Deploying New Hash Algorithms in Secure Neighbor Discovery

Valter Vasić
University of Zagreb
Email: valter.vasic@fer.hr

Ana Kukec
University of Zagreb
Email: ana.kukec@fer.hr

Miljenko Mikuc
University of Zagreb
Email: miljenko.mikuc@fer.hr

***Abstract*—IP protocol version 6 (IPv6), a successor of IP protocol version 4 (IPv4), puts a significant effort in enhancing security facilities. Secure Neighbor Discovery (SEND) is an optional IPv6 protocol that counters threats in link-local communication allowed by Neighbor Discovery protocol (NDP). It protects from attacks against the integrity and authentication capabilities, relying on trustworthiness of cryptographic hash functions. After a recent discovery of reduced hash function resistance, their efficiency has been called into question. Attacks against the collision-free property of hash functions lead to the reexamination of how Internet protocols use hashes. In the paper we contribute with analyzes of attacks on hash functions, use of hashes in SEND, impact of hash attacks on each use of hash functions in SEND, propose and evaluate possible approaches to allowing hash agility, and finally propose the most efficient solution - a solution for SEND hash agility based on a negotiation approach.**

1. INTRODUCTION

There is a great variety of hash functions, but only MD5 and SHA-1 are in the wide use. Both SHA-1 and MD5 derive from MD4, which has been known for its weaknesses [7]. MD5 is a hash algorithm with 128-bit output, designed in 1991 to replace MD4, its earlier version. First weaknesses were discovered in 1995 in the MD5's compression function. In 2005, researchers created a pair of X.509 certificates that resulted in the same MD5 hash, in 2^{40} operations. [14] shows the aforementioned attack, which was performed just by adding additional 4KB of data to the modified X.509 certificate so that the original and crafted certificate would had identical MD5 hashes. Cryptographers then began recommending the use of SHA-1 algorithm, as a safer alternative. However, demonstrated attacks against SHA-1 appeared soon after. In 2005 researchers showed that two messages can have the same computed SHA-1 hash value. The attack was performed in 2^{69} operations. Since the brute force attack against SHA-1 requires 2^{80} operations, 2^{69} operations to break SHA-1 represent a significant improvement of attackers and obvious weakness of the full SHA-1 algorithm [15]. In the case of simpler SHA variants the number of rounds is even fewer. The collision in SHA-0 requires 2^{39} operations while in the 58-round SHA-1 takes 2^{33} operations [13]. This yields a need to migrate from SHA-1 further on. The National Institute of Standards and Technology (NIST) already has standards for, as they claim, longer and harder-to-break, hash functions: SHA-224, SHA-256, SHA-384 and SHA-512. They are named after their digest lengths (in bits) and all belong into the SHA-2 family. Despite the higher level of security, SHA-2 is still not widely used. In November 2007, NIST announced an open competition for a new SHA-3 function. The competition has entered it's final (third) round with five contestants and the selection of a winning function is announced to be in 2012.

Despite the variety of available hash functions, current security protocols rarely support hash agility, but instead require the use of a single hash function, mainly SHA-1. Hash agility represents the possibility of negotiating and selecting different hash algorithms for securing network communication. Secure Neighbor Discovery [1] supports only the use of SHA-1 and does not offer a possibility to negotiate and use multiple hash functions.

In this paper we analyze: attacks on hash functions, use of hashes in SEND, impact of hash attacks on each use of hash functions in SEND and we give a survey of possible approaches to allowing hash agility as well as a solution for SEND hash agility based on a negotiation approach.

SEND is a security protocol mandatory for IPv6, which uses hashes in a number of different ways. SEND counters threats to Neighbor Discovery Protocol (NDP), a rough equivalent of Address Resolution Protocol and a number of Internet Control Message Protocol mechanisms which allow IPv6 link-local communication. SEND uses hashes through the use of X.509 certificates, Cryptographically Generated Addresses [2] and also as raw hash outputs in NDP packet fields. SEND does not provide any mean to use the hash algorithm different than SHA-1. In the rest of the paper we first give a brief introduction into the hash functions and known vulnerabilities, describe the use of hashes in SEND and following this analyze and suggest the use of multiple hashes within SEND.

