WEAK CONTEXT ESTABLISHMENT PROCEDURE FOR MOBILITY AND MULTI-HOMING MANAGEMENT

Vesa Torvinen and Jukka Ylitalo
Ericsson Research, NomadicLab, Finland

Abstract: Trust establishment seems to be the most difficult problem in mobility and multi-homing management. Many protocol proposals assume the presence of some security infrastructure (e.g. a Public-Key Infrastructure). However, building such a global infrastructure has not taken place, maybe because it would be too expensive and difficult to deploy. In this paper, we introduce a security context establishment procedure that utilizes reverse hash chains, and does not require pre-existing security information. The procedure is known to be vulnerable to an active Man-in-the-Middle attack in the first message exchange, however, the procedure is efficient, and does not have inherent scalability problems.

Key words: security, mobility management, multi-homing management, and trust establishment

1. INTRODUCTION

Within the last couple of years, we have witnessed a lack of security awareness in many protocol design proposals. Even though many designers acknowledge the importance of considering security aspects right from the beginning, security is still far too often seen as an add-on, rather than an inherent part of the design process. The reason for the current situation is probably related to the complexity of current telecommunication and security protocols. Also, protocol designers are typically strongly discouraged from making their own security designs, which may alienate the designers from considering security related issues.

The situation is not much better from the security community point of view. There has been a lack of resources for doing security analysis in different application contexts. Furthermore, security requirements themselves may not have been realistic from a deployment point of view. Even though we may have "bullet-proof" security protocols, they may not be widely deployed.

In this paper, we study mobility and multi-homing management problems from a security point of view. We understand mobility management as a procedure in which the locator of an entity changes over time [cf. 21, 10]. Mobility mechanisms allow mobile nodes to remain reachable while moving around in the network. We assume that a mobile node changes its IP address every time it moves to a new link. Location changes are challenging especially for transport and higher-layer connections that should be maintained while moving around the network. Also, the protocol design should be resistant to various attacks, such as Denial-of-Service and re-direction attacks.

Multi-homing, on the other hand, comes very close to the mobility management problem. In this case, the node has several alternative access paths valid at the same time. Two entities may want to communicate via parallel paths at the same time especially if access paths are good for different types of traffic [cf. 1]. In multi-homing, the change of locators may be slower than in the mobility case (e.g. multi-homing may require re-numbering a site's address space), however, the problem of changing locators over time remains the same.

From a security point of view, we further develop the idea of "weak" security. Our goal is to develop a weak context establishment and update procedure that is reasonably secure against MitM, DoS and re-direction attacks, and that is not based on the use of public key cryptography. The procedure should be usable for mobility management and multi-homing. We also assume that the procedure does not need to take care of traffic confidentiality protection because there are other usable upper layer protocols available for this purpose.

The rest of the paper is organized as follows. The next section goes deeper into the security problems in the mobility and multi-homing context. The third section introduces the theoretical background to the security mechanisms we intend to use, i.e. reverse hash chains. Our generalized solution is presented in the fourth section, followed by a section utilizing the framework for multi-homing and local mobility management problems. Finally, we draw some conclusions based on our experience.

2. BACKGROUND

IP-based mobility, in which IP addresses are frequently changed, is challenging from an efficiency point of view. Each new network connection requires lots of processing, and message exchanges, e.g. network discovery, authorization, IP address configuration, router discovery, and mobility management procedures. Depending on the network and IP version, the cost of movement in terms of message count may be up to 16 messages. Multi-homing management has not been in the scope of mobility management protocols, but some recent development initiatives would like to look at both problems together [see e.g. 29]. In multi-homing, the frequency of location changes is typically assumed to be slower than in mobility but the primary management problems remain more or less the same.

There are different approaches for lowering the costs of movements, for example optimizing the procedures at different protocols layers [5, 11, 25, 13], or maintaining context information at the upper layer while isolating the changes to lower layers [e.g. 27, 4, 22, 7]. Local mobility management in different roaming scenarios has produced different architectural proposals, e.g. hierarchical structures of mobile anchor points [5, 25], or fast vertical handovers and context transfers between adjacent routers [13]. The shortcomings of these approaches are typically related to security. Most of the proposals require a Public-Key Infrastructure (PKI), and heavy IPsec processing even though there is no global key management infrastructure [18].

