Securing Dynamic Home Agent Address Discovery with Cryptographically Generated Addresses and RSA Signatures

Christian Bauer
Institute of Communications and Navigation
German Aerospace Center (DLR)
82234 Wessling, Germany
Email: Christian.Bauer@dlr.de

Max Ehammer
Institute of Computer Science
University of Salzburg
5020 Salzburg, Austria
Email: mehammer@cosy.sbg.ac.at

***Abstract*-With Dynamic Home Agent Address Discovery (DHAAD), as specified in Mobile IPv6, a Mobile Node can discover the address of a suitable Home Agent on the home link. However, DHAAD suffers from security problems as the signaling is not authenticated nor integrity protected. The IETF has defined SEcure Neighbor Discovery that is providing security for the IPv6 Neighbor Discovery protocol, based on several asymmetric cryptographic mechanisms. It is shown that these mechanisms can also be used to secure DHAAD to increase its level of protection and to provide resistance against attacks.**

I. INTRODUCTION

The Mobile IPv6 (MIPv6) protocol [9] is the standard global mobility management protocol of the IETF that has been adopted by 3GPP, 3GPP2 or Next Generation Networks in general.

The Network Mobility (NEMO) Basic Support protocol [5] extends the MIPv6 specification to support a Mobile Router (MR). Unfortunately, NEMO did not specify Route Optimization (RO) so far. The Car2Car communication consortium and the aviation industry are interested in deploying IP networks in general and NEMO in particular. Therefore, a solution to the NEMO RO problem has to be found as this is a deployment obstacle for these industries.

One of the proposed solutions to solve the NEMO RO problem is the Global Home Agent to Home Agent protocol [15]. This protocol extends the Home Agent (HA) concept from having a single entity on the home link towards several geographically distributed Home Agents (perhaps world-wide). A problem that has to be solved within this context is locating the closest HA to the Mobile Node (MN) or Mobile Router (MR), for which the Anycast based Dynamic Home Agent Address Discovery mechanism, as specified in [9], is a reasonable candidate. The semantics of Anycast are suitable for this task as data is routed to the "nearest" destination, with distance being defined by the routing metrics.

MIPv6 Dynamic Home Agent Address Discovery (DHAAD) specifies that a Mobile Node may discover the address of a suitable Home Agent on the home link. Due to the fact that the message exchange can be misused and the transported information is important for a MN/MR (it contains the HA addresses), it becomes necessary to secure this DHAAD signaling, which is currently not the case for [9]. One way of improving the security is presented within this paper, and we argue that asymmetric cryptography provides the necessary means to accomplish this task.

A comparable proposal has already been presented in [13], but it relies on shared secrets and therefore inherently suffers from the problems of scalability (e.g. how to distribute shared secrets among all MN-HA pairs). An asymmetric cryptographic method is to be preferred. The SEcure Neighbor Discovery protocol [2] provides mechanisms that can be reused for this purpose. Therefore, our proposal is based on asymmetric mechanisms, reusing parts of SEND, increasing the level of protection for the DHAAD signaling between MN and HA. The parts of DHAAD that relate to Home Agents keeping track of available neighboring Home Agents is not covered by our proposal. It is assumed that the home link is physically secured against attackers trying to attach to this link.

The presented mechanisms are applicable to both a MN and a MR, but for simplicity we will from now on only use the notion MN, although this is not excluding a MR from using these enhancements – in the opposite, from the aviation industry point of view Network Mobility [5] is far more usable than Host Mobility as in MIPv6 [9].

The rest of this paper is structured as follows: Sections II and III provide an overview of DHAAD, its problems and proposals for enhancing its security that have been presented so far. In Section IV we define the requirements for our solution and in Section V we introduce SEcure Neighbor Discovery (SEND) that is partially reused to increase the level of security for the Neighbor Discovery Protocol. In Section VI we provide an overview of all options that we are using from DHAAD and SEND. In Section VII the proposed mechanisms to secure DHAAD, based on SEND, are presented. We investigate the signaling overhead in Section VIII and finally analyze the advantages and disadvantages of our proposal in Section IX.

