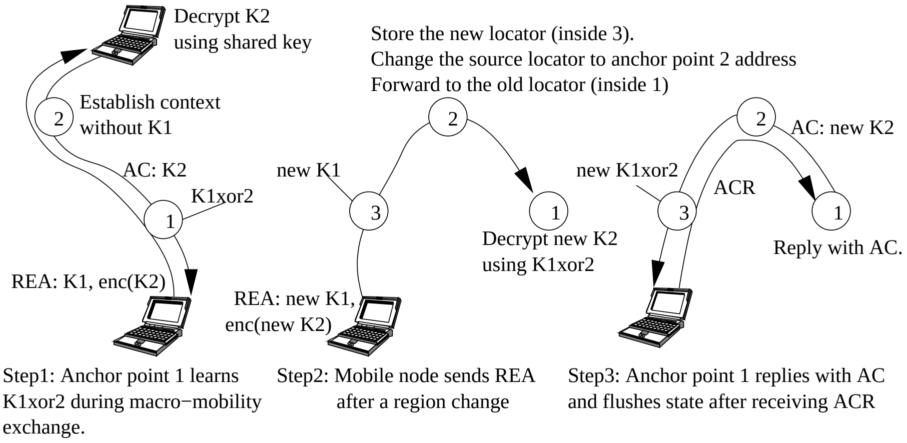
Decrypt K2
using shared key
Establish context
without K1
2
AC: K2
K1xor2
1
REA: K1, enc(K2)
Step1: Anchor point 1 learns
K1xor2 during macro−mobility
exchange.
Store the new locator (inside 3).
Change the source locator to anchor point 2 address
Forward to the old locator (inside 1)
2
new K1
3
1
Decrypt new K2
using K1xor2
REA: new K1,
enc(new K2)
Step2: Mobile node sends REA
after a region change
2
AC: new K2
new K1xor2
ACR
3
1
Reply with AC.
Step3: Anchor point 1 replies with AC
and flushes state after receiving ACR

Fig. 3. A mobile node attaches to a network inside the anchor point 1 region. Later, it makes a regional hand-off and moves to anchor point 3 region.

to hard hand-offs between anchor point regions. Different micro-mobility hand-off schemes are analyzed, e.g., by Ghassemian and Aghvami[19].

The presented three-way handshake protocol (Figure 1) supports deep anchor point hierarchies. In our approach the anchor points locate on the communication path, like NAT devices. As a result, the micro-mobility communication is transparent to the mobile node. Figure 3 illustrates a situation when a mobile node moves inside a two-level anchor point hierarchy. We assume that the mobile node has earlier established a security association with the other end-point. If an anchor point does not have a context for the mobile node it forwards the packets to the peer. During the macro-mobility exchange the anchor points 1 and 2 establish a context, and learn the current location of the mobile node. The nearest anchor point 1, in the hierarchy, learns also the shared key, K_{1xor2}. Later, the mobile node changes to the anchor point 3 region.

The hand-off triggers a new re-addressing exchange. As a result, the mobile node generates fresh key pieces and sends a REA packet to the other end-point. The anchor point 3 learns the first key piece in the normal way. Once the root [6] anchor point 2 receives the REA packet, it verifies that the hash value is a successor value of the earlier received anchor value and updates the context with the new hash value. Furthermore, it forwards the REA packet back to the mobile node's old location, stored in the context. However, before forwarding the packet, the anchor point 2 changes the source IP address with its own. Otherwise, the anchor point 1 might route the packet directly to mobile node, instead of routing it via the anchor point 2. In this way, the anchor point may verify the REA and AC messages, and update its context.

[6] The first anchor point on the path that has a context for the mobile node, but does not know the shared key works as a root during the region change.

After receiving the REA message, the anchor point 1 decrypts the new key piece, K_2, using the earlier learned shared key, K_{1xor2}, and includes the decrypted key piece to the AC reply message. The anchor point 2 verifies the REA message, and forwards the AC packet back to the mobile node's new location. The anchor point 3 learns the new shared secret, known only by it and the mobile node. The mobile node replies with ACR message that is routed again via anchor point 2 to the anchor point 1. Each anchor point verifies the ACR and updates its context with the mobile node's new location. The anchor point 1 also flushes its state. It is good to notice that every re-addressing exchange initializes a new shared secret. Thus, the old anchor point 1 cannot send spoofed messages after the mobile node has established a context with the anchor point 3.

5 Implementation

Our implementation is based on the Host Identity Protocol (HIP)[20][18]. HIP separates end-point identifiers from locators by defining a new cryptographical name space. The translation between the EIDs and locators happens at a logical layer between transport and networking layers. HIP consists of a base exchange and a re-addressing exchange. The base exchange is basically a two-round-trip end-to-end authenticated Diffie-Hellman key exchange protocol. We have replaced the original re-addressing exchange with the new version, presented in this paper, and implemented the regional anchor point functionality. The implementation is based on the FreeBSD 5.2 operating system.