There has been recent interest in"opportunistic" or "weak" security procedures that are known to be vulnerable to active man-in-the-middle (MitM) attacks in the first message exchange, but which would still provide some security. In these approaches, the end-points are typically not authenticated in terms of knowing the real identities. Instead, the goal is to know that the entity remains the same during the communication. One example of such a procedure is a Diffie-Hellman key exchange using self-signed public key certificates [e.g. 23]. A benefit of this procedure is that deployments could start using Public-Key based cryptography even though key distribution and verification infrastructures did not exist. Another example of weak security is a procedure in which shared secrets or tokens are exchanged in clear text via two separate communication paths. For example, the MIPv6 return routability procedure assumes that attackers are not able to see messages in both paths, and consequently are not able to construct the secret [3, 10].

A lot of focus has been put on two kinds of attack, namely Denial-of-Service (DoS) attacks, and re-direction (or Distributed DoS, DDoS) attacks

[3]. DoS is typically prevented by delaying the phase when a state is created. The entity that initiates the communication is generally required to do most of the security processing before the responder gives much attention to him. The entity that responds to requests tries to remain stateless as late in the procedure as possible. Creating state too early opens a door for various DoS attacks. Another common method for DoS resistance is delaying the processing load. For example, public key operations are vulnerable to DoS attacks if the communication protocol requires lots of public key checking by the responder at the beginning. In most cases, protocols add computational load (e.g. by introducing cryptographic puzzles) to the initiator side.

Protection against re-direction attacks requires confirmation that there is really someone expecting a response at the source address, i.e. the attacker is not trying to re-direct the message flow to the victim's current location. It is generally not wise to trust blindly the location information. Quite often, communication protocols check that the communication peer is reachable at the source address, and is able to return some negotiation parameters from that address.

3. REVERSE HASH CHAINS

Our work is based on the simple, and well-known cryptographic construction called the "reverse hash chain" (or "hash chain" for short). [15] first introduced the method, and it has been applied in several areas, for example for public key certificate management [17], micro payments [24, 28], (anonymous) authentication [15, 8, 12], and micro mobility and routing protocols [26, 9]. Hash chains have also been deployed in a binary tree format [cf. 16, 28, 26, 9], however, in this paper we focus on the chain structures.

Technically speaking, a hash chain is a cryptographically generated list of inter-related data entities. It is practically impossible to calculate or otherwise figure out the next value in the chain even when you know the previous value. However, it is very easy to verify that some given value is the next value of a chain. A hash chain is a relatively secure method to be used in communication protocol designs when compared with other similar weak methods, such as the use of cookies, tokens or secret splitting.

A hash chain is created by recursively computing a hash function over a result of the same function. The initial argument for the first hash value computation is typically a large random number. The last generated value of the chain is called the "anchor" or "root" value. The hash values are revealed in reverse order starting from the anchor value. This technique is usually

applied based on an assumption that only an authentic end-point knows the correct predecessor values of the chain.

Reverse hash chains can be used as keys in integrity protection and message origin authentication [cf. HMAC in 14]. However, the result is somewhat different from more typical message protection methods, such as shared secret based schemes. Firstly, anybody who is able to receive the subsequent messages is able to verify that the messages belong together. Secondly, message authentication with hash chain values needs to be delayed because the input value (the key) is not revealed until the next message. Even though the verification is delayed, this procedure can be used to verify that all subsequent messages come from the same entity as the first message if the hash chain is used to bind the messages together.

If two communicating entities want to use hash chains to protect their communication, they need to exchange anchor values. If the exchange is done without protection, a Man-in-the-Middle (MitM) attacker may replace the anchor value with its own hash chain. Note, however, that the use of hash chains makes the MitM attack much harder than if, for example, clear text passwords were used. With clear text passwords, the attacker can be passive, and just monitor the traffic to get the password, but with hash chains the attacker must be active right from the beginning in order to replace the anchor values.