This work is partially funded by the European Commission through the NEWSKY project under contract no. 37160.

II. DHAAD

The DHAAD procedure [9] consists of a request-response signaling: the MN creates an ICMP Home Agent Address Discovery Request Message and sends it to the Mobile IPv6 Home Agents Anycast address that is constructed from the home subnet prefix, with the current Care-of Address (CoA) of the MN as source address of the request. The HA that receives the message responds with an ICMP Home Agent Address Discovery Reply Message that includes a list of available Home Agents on the home link, sent to the CoA of the MN.

DHAAD was considered insufficient for bootstrapping Mobile IP, which is – in general - a procedure to dynamically configure a Mobile Node (i.e. MIPv6 requires a MN to be pre-configured with a valid Home Address, a Home Agent Address, a Home Network Prefix, and cryptographic material to establish a Security Association (SA) with its Home Agent). The reason why DHAAD was considered insufficient is that a MN needs to be preconfigured with several of the parameters mentioned before. In addition bootstrapping based on DHAAD also prevents effective load balancing between different HAs if an operator runs several home links and would like do distribute the MNs among these. Hence separate bootstrapping specifications [7, 4] were defined to address these deficiencies. Either DNS or DHCPv6 may be used to retrieve the address of the HA and subsequently assign the MN a Home Network Prefix and a Home Address to allow establishment of a Security Association between MN and HA.

Additionally, DHAAD suffers from the security point of view. Without any already established Security Association no confidentiality nor integrity or any means to authenticate the MN, which is the sender of the ICMP Home Agent Address Discovery Request Message, can be provided. The same is valid for the reply message. The content of the Home Agent Address Discovery Reply Message cannot be integrity protected and the sender cannot be authenticated as well (that ought to be a valid HA).

Nevertheless, DHAAD is still useful if Home Agents are not operated by a Mobility Services Provider offering DHCPv6 or DNS services. Additionally, DHAAD is reasonable if the Global Home Agent to Home Agent protocol [15] is deployed, where modified DHAAD signaling could be used to locate the Home Agent that is topologically closest to the Mobile Nodes current point of attachment.

III. RELATED WORK

In [13] the authors suggest to use a shared secret key between the HA and the MN that is then used to calculate a hash value for both authenticating the sender of the message and verifying the integrity of the message itself.

In the DHAAD request (shown in Fig. 2), a new Mobile Node Identifier Option is used to carry the identity of the MN; in addition the already existing identifier field, set to a random value, is extended from 16 to 24 bits (using the reserved bits for this purpose) to lower the probability of identifier collisions. The identifier field is used for ensuring protection against replay attacks, as the reply message that matches the request has to use the same value in its own identifier field.

Although this works for MIPv6, it does not necessarily work for NEMO Basic Support where one bit from the reserved ones is used to set a router flag to indicate that the sender is looking for a HA that supports Mobile Routers.

The authors suggest obtaining the shared key via pre-configuration or the bootstrapping mechanism. In both cases the complexity of managing a high number of keys remains a major problem. If a unique (MN, HA) shared key is used, as suggested by the authors, its application in a Global HA to HA [15] network layout with many geographically distributed HAs becomes problematic, as each (MN, HA) key pair has to be distributed and stored at all Home Agents. Given the potentially high number of MNs for the Car2Car or for the aviation scenario, a scalability problem would have to be expected. For the latter, forecasts expect 15.5 to 18.9 million flights in Europe for the year 2025, which corresponds to approximately 50.000 flights per day if a uniform distribution is assumed [6]. This problem also occurs in a classical MIPv6 deployment without any geographically distributed HAs, as the scalability factor is the number of MNs.

Considering only a single shared key between all MNs and HAs, the scalability problem vanishes but a security problem arises if one of the MNs becomes compromised. This malicious node could use the identity of someone else (e.g. by faking the content of the Mobile Node Identifier Option) for any purpose when conducting an attack. Additionally, a new shared key has to be distributed among all HAs and MNs to counter this threat, as soon as the attack has been recognized.