2. ATTACKS ON HASH FUNCTIONS

Basic cryptographic properties of hash functions [13] are:

- one-way property and
- collision free property.

There are two types of attack against the one-way property, the first pre-image attack and the second pre-image attack. In the first pre-image attack, given a knowledge of a particular hash value y, an attacker finds an input message m' such that $hash(m')$ yields y. The first pre-image attack resistance

is described as the infeasibility to find:

m' such that H (m') = y

The second pre-image attack deals with fixed messages. Given a knowledge of a fixed value m used as the input message to the hash function, an attacker finds a new value m' that yields $hash(m) = hash(m')$. The second pre-image attack resistance is described as the infeasibility to find:

m' such that H (m') = H (m)

The first and the second pre-image attacks might seem similar, but they are actually essentially different. In the first case, the attacker knows a hash value but not the message that was hashed, while in the second pre-image attack the attacker already knows the message and has to find a second message that will result in the same hash. This makes the existence of the first pre-image attack potentially much more dangerous than the second pre-image attack because the hash value is always visible. Supposing that the hash function produces an n-bit long output, since each output is equally likely, attack against the one-way property would take an order of 2^n operations to be successful. Due to the birthday attack, if the hash function is supplied with a random input, it would return one of the k equally-likely values, and the number of operations could be reduced to the number of $1.2 * 2^{n/2}$ operations. Up to date, attacks against the one-way property are not yet feasible. All demonstrated attacks are attacks against a collision-free property, in which an attacker produces two different messages m and m' such that $hash(m) = hash(m')$. Collision resistance is described as the infeasibility to find:

m, y, m ≠ y such that H (m) = H (y)

Internet Protocols use hash algorithms in different ways [8]. On the application level, its common use is the non-repudiable digital signature on messages. They protect against the false denial of being a participant in a communication. Digital signatures in certificates are another common use of hashes. They are used in Internet protocols on all layers, from the network layer up to the application layer. They assume achieving trust through a third party. The integrity protection or the fingerprinting, i.e. raw hash outputs, is one of the simplest uses of hashes. There are also mixed uses of hashes such as the message authentication with shared secrets known as Hash-based Message Authentication Code (HMAC), in which the message is coming with a shared secret before the hashing.

The strength of the Internet protocol does not have to be necessarily affected by the weakness of the underlaying hash function. Out of mentioned uses of hashes, only the non-repudiable digital signature and the digital signature in certificate are affected by collision attacks [8], and only under limited circumstances. In the following sections we prove this statement by analyzing the impact of hash vulnerabilities on Internet protocols. In particular, we analyze whether the weaknesses of the underlaying hash functions affect the strength of SEND (Secure Neighbor Discovery) protocol. We also suggest the hash agility mechanisms.

3. USE OF HASHES IN SEND

Secure Neighbor Discovery (SEND) enhances the security of the Neighbor Discovery Protocol (NDP) [11]. NDP enables nodes to discover other nodes within the local link, determine their link-layer addresses, find on-link routers, maintain reachability information about the paths to active neighbors [1], and others. The protection shield offered by NDP itself [12] is not enough to counter vulnerabilities because they originate in the lack of authentication, message protection and router authorization capabilities.

Secure Neighbor Discovery (SEND) enhances NDP with additional features: proof of address ownership, message protection, router authorization and successfully counters vulnerabilities. However, it assumes that hash functions are not vulnerable. Each of the features involves the use of hashes, as discussed in further subsections.

3.1 Cryptographically Generated Addresses

The address ownership proof feature is based on IPv6 addresses known as Cryptographically Generated Addresses (CGAs). CGA [2] is a mechanism that binds the public component of a public-private key pair to an IPv6 address. It is generated as a one-way hash of the four input values: a 64-bit subnet prefix, the public key of the address owner, the security parameter (sec) and a random nonce (modifier).

$$CGA(128) = Prefix(64)|IID(64)$$

$$IID(64) = hash(CGA_{parameters})$$

$$CGA_{parameters} = (prefix, pubkey, sec, modifier)$$

The owner of the CGA address sends corresponding CGA Parameters, including all required input data for the CGA generation together with the CGA address to a verifier. The CGA verification consists of the re-computation and comparison of received CGA value based on the received CGA parameters, including the public key.