The end-point identifiers are not present in the regular traffic between the hosts. However, each packet must logically include both the end-point identifiers and IP addresses of the sender and recipient. The IPSec Security Parameter Index (SPI) values, together with IP addresses, can be used as *indices for end-point identifiers*, resulting in packets that are syntactically similar to those used today [18]. A regional anchor point learns the SPIs together with the EIDs during re-addressing exchanges. The anchor point uses the SPIs to properly demultiplex any packets arriving to a shared IP address, i.e., implementing SPI multiplexed NAT (SPINAT) (see anchor point definition, Section 4). Basically, SPINAT works in the same way as port multiplexed NAT (NAPT) [21], but with SPI values. This means that the SPI values in the exchanges cannot be encrypted or included into signatures. The security properties of SPINAT are discussed in more detail in [22].

5.1 Simulation results

In our simulation, a mobile node negotiated the HIP end-to-end base exchange once with a web-server using IPv4. When it arrived to a new anchor point region it negotiated macro-mobility exchange with a web-server via an anchor point. [7] After a successful exchange, the mobile node moved inside the anchor point region sustaining the connection.

[7] The mobile node had 1.4 GHz Mobile Pentium processor, while the anchor point and the web-server had 2 GHz Pentium 4 processors.

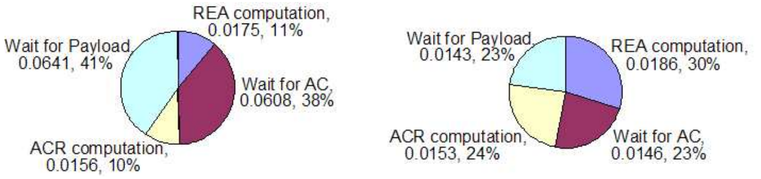

Fig. 4. Left: Macro-mobility exchange between the mobile node (MN) and the webserver in seconds (total 0.162 sec). Right: Micro-mobility exchange between the MN and the anchor point in seconds (total 0.062 sec). From MN viewpoint.

We ran our simulation 100 times to get statistically valid results. We simulated the latency[8] when the mobile node and anchor point located in Finland (nomadiclab.com) and communicated with a web-server at White House (www.whitehouse.gov). The mobile node was attached to the local network using 802.11b. The average Round-Trip-Time (RTT) between mobile node and the anchor point was 2.4ms. The simulated end-to-end RTT over the Internet was 55ms.

Figure 4 illustrates the hand-off times during macro-mobility and when a mobile node moves inside an anchor point region. The first slice in both figures describes the total computation time of a REA packet. The lower layer delays, caused by e.g. 802.1x authentication, are not included in the figures. They are considered to be the same in both cases. The second slice illustrates the network latency and AC computation time at the peer node. The third slice contains the ACR processing time. The exchange ends when the mobile node receives the first payload packet from the server.

Figure 5 compares the micro and macro-mobility hand-off times. When a mobile node changes an anchor point region it negotiates macro-mobility exchange once with its peer. The micro-mobility scheme does not increase the amount of signaling nor computation time compared to macro-mobility. The actual packet processing times are similar in both cases. However, the network latency causes most of the delay in the macro-mobility. In our case, the micro-mobility scheme allows 2.6 times faster hand-offs (micro-mobility 62ms vs. macro-mobility 162ms). The obtained benefit depends directly on the total network latency between mobile node and the server, because the anchor point does not cause extra signaling.

A good reference protocol is IKE with OAKLEY Quick-mode using 1536 MODP group that is run after HMIP MAP region exchange (Section 2). The exchange took over 1 second to complete between the mobile node and the anchor point using our IKE installation.

[8] Freebsd 5.2 ipfw property.

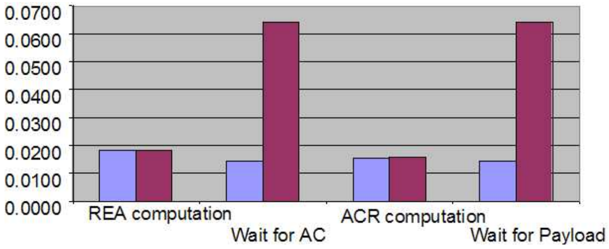

Fig. 5. Micro (left) and macro-mobility (right) hand-offs. Delay in seconds.

6 Conclusion

In this paper, we have presented a three-way handshake that is used both for micro and macro-mobility to update address bindings. The trust between the mobile node and the anchor point is established applying Lamport hash chains with delayed authentication and secret splitting techniques. The key-sharing model scales well between administrative domains and makes possible to implement fast hand-offs between anchor point regions.