A MitM attack can be mitigated by protecting the anchor value with a delayed message authentication code, and by sending the plain text anchor value and the message authentication code via different communication channels. In this case, the attacker must have access to both channels in order to perform the attack.

If the chains are short (which they should be in order keep the computational load low), there is a risk that a chain runs out of values. In this situation, the principles may need to re-negotiate new anchor values. However, it is also possible to link subsequent hash chains together into a longer chain by using the last value of one hash chain to protect the message carrying the anchor value of the next chain. For this reason, the length of the hash chains is not considered as a problem in this study.

4. FRAMEWORK

Our solution framework mimics the message structure of MIPv6 route optimization [10]. Context Establishment is used to establish state, to exchange the anchor values of reverse hash chains, and to initiate two

locators. A Binding ModificAtion message updates location information, and it is only sent from an already verified location.

4.1 Context establishment

The context establishment (CE) exchange creates a state between an initiator (I) and a responder (R). The procedure uses the delayed authentication principle in which the initial message exchange is verified with the parameters included in the next message. The anchor values of the hash chains are agreed via two separate communication channels in order to make the MitM attack more difficult. The first round-trip of context establishment is designed to be stateless for the responder side. At the end of the exchange both initiator and responder have the anchor value of the other communication peer.

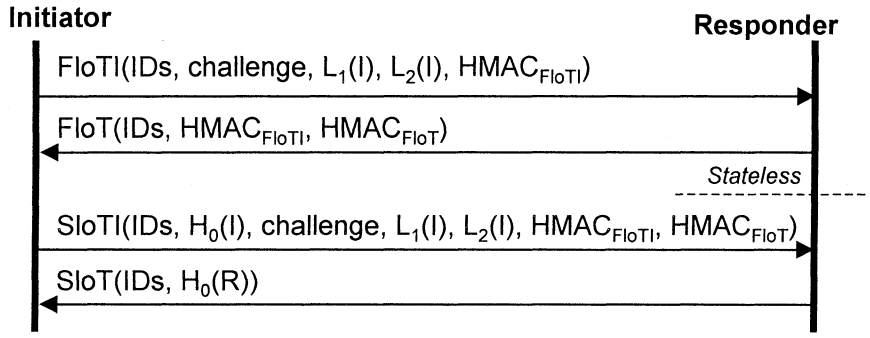

Figure 1: Context establishment

The initiator first sends the First Locator Test Init message, FloTI, to the responder via the first location L1(I). The FloTI message contains the identities of the initiator and responder (IDs), a challenge, location information L1(I) and L2(I), and a keyed hash, HMACFloTI. The HMACFloTI includes the anchor of a newly generated hash chain as a key, and it is computed over all other parameters in the message ("||"represents concatenation):

- HMACFloTI = {keyFloTI, messageFloTI}
- keyFloTI = H0(I)
- messageFloTI = IDs||challenge||L1(I)||L2(I)

Once the responder receives the FloTI message, it must check that the message has one of the locators as a source address. The responder must also check that it does not already have a context with the ID pair. If the context is not found, the responder continues with the negotiation. However, it does not want to establish a state because it is not able to verify the origin of the

message. In order to remain stateless, the responder computes a temporary hash chain using the initiator's parameters in the FloTI message, and sends a First Locator Test message (FloT) to the initiator. The FloT message is protected with the anchor value of the responder hash chain. Note that the responder must be able to reconstruct the same hash chain based on the parameters that are present in the SloTI message in order to be stateless during the test of the first location (FloTI/FloT). This can be done securely, for example, by using a local secret as one input to the hash chain generation. Other useful input parameters are end-point identifiers, the challenge of the initiator, and the initiator's location information.

The keyed hash for the FloT message is computed using the anchor value as a key:

- HMACFloT = {keyFloT, messageFloT}
- keyFloT = H0(R)
- messageFloT = IDs|| HMACFloTI

The initiator replies to the FloT message with a Second Location Test Init message (SloTI). The SloTI message reveals the initiator's anchor value, and it is sent from the second location .