Our proposal is supposed to address these deficiencies, but before continuing we first want to define the security and scalability requirements for our solution.

IV. GOALS

Our goals for securing the DHAAD mechanism are the following:

- It shall be possible to authenticate the sender of the request message (valid MN).
- Replay attacks with an old request message shall be prevented.
- It shall be possible to provide integrity protection for the response message
- It shall be possible to authenticate the sender of the response message (valid HA)
- Replay attacks with an old response message shall be prevented.
- The approach should be scalable to a high number of MNs and HAs.

Therefore, we propose to use mechanisms of asymmetric cryptography. This eliminates the scalability problem and allows reliable identification of every individual MN and HA.

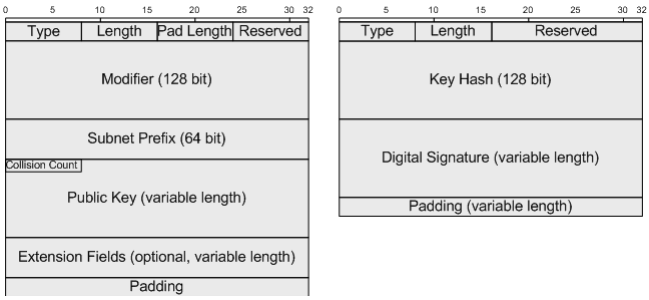

Fig. 1. CGA (left side) and RSA Signature (right side) options

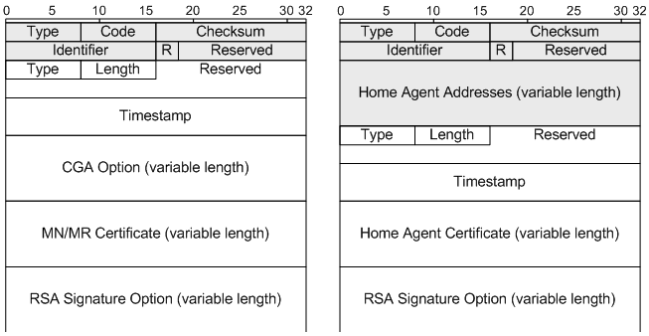

Fig. 2. DHAAD Request (left side) and Response (right side) messages. Original fields in grey, new parts in white.

A similar approach was used for the Secure Neighbor Discovery (SEND) protocol, which we are going to present shortly.

V. SECURE NEIGHBOR DISCOVERY

The Neighbor Discovery Protocol (NDP) as specified in [11] is used by hosts and routers in IPv6 networks alike to learn the local topology. Among others, this is done by building an IP to MAC address mapping (based on Neighbor Solicitations and Neighbor Advertisements) and by learning the prefixes advertised by the routers on the local link (based on Router Advertisements (RAs)). NDP suffers from several vulnerabilities that were identified in [12]. To counter these threats, the SEcure Neighbor Discovery Protocol [2] has been defined where a zero-configuration mechanism proves the (link local) IP address-ownership of individual nodes and access routers are certified by a trust anchor, the latter requiring a PKI infrastructure covering both MNs and routers.

The two probably most significant parts of SEND are Cryptographically Generated Addresses (CGAs) and certificates that authorize access routers to advertise a certain subnet prefix. Both mechanisms are of interest for securing the DHAAD procedure.

Cryptographically Generated Addresses [3] take a 64-bit subnet prefix, the public key of the address owner and auxiliary parameters as input to an algorithm. This algorithm then generates the 64 bit interface identifier that is finally concatenated with the 64 bit subnet prefix to form the 128 bit IPv6 address of a node. This way the IPv6 address is cryptographically bound to the public key of the address owner and cannot be regenerated by any other public key (or more precisely, the probability for being able to generate the same CGA with a different public key is very low). The association of a CGA to a certain public key can be verified by any node that knows the public key, the values of the auxiliary parameters used for generation as well as the address itself. These parameters are carried within a CGA Option that is shown in Fig. 1.