The presented protocol is vulnerable for certain Man-in-the-middle attacks. While the security provided by our protocol is relatively low, it is sufficient to prevent the new attacks enabled by the addition of micro-mobility to the Internet mobility protocols. We believe that a scalable reachability test in micro-mobility may turn out be as important thing as the return routability protocol has been for macro-mobility protocols.

The presented simulation results show that it is possible to build simultaneously secure and fast micro-mobility architecture. The architecture does not require an anchor point discovery protocol, which makes network configuration easy. In addition, the mobile node does not need to make complex routing desicions. As a result, the presented secure micro-mobility architecture is easy to deploy.

Acknowledgments

The authors would like to thank Tuomas Aura, Mats Naslund, Goran Schultz, Tommi Linnakangas, Miika Komu, Mika Kousa and Kristian Slavov for their valuable comments.

References

1. Aura, T., Roe, M., Arkko, J.: Security of Internet Location Management. In Proc. Asia-Pacific Computer Systems Architecture Conference, ACSAC'02, Monash University, Melbourne, Australia. Feb. 2002.

2. Campbell, A., Gomez, J., Kim, S., Valko, A., Wan, C., Turanyi, Z.: Design, implementation, and evaluation of Cellular IP. IEEE Personal Commun. Mag., vol. 7, no. 4. Aug. 2000.
3. Ramjee, R., Porta, T., Salgarelli, L., Thuel, S., Varadhan, K.: IP-based Access Network Infrastructure for next Generation Wireless Data Networks. IEEE Personal Commun. Mag., vol. 7, no.4. Aug. 2000.
4. Grilo, A., Estrela, P., Nunes, M.: Terminal Independent Mobility for IP (TIMIP). IEEE Commun. Mag. Dec. 2001.
5. Castelluccia, C.: HMIPv6: A Hierarchical Mobile IPv6 Proposal. ACM Mobile Computing and Communication Review (MC2R). Apr. 2000.
6. Perkins, C.: IP Mobility Support. RFC 2002. 1996.
7. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. RFC 3775. 2004.
8. Eardley, P., Mihailovic, M., Suihko, T.: A Framework for the Evaluation of IP Mobility Protocols. In Proc. the 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'00). Sept. 2000.
9. Soliman, H., Castelluccia, C., El-Malki, K., Bellier, L.: Hierarchical Mobile IPv6 mobility management (HMIPv6). Internet Draft, work in progress. June 2004.
10. Mink, S., Pahlke, F., Schafer, G., Schiller, J.: Towards Secure Mobility Support for IP Networks. In Proc. IFIP International Conference on Communication Technologies (ICCT). Aug. 2000.
11. Mink, S., et.al.: FATIMA: A Firewall-Aware Transparent Internet Mobility Architecture. In Proc. the 5th IEEE Symposium on Computers and Communications (ISCC). July 2000.
12. Lamport, L.: Password authentication with insecure communication. Commun. Mag. of ACM, 24 (11), pp. 770-772. 1981.
13. Hu, Y.-C., Perrig, A., Johnson, D.: Efficient Security Mechanism for Routing Protocols. In Proc. Network and Distributed Systems Security Symposium (NDSS'03). Feb. 2003.
14. Shamir, A.: How to Share a Secret. Comm. of the ACM, 22(11):612- 613. Nov. 1979.
15. Blakely, G: Safeguarding Cryptographic Keys. In Proc. AFIPS National Computer Conference. pp. 313-317. 1979.
16. Perrig, A., Canetti, R., Tygar, J.D., Song, D.: Efficient Authentication and Signing of Multicast Streams over Lossy Channels. In Proc. IEEE Security and Privacy Symposium SP2000. May 2000.
17. Saltzer, J., Reed, D., Clark, D.: End-To-End Arguments in System Design. ACM Transactions on Computer Systems, vol. 2. Nov. 1984.
18. Nikander, P., Ylitalo, J., Wall, J.: Integrating Security, Mobility, and Multi-Homing in a HIP Way. In Proc. Network and Distributed Systems Security Symposium (NDSS'03). Feb. 2003.
19. Ghassemian, M., Aghvami, A.: Comparing different handoff schemes in IP based Micro-Mobility Protocols. In Proc. IST2002. Nov. 2002.
20. Moskowitz, R., Nikander, P., Jokela, P., Henderson, T.: Host Identity Protocol. Internet Draft, work in progress. June 2004.
21. Srisuresh, P. Holdrege, M.: IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663. 1999.
22. Ylitalo, J., Nikander, P.: BLIND: A Complete Identity Protection Framework for End-points. In Proc. the Twelfth International Workshop on Security Protocols. Apr. 2004.