Again, the responder does not accept SloTI packets with an ID pair that already has a host pair-context. If the context is not found, the responder re-computes its own hash chain and verifies the message authentication codes (HMACFloTI and HMACFloT). The anchor value of the initiator hash chain binds the FloTI and SloTI messages together, and in this way the responder is able to verify that the messages are coming from the same entity. If the keyed hashes are valid, the responder creates the state, and replies with a Second Locator Test message (SloT) revealing its own anchor value.

The initiator verifies the keyed hash in the FloT message with the anchor value received in the SloT message, and finalizes its state.

From the responder's point of view, the context establishment is able to verify only the first location of the initiator. The responder cannot trust that the second location (L2(I)) is authentic until this locator is tested. For example, it is still possible that the initiator forges the source locator in the SloTI message (source address spoofing). In this case, the attacker never receives the SloT message, however, it may try to fool the responder to e.g. forward a media flow to a victim (re-direction attack).

Note also that the procedure includes some identity and security context information (marked as "IDs" in Figure 1), which is left open on purpose. Identities and/or security context names are a crucial part of the security of this framework. For example, naming a security context solely by IP addresses is not wise unless the ownership of the IP addresses can be confirmed by some other means [e.g. by the use of cryptographically

generated addresses as specified in 3]. Otherwise, an attacker is able to "steal" the IP addresses from authorized parties. Allowing multiple contexts from/to the same IP address is a better strategy if IP address ownership cannot be verified. An attacker may still use a false IP address, however, the real user can also use it.

4.2 Binding modifications

Once the state has been completed, both entities may send to their peers an update message on the locator sets. The hash chains are used as keys in delayed message authentication, and consequently each locator update operation will require three messages. However, this is wise anyhow because of a potential re-direction attack, i.e. the new locator may be pointing to the victim's current location instead of the initiator's current location.

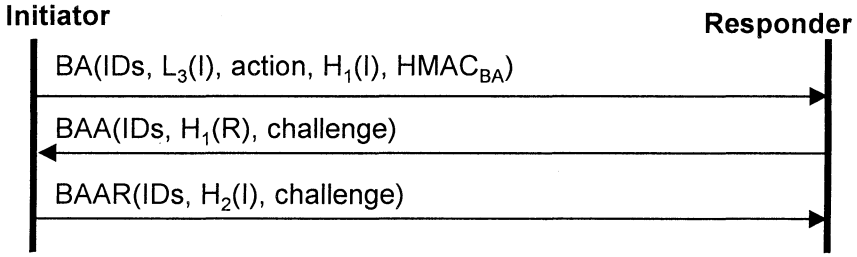

Figure 2: Binding ModificAtion

The Binding ModificAtion message (BA) includes the locator, which is about to be modified, e.g. locator "L3(I)" in the figure above. It also includes information about the action to be performed for this location, e.g. added as a new locator, or deleted because it is not in use anymore. Adding a message authentication code HMACBA protects the locator update;

- HMACBA = {keyBA, messageBA}
- keyBA = H2(I)
- messageBA = IDs||L3(I)||action

Once the responder receives the BA message, it verifies that the hash chain value H1(I) belongs to the initiator (this example assumes that the previously revealed hash chain value was the anchor, H0(I)). The responder replies with the Binding ModificAtion Acknowledgement message (BAA) to the received location. The BAA message includes the next value of the responder's hash chain, and a challenge. The challenge is returned back in the next message, and it is needed in order to avoid a re-direction attack.

The initiator verifies that the hash chain value H1(R) belongs to the responder. The Binding ModificAtion Acknowledgement Reply message

(BAAR) completes the locator update procedure, and it includes the next value of the initiator's hash chain, and the challenge from the BAA message. By returning the challenge, the initiator demonstrates that it really received the BAA message, and did not just wait for some time, and forward the BAAR message from some location (i.e. source address spoofing, re-direction attack).

The responder verifies the challenge and that the hash chain value H2(I) belongs to the initiator. The responder also verifies that all parameters in the original BA packet were unmodified using HMACBA. After successful verifications, the responder changes the state of locator L3(I) according to the requested action.