Additionally, the sender signs the associated message with a RSA signature by using the complementary private key that was used to generate the CGA. The RSA Signature option – shown in Fig. 1 – has a key hash field that contains the most significant 128 bits of the SHA-1 hash of the public key that was used to generate the signature. This value will usually correspond to the public key that is carried within the CGA option of the same message. The Digital Signature field contains the signature value itself that is generated with the RSASSA-PKCS1-v1_5 algorithm and SHA-1 hash as defined in [14].

The receiver of the protected message can then verify the message's authenticity primarily based on the RSA signature, while also using the CGA data structure. The receiver obtains this information (Signature and CGA data structure) by means of in-band signaling with the Neighbor Discovery message that has to be protected – this means that the options are directly attached to the message that has to be protected.

CGAs themselves do not have to be certified and do not rely on a public key infrastructure, hence attackers can use an arbitrary public key (e.g. of someone else) and generate a CGA from it. However, it is not possible for this attacker to take ownership of someone else's CGA for use in SEND as he cannot sign messages (generating a valid RSA Signature option) due to the missing complementary private key. As already mentioned before, searching for a public key that generates the same address is a brute force attack, which cannot be considered as a serious threat.

In more detail, the sending/verification process of a protected message is as follows: the sender of the message to be protected, either host or router, uses the CGA as source address, generates and attaches the CGA and RSA signature options and sends the message to its recipient. The receiver first verifies the claimed CGA (taken from the CGA option) as specified in Section 5 of [3], basically generating a SHA-1 value and performing simple comparison operations. If this verification is successful, the CGA is considered to be valid. The next step involves the more time consuming cryptographic check of the signature that verifies the message content and asserts that the sender is also the CGA owner (that is, the RSA signature was generated from the private key that complements the public key that was used to generate the CGA). Only if both checks are completed successfully the message can be considered as authentic.

TABLE I
SIZE OF MESSAGES AND OPTIONS

Message / Option	Size in Bits
DHAAD Request Message [9]	320 + 32 + 64
DHAAD Response Message [9]	320 + 32 + 64 + 128 * #(HAs) = 416 + 128 * #(HAs)
Timestamp Option [2]	128
CGA Option [3,2]	232 + ~100 + 1024
MN Certificate Option (estimated)	32 + ~8800
HA Certificate Option (estimated)	32 + ~8960
RSA Signature Option [2]	160 + 1024

Authenticating a Router Advertisement requires additional mechanisms as CGAs only protect addresses but not prefixes. It has to be ensured that the sending node is authorized to act as a router and can advertise a certain set of prefixes. For this reason RAs contain Router Authorization Certificates that are rooted in a trust anchor. The private key used to generate the RSA signature for the RA is the one that complements the public key contained in the Router Authorization Certificate. These certificates are X.509v3 certificates that contain X.509 IP address extensions that specify which subnet prefix a router is authorized to advertise.

VI. MESSAGES AND OPTIONS

We provide an overview of all messages, options and certificates that we are going to use and modify. For a better understanding these messages and options are shown in Figures 1 and 2. The original DHAAD Request and Response messages from [9] are the grey parts shown in Fig. 2. The relevant parts of SEND that we are going to use to protect the signaling are the CGA and RSA signature options shown in Fig. 1, the Timestamp option and the Certificate option containing a X.509v3 certificate for either MN or HA. A structural sketch of a X.509v3 certificate is shown in Fig. 3.

Both DHAAD request and response are extended with these options introduced by SEND. Both can be seen in Fig. 2 where they are represented with the new additional options appended to the original DHAAD messages.

The Timestamp option contains the current time and date, expressed in number of seconds since January 1, 1970, 00:00 UTC.

The CGA option contains all necessary information to allow a recipient to verify the CGA itself and the RSA signature of subsequent messages. It includes the public key, the subnet prefix and the modifiers used for CGA generation.

The RSA signature consists of the 128 bit SHA-1 hash of the public key that is transported with the CGA option as well as a variable length field for the signature itself that is used for the integrity check.