IPv6 nodes are configured with the CGA using the stateless auto-configuration process. Impacts of collision attacks on current uses of CGAs are analyzed in the update of the CGA specification [3]. The purpose of CGA is to provide the proof-of-ownership of the private key corresponding to the public key used to generate the CGA. Through the secure binding of the public key and the owner's IPv6 address, CGAs ensure that the person that we are talking to is the same as the one that we talked to in the previous message. Since in the beginning we do not actually know who the owner is, because there

is no third party involved to provide this information, CGAs does not provide the non-repudiation protection. Taking into account that the attacker has to generate both the message *m* and the false message *m'*, this confirms the conclusion that CGAs do not deal with the non-repudiation feature. Collision attacks are mainly about affecting the non-repudiation feature. In the collision attack against the CGA both of the CGA Parameters sets are chosen by an attacker, so there is no useful real-word attack that the attacker can perform against CGAs in SEND. Therefore, as [3] points out CGA based protocols, including SEND, are not affected by the recent collision attacks.

3.2 Router authorization

NDP enables stateless auto-configuration of IPv6 hosts based on the information that the host learns from the link local router. The trust between the on link router and the host is achieved through the X.509 Certification Authority. As we noted out in the Introduction, the use of hashes within the digital signature in the certificate has been affected by the recent collision attacks. These vulnerabilities reflect on SEND as well.

A new host on the link can easily configure itself using the information that it gets from the router. The problem here is that there is no way a host can tell from the Neighbor Discovery information whether the router is the authorized router or the bogus one. If the link is unsecured, the router might be a rogue router. At the moment when the host is supposed to validate the router, the host is not able to do so since it is not able to communicate with the off-link hosts. To solve this issue, SEND uses the trust achieved through the third party which is the X.509 Certification Authority [6]. The mechanism is based on the X.509 certification path information that the router sends to the host. The certification path consists of the Router Authorization Certificate, which is X.509 certificate that authorizes a specific IPv6 node to act as a router, followed by intermediate certificates that lead to the trust anchor trusted both by the router and the host. The Router Authorization Certificate contains the information about the prefix that the router is authorized to advertise within the X.509 Extensions for IP Addresses and AS Identifiers [10].

Researchers demonstrated attacks against PKIX certificates with MD5 signature, in 2005 [4] and in 2007 [14]. In 2005 they constructed the original and the false certificate that had the same identity data and the same digital signature, but different public keys [4]. The problem for the attacker is that two certificates with the same identity are not actually useful in real-world scenarios, because the Certification Authority is not allowed to provide such two certificates. In addition, the identity field is one of the human-readable fields and attacks against the human-readable fields demand much more effort than the attacks against non human-readable fields, such as the public key field. In case of the identity field, an attacker is faced with the problem of the prediction and the generation of the false but meaningful identity data, which at the end might be revealed by the Certification Authority. Thus, in practice, in order to lower the probability for the collision attack, non human-readable fields should be used as less as possible. In 2007 researchers demonstrated certificates which differ in the identity data and in the public key, but still result in the same signature value. In such attack, even if an attacker produced such two certificates in order to claim that he was someone else, he would still need to predict the content of all fields appearing before the public key, e.g. the serial number and validity periods. Although a relying party cannot verify the content of these fields (each certificate by itself is unsuspicious), the Certification Authority keeps track of those fields and it can reveal the false certificate during the fraud analysis.

Regarding certificates in SEND, potentially dangerous are attacks against the X.509 certificate extensions. For example, an attack against the IP address extension [10] would enable the router to advertise the changed IP prefix range, although, not broader than the prefix range of the parent certificate in the X.509 certificate chain. A special care has to be taken regarding the attacks against middle-certificates in the certification path, where for the cost of the one false certificate, an attacker launches an attack on multiple routers.

3.3 Message protection

SEND allows the integrity and authenticity protection of the NDP message by inserting two fields into the NDP message: RSA Signature option and Key Hash field. The first one is the digital signature, while the second one is pure hash, a fingerprint. As the previous section denoted, neither the digital signature nor the hash fingerprint does not introduce new vulnerabilities to Internet protocols.