Even though we considered the length of the hash chain as a non-issue for this framework, it should be noted that the first message of Binding ModificAtions could be used for bootstrapping new hash chains. In this case, the BA and/or BAA message(s) includes also the anchor of the new hash chain. The anchor values must naturally be protected with HMAC using a value from an already existing hash chain as a key.

5. USE CASES

This section demonstrates the use of the framework in multi-homing and mobility contexts. Examples are not intended to be exhaustive protocol designs but rather act as simplified "proofs of concept". The first case example focuses on multi-homing, and the second on local mobility management. Note that the use cases cover two fundamentally different deployments of the framework, i.e. in the first example the communication paths are physically separate while in the second case the separation is logical.

5.1 Multi-homing management

In multi-homing management, the Multi-homing Node and some Responder have two physically separated communication paths – at least on the Multi-homing side. Communication paths may join close to the Responder. See figure 3.

Utilization of our framework is straight forward for this use case. HMAC values are exchanged via the first location (e.g. by piggybacking them in TCP SYN messages), and the clear text hash chain anchor values via the second location. Note, however, that there is no absolute need to finish the context establishment until the multi-homing node wants to start using the

second location (if ever). This use case may cause two parallel context establishment procedures, and should be further studied.

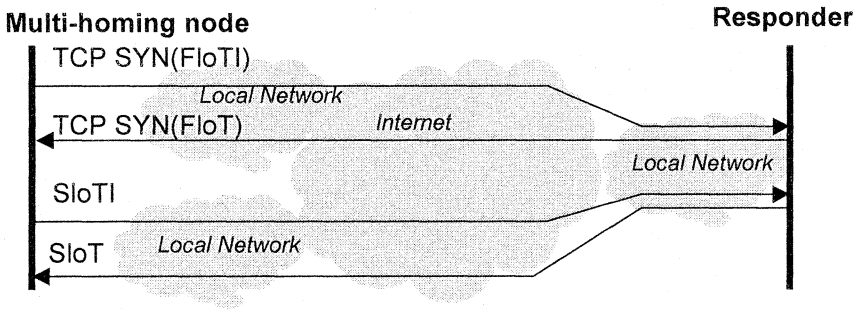

Figure 3: Context establishment in multi-homing context

An active MitM is able to change the hash chain anchor values from the context establishment close to the Responder. However, this is not weaker if compared to the use of IPsec in opportunistic mode, for example.

5.2 Local mobility management

As we stated before, security has been the biggest problem in developing efficient mobility management procedures. Most of the protocol proposals simply require the use of PKI in order to work in real-life roaming situations. Alternatively, the security associations must be configured manually.

Optimized mobility management proposals typically include some local mobility management entity (LME) in the visited/access network, e.g. a mobile anchor point in [25], or previous/next access router in [13]. Common for all these proposals is that mobile node (MN) needs to set security association with this entity.

The use of a LME does not remove the need for the MN to have a security association with the Home Agent (HA). Every time the MN changes its location, it must still update its new location with the HA – no matter if the new location is the real location of the MN, or the address of its LME. Once the MN is behind the LME, it does not need to update its location information while moving under the area of the LME. Binding Updates (BUs) are typically assumed to be sent to HAs using IPsec.

The use of our framework in this context requires the presence of two logically separated communication channels. Even though the MN and the LME do not have two physically separated communication channels, they do have two logical channels; one direct end-to-end path, and another path via the MN's HA (protected with IPsec between the MN and HA). Note also

that the FloTI and SloTI messages are likely to arrive at the LME from different directions, especially if the LME is a "NAT-like" device.

The framework can be applied in this way: the FloTI and FloT messages are tunnelled via the HA using the HoTI/HoT message pair from the MIPv6 return routability procedure [10]. From the HA's point of view, the LME acts as the Correspondent Node (CN). The SloTI/SloT message pair can be exchanged directly between the MN and the LME without the MIPv6 return routability tunnelling. See figure 4.