The MN/MR/HA Certificate options contain X.509v3 certificates [8] that have a variable length dependent on the implemented extensions.

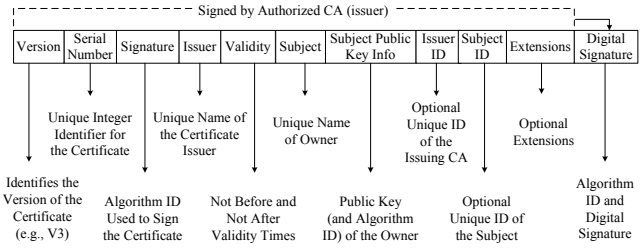

Fig. 3. X.509v3 certificate structure [1].

In the following a short overview of the X.509v3 certificate is given. The *version* field indicates the version (either 1, 2, or 3). The *serial number* is a unique identifier of the certificate relative to the certificate issuer. The *signature* indicates the algorithm identifier of the algorithm used to calculate the digital signature on the certificate. The *issuer* field contains a name of the Certification Authority (CA) that issued the certificate and must always be present. In the *validity* field the time window that this certificate should be considered valid unless revoked is given. The *subject* indicates the name of the certificate owner and must not be null unless an alternate name form is used. The *subject public key info* contains the public key and the algorithm identifier of the associated subject and must always be present. The *issuer and subject unique ID* fields are optional and are usually not present in certificates. At the end of the certificate several extensions may be appended. The very last field contains an algorithm identifier and the signature that is generated by the issuer. It is calculated over all previous fields.

Similarly, the public key inside the CGA option (cf. Fig. 1) is also stored in a field of type *subject public key info*.

For the Router Authorization Certificate (that we reuse as Home Agent Certificate) a X.509v3 extension is necessary. [10] specifies extensions for IP addresses and AS identifiers. For our purposes only the IP address extension is of interest. In this case prefixes are bound to the subject of the certificate.

The length of X.509v3 certificates varies dependent on the entries within the general fields and the specific extensions. Therefore, it is difficult to estimate the size of a certificate. However, in general we can assume that the system deployment will be conducted in a favorable way, which means that the certificate will be kept small if bandwidth is an issue. Currently certificates have a typical length of approximately 1100 bytes and more, dependent on the number of extensions present in a specific certificate. Therefore, we assume such a size as reasonable.

X.509 certificates are already in use in the aviation environment and are expected to be even more used in the future. We therefore think that it is valid to propose their usage in this environment that we are mainly targeting.

VII. SECURE DHAAD

The Mobile Node constructs the basic DHAAD Request message as defined in [9], with the source address being a CGA based Care-of-Address. Additionally, a Timestamp option (same format as in [2]) is attached with the current time. The MN also adds the CGA option, a Certificate option (comparable to the Router Authorization Certificate but without the IP address extensions), and a RSA Signature option whose signature is calculated over the source and destination address and the ICMP DHAAD request message. The public key in the CGA option must be the same as in the Certificate option, and the latter must have an issuer/trust anchor that is known by the receiving HA and is signed by this entity. The final message can be seen in Fig. 2.

Once the message was constructed, it is sent to the Mobile IPv6 Home-Agents Anycast address for its home subnet prefix, which is a /64 (in the case of [9]) or a shorter prefix covering all HAs (in the case of Global HA to HA [15]). This prefix was learned from the bootstrapping process [7, 4] (also explained in Section II) that would have to be extended in order to provide this information.

Due to the semantics of Anycast, the HA closest to the MNs current point of attachment will receive the request message and start verifying it. The first, computationally cheap, steps consist of checking whether the issuer of the MN's certificate is a known trust anchor, whether the public key in the CGA option is equal to the one in the Certificate option, and whether the time in the Timestamp option is not too old (e.g. should not deviate from the current time by 1.5 seconds) in order to disallow replay attacks. The next steps involve the verification of the CGA (whether it was generated from the provided public key), the signature verification of the issuer on the MN's certificate, and the verification of the RSA signature. If any of these steps fails, the message is dropped and ignored.