The computation of the Digital Signature field is described in [1]. The digital signature in the RSA Signature option is produced as the SHA-1 hash over the IPv6 header, the ICMPv6 header and the NDP message, that is signed with the sender's private key. The sender's private key corresponds to the public key used in the CGA parameters structure. It is usually authorized through CGAs. Possible attack on such explicit digital signature is a typical non- repudiation attack, i.e. the collision attack. An attacker produces two different messages, m and m', where $hash(m) = hash(m')$. It underlays one of the messages to be signed with the key authorized through CGAs, but uses another message afterwards. If a SEND attacker would manage to find the collision in the message made of the ICMPv6 and NDP fields mentioned in the beginning of the paragraph, he could underlay the false message. The fields that make sense to be changed are NDP fields, as opposite to the ICMPv6 fields which would be easily revealed. However, as denoted in [8], the structure of at least one of two messages (m, m') in a collision attack is strictly predefined. The previous requirement makes the use of digital

signatures within SEND resistant to the real-world collision attacks.

The Key Hash field is a SHA-1 hash of the public key from the CGA Parameters structure. The pre-image attack against the Key Hash field would potentially be dangerous, as in the case of all other hashes in many other Internet protocols, because the Key Hash field contains non human-readable data. However the Key Hash field is not susceptible to the collision attack, since in the collision attack an attacker itself chooses both keys, k and k', where $hash(k)$ must be equal to $hash(k')$. The reason for the former is that the associated public key is already authorized through the use of CGAs, which means that it has to correspond to other SEND fields. Otherwise, the key is outside of the context of the attacked message. This proves that the use of fingerprints within SEND is not susceptible to the recently discovered collision attacks.

4. SUPPORTING HASH AGILITY IN SEND

SHA-1 and MD5 are the most frequently used hash algorithms in Internet protocols and they are also used in the SEND protocol. Both protocols are already affected by collision attacks and in the future they could be affected by first and second pre-image attacks. As opposite to currently know hash collision attacks, pre-image attacks could introduce new vulnerabilities to SEND. Though there is no direct and exploitable vulnerability we assume that attackers abilities are constantly evolving and that future attacks could pose a greater security problem.

We propose methods and guidelines on how to introduce new hash algorithms in SEND along with hash agility. Hash agility enables the improvement and adaptation of new hash algorithms opposed to the currently specified fixed selection of hash algorithms.

In order to increase the future security of SEND, the following approaches can be used [9]:

- The most effective and secure way would be to bind the hash function option with *something* that can not be changed at all, like [3] does for CGA by encoding the hash function information into addresses. There is no possibility to do that in SEND. We could use by default the same hash function in SEND as in CGA. The security of all hashes in SEND depends on CGA, i.e. if an attacker could break CGA, all other hashes are automatically broken. Additionally, if using the hash algorithm from the CGA, no bidding down attacks are possible, but this solution introduces the limitation for SEND to be used exclusively with CGAs.
- Another way to support the hash agility is to incorporate the information about hash algorithms for each hash (in certificate, RSA Signature option, Key Hash field and CGA) within the hash option in the SEND message. The advantage of this approach is the flexibility to offer and choose multiple hash algorithms for different SEND or certificate hash fields. The major disadvantage is the vulnerability to the downgrade attack in which an attacker enforces its peer to use the exploitable hash algorithm by offering him a weak hash algorithm for the mutual intersection.
- The third possible solution is to encode the algorithm in the CGA. However, this will reduce the strength of the CGAs and make them vulnerable to brute force attacks.
- One of the possible solutions is also the hybrid solution, i.e. to require the hash algorithm to be the same as CGA, if CGA option is present, and to use the hash agility option if the CGA option is not present.

4.1 The negotiation approach in SEND hash agility

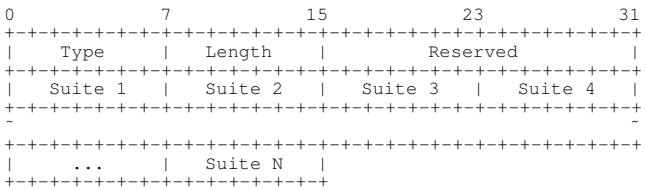
None of the previous solutions supports the negotiation of the hash function. Another possibility is to use a negotiation approach for the