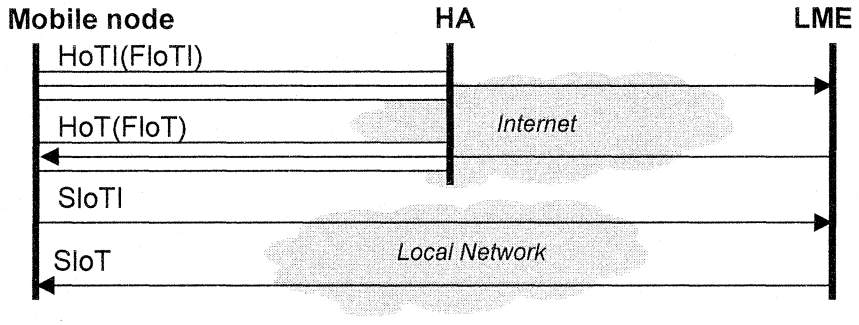

Figure 4: Local mobility management

Once the MN moves to a new location, it can send the Binding ModificAtion message to the LME. Note that the MN and the LME do not have any pre-configured security association, but they are able to create a weak one by relying on the hash chains and separate communication paths.

In theory, the context establishment could also be used in situations where the exchange messages are sent via two access routers (e.g. via the previous and next access routers). This scenario is more vulnerable to certain attacks because IPsec cannot be used, however, it could still be useful for some more limited use cases. For example, the use of this procedure could be secure enough to protect binding updates for ongoing upper layer sessions. An attacker acting as an access router may be able to temporarily hijack the session, however, there is nothing to prevent the MN from sending new binding updates to the CN via the HA. Also, the upper layer security procedures may still be used to protect the confidentiality of the communications.

6. CONCLUSION

In this paper, we have further developed the ideas related to "weak" or "opportunistic" security procedures in a mobile and multi-homing context.

Our exercise demonstrates that development of weak security protocols is possible, and that weak security seems to have some attractive properties especially from efficiency and effectiveness points of view.

In general, our security context establishment procedure is more secure than the return routability procedure in MIPv6 because our procedure requires an active MitM attacker to maintain reverse hash chains. Our procedure is also more efficient and scalable compared to existing protocols, and protocol proposals that mostly rely on PKI or manual keying.

We believe that weak security mechanisms may play an important role in mobility and multi-homing management in the near future. However, developing "forwards compatibility" with stronger security methods, such as PKI, HIP [e.g. 19] or cryptographically generated addresses [e.g. 2, 20], is not a bad idea assuming that these kinds of mechanisms may take over some day. Integration of a public key based method to our procedure can be easily done by adding public key information as part of the initial value of the hash chain operation. For example, signing some parameters from the context establishment, and revealing the signature later in the process could provide a nice migration path between these technologies.

7. REFERENCES

[1] Abley, J. Black, B. & Gill, B. Goals for IPv6 Site-Multihoming Architectures, Internet Engineering Task Force (IETF), RFC 3582, 2003.

[2] Aura, T. Cryptographically Generated Addresses (CGA), in Proceedings of 6th Information Security Conference (ISC'03), Bristol, UK, 2003.

[3] Aura, T. Roe, M. & Arkko. J. Security of Internet Location Management, in Asia-Pacific Computer Systems Architecture Conference, ACSAC'02, Monash University, Melbourne, Australia, February 2002.

[4] 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, pp. 42-49, 2000.

[5] Castelluccia, C. HMIPv6: A Hierarchical Mobile Ipv6 Proposal, ACM Mobile Computing and Communication Review (MC2R), Apr. 2000.

[6] Fischlin, M. Fast Verification of Hash Chains, to appear in the Proceedings of RSA Security 2004, Cryptographer's Track.

[7] Grilo, A. Estrela, P. & Numes, M. Terminal Independent Mobility for IP (TIMIP), IEEE Commun. Mag., Dec. 2001.

[8] Haller, N. The S/KEY One-Time Password System, Internet Engineering Task Force (IETF), RFC 1760, 1995.

[9] Hu, Y-C. Perring, A. & Johnson, D.B. Efficient Security Mechanisms for Routing Protocols, in Proceedings of Network & Distributed System Security Symposium 2003 (NDSS '03), February 6-7, San Diego, CA, pp. 57-73.