In case all checks have been successfully passed the message is considered valid and the HA prepares an appropriate response message. It is constructed with the fields set as in [9], but in addition has a Certificate option with a X.509v3 certificate (comparable to the Router Authorization Certificate in SEND) with an IP address delegation extension that includes the Home Network Prefix of the MN or a prefix that aggregates the Home Network Prefix. The issuer must be a trust anchor that is common to the MN and HA. The Timestamp option from the request message is copied to the response. This way the timestamp acts as a kind of nonce that matches the request and therefore protects against replay attacks, as an attacker can not reply with any previously eavesdropped or intercepted response that was sent to another MN. Finally, a RSA Signature option is appended that is generated from the private key of the HA (that complements the public key in the attached certificate), calculated over the source and destination address, the ICMP header, the field containing the HA address(es), and the Timestamp option of the message.

This response (shown in Fig. 2) is sent from the HA to the MN, that in turn performs the following checks: it validates that the HA certificate has a known issuer and that the prefix contained in the IP address extension of this certificate is equal to the Home Network Prefix of the MN (or is at least an aggregation thereof).

Then it is verified that the value of the Timestamp option matches the one that was sent with the request message. Finally the signature of the issuer in the router certificate and the RSA Signature option are validated.

In case of a positive validation, the MN extracts the Home Agent addresses and uses one of them to establish an IPsec Security Association (SA). Afterwards Binding Updates are exchanged via the SA with this HA.

VIII. Message Sizes

In our secured DHAAD proposal, these messages have significantly improved security, mainly due to the public keys and certificates. However, a disadvantage of this suggestion is the increase of the message sizes. In the original version of MIPv6, DHAAD request and response messages have a size of 52 and 68 bytes each (including the IPv6 header), assuming a single Home Agent listed in the response message. The compared message sizes are based on the values provided in Table I. We have assumed public key sizes of 1024 bits and have not taken into account the padding bits that are usually used at the end of an option to meet the 32bit alignment requirement.

The request now consists of the original message (52 byte) with a timestamp (16 byte), a CGA (170 byte), a (MN) Certificate option (1104 bytes), and the RSA signature (148 bytes): 52+16+170+1104+148 = 1490 bytes.

The response message consists of the original response (68 bytes) with a timestamp (16 bytes), a (HA) Certificate option (1124 bytes) and a RSA signature option (148 bytes): 68+16+1124+40 = 1356 bytes.

Based on that observation it becomes apparent that such a mechanism is only well suited for networks which do not have too stringent bandwidth constraints. The message sizes are still below the common IEEE maximum transmission unit of 1500 bytes that is common for many access technologies. However, it should also be noted that this message exchange only takes place once in a while.

IX. Conclusion

We have presented a possible way of securing DHAAD by means of asymmetric cryptographic mechanisms as defined in SEND. The MN constructs the DHAAD request with several additional options (Timestamp, CGA, Certificate and RSA Signature) that allows the receiving HA to verify the authenticity of the sender. Similarly, the HA sends a response message (with Timestamp, HA Certificate and RSA Signature)

where the MN is able to verify the authenticity and integrity of the response message.

The advantage is that a HA will only respond to requests that are coming from a valid MN – that means, a Mobile Node that has a certificate from a common trust anchor. Replay attacks with DHAAD requests are not possible due to the Timestamp option that gives messages only a certain lifetime. The same holds for the response message as it has to include the timestamp of the request, making sure that responses intercepted by an attacker can not be replayed.

While asymmetric encryption provides strong protection, it is also a disadvantage at the same time as the computational overhead for verifying RSA signatures is significant and can be exploited for Denial-of-Service attacks (DoS). This vulnerability is also present for SEND, however, there it is limited to the on-link location – the attacker has to be on the same subnet as the victim in order to launch his attacks. However, in our secured DHAAD an attacker can be at any location of the overall internetwork for as long as there is a route between him and the victim (HA).