[10] Johnson, D. Perkins, C. and Arkko J. Mobility Support in IPv6, Internet Engineering Task Force (IETF), RFC 3775, 2004.

[11] Kempf, J. (editor) Problem Description: Reasons For Performing Context Transfers Between Nodes in an IP Access Network, Internet Engineering Task Force (IETF), RFC 3374, 2002.

[12] Kim, J. Provable Secure Anonymous Authentication Protocol based on Hash Chains, A Thesis for the Degree of Master of Science, Information and Communications University, South Korea, available: http://caislab.icu.ac.kr/pub/thesis/down/jskim.pdf, 2003.

[13] Koodli, R. (editor) Fast Handovers for Mobile IPv6, Internet Engineering Task Force (IETF), work in progress, draft-ietf-mipshop-fast-mipv6-01.txt, 2004.

[14] Krawczyk, H. Bellare, M. & Canetti, R. HMAC: Keyed-Hashing for Message Authentication, Internet Engineering Task Force (IETF), RFC 2104, 1997.

[15] Lamport, L. Password Authentication with Insecure Communication, Communications of ACM, Vol 24, No 11, pp. 770-772, 1981.

[16] Merkle, R. Secrecy, authentication, and public key systems, Ph.D. dissertation, Dept. of Electrical Engineering, Stanford University, 1979.

[17] Micali, S. Efficient Certificate Revocation, Technical Report, MIT/LCS/TM-542b, MIT Laboratory for Computer Science, 1996.

[18] Mink, S. Pahlke, F. Schafer, G. & Schiller, J. Towards Secure Mobility Support for IP Networks, in Proceedings of the IFIP International Conference on Communication Technologies (ICCT), Aug. 2000.

[19] Moskowitz, R. & Nikander, P. Host Identity Protocol Architecture, Internet Engineering Task Force (IETF), work in progress, draft-moskowitz-hip-arch-05.txt, 2004.

[20] O'Shea, G. & Roe, M. Child-proof authentication for MIPv6 (CAM), ACM SIGCOMM Computer Communication Review, Vol. 31, No 2, pp. 4-8, 2001.

[21] Perkins, C. IP Mobility Support, Internet Engineering Task Force (IETF), RFC 2002, 1996.

[22] Ramjee, R. Porta, T. Salgarelli, L. Thuel, S. & Varadhan, K. IP-based Acess Network Infrastructure for next Generation Wireless Data Networks, IEEE Personal Commun. Mag., Vol. 7, No. 4, 2000.

[23] Richardson, M. & Redelmeier, D. Opportunistic Encryption using The Internet Key Exchange (IKE), Internet Engineering Task Force (IETF), work in progress, draft-richardson-ipsec-opportunistic-15.txt, 2004.

[24] Rivest, R.L. & Shamir, A. PayWord and MicroMint--Two Simple Micropayment Schemes, CryptoBytes, volume 2, number 1 (RSA Laboratories, Spring 1996), pp. 7-11.

[25] Soliman, H. Castelluccia, C. El Malki, K. & Bellier L. Hierarchical Mobile IPv6 mobility management (HMIPv6), Internet Engineering Task Force (IETF), work in progress, draft-ietf-mipshop-hmipv6-01.txt, 2004.

[26] Tewari, H. & O'Mahony, D. Lightweight AAA for Cellular IP, in Proceedings of European Wireless 2002, February 25-28, 2002 – Florence, Italy, available in http://www.ing.unipi.it/ew2002/.

[27] Valko, A.G. Cellular IP: a new approach to Internet host mobility, ACM SIGCOMM Computer Communication Review, Vol. 29, Number 1, pp. 50-65, 1999.

[28] Yen, S. Ho L., Huang, C. Internet Micropayment Based on Unbalanced One-way Binary Tree, Proceedings of CrypTEC'99, Hong Kong, July, pp. 155-162.

[29] Ylitalo, J. Jokela, P. Wall, J. and Nikander, P. End-point Identifiers in Secure Multi-Homed Mobility", in Proceedings of the 6th International Conference On Principles Of Distributed Systems, Reims, France, December 11-13, pp. 17-28, 2002.