To minimize this vulnerability, the verification of the DHAAD request at the HA is a two-stage process. The first check – verification of the CGA – is computationally inexpensive and forces the attacker to generate a valid CGA, which is a resource expensive task for the attacker himself. In addition the MN also needs a valid certificate, issued and signed by a common trust anchor. Only if the CGA is valid, the signatures of the MN's certificate and the RSA option are checked, too.

The threat to the MN is minimal as it will only attempt to verify response messages that have both an Identifier value (stored in the corresponding field shown in Fig. 2 according to [9]) and a Timestamp option (introduced by our new proposal) that equal those that were used in the original request message. An attacker would have to be on the communication path between MN and HA while the request is sent, learn of the Identifier and Timestamp options by eavesdropping and send a forged reply that uses these two values. Only under this precondition an attacker can force the MN into verifying the certificate and RSA signature.

The proposed mechanism is not providing any confidentiality; hence it is still possible for an eavesdropper to learn the addresses of HA(s) in the home network. However, this is possible through other means as well e.g. the MN established an IKE Security Association with the HA where destination address (HA address) is readable by everyone on the path. Hence, the current security provided by other parts of MIPv6 is not reduced by our proposed mechanism.

The overhead caused by our proposed security scheme could be reduced if the CGA Option would allow a carriage of the complete certificate of the MN (instead of the public key only). This way the additional Certificate option for the request message (as shown in Fig. 2) could be removed. However, this would require a modification of the existing CGA and SEND specifications [3, 2].

The remaining problem is the availability of a public/private key pair at the MN that has to be signed by the common trust anchor/certificate authority. It will usually be the case that devices are already preconfigured with certain information and an additional public/private key pair and certificates should not incur a significant burden, as this is independent to mobility and IP address configuration. In the aviation environment certificates are already partially in use and are considered as even more important in the future. Hence we think that this is not a problem from the deployment point of view.

In the future we will provide algorithm agility (instead of just reusing SEND functionality with hardcoded RSA and SHA-1) within our proposal so that we can also use elliptic curve cryptography that provides higher cryptographic strength with shorter key lengths when compared to RSA. This also helps reducing the message sizes that are a significant problem within low bandwidth environments. At the same time this also addresses the problem of attacks that have been reported on SHA-1 in the past, as we could rely on a different hash algorithm.

Simple digital signatures that allow for fast screening could also help mitigating the CPU exhaustion attacks. The next steps also involve an implementation of the proposed scheme that allows us to investigate the problems of DoS attacks in more detail.

REFERENCES

[1] Adams C. and Lloyd S "Understanding PKI" 2nd Edition, 2002

[2] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[3] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[4] Chowdhury, K. and A. Yegin, "MIP6-bootstrapping via DHCPv6 for the Integrated Scenario", draft-ietf-mip6-bootstrapping-integrated-dhc-05 (work in progress), June 2006.

[5] Devarapalli, V., Wakikawa, R., Petrescu, A., and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, January 2005.

[6] Eurocontrol, Statistics and Forecast Service (STATFOR), "Long-Term Forecast. Flight Movements 2006-2025", Edition Number 1, December 2006.

[7] Giaretta, G., Ed., J. Kempf and V. Devarapalli, Ed., "Mobile IPv6 bootstrapping in split scenario", RFC 5026, October 2007.

[8] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

[9] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[10] Lynn, C., Kent, S., and K. Seo, "X.509 Extensions for IP Addresses and AS Identifiers", RFC 3779, June 2004.

[11] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[12] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.

[13] H. Petander, K. Lan, M. Hassan, S. Qian, M. Lei, "On securing dynamic home agent address discovery of on-board mobile router in mobile IPv6 networks", 12th International conference on telecommunications, IEEE, New Jersey, USA, 2005.

[14] RSA Laboratories, "RSA Encryption Standard, Version 2.1", PKCS 1, November 2002.

[15] Thubert, P., Wakikawa R., and V. Devarapalli, "Global HA to HA protocol", draft-thubert-mext-global-haha-00, (work in progress), March 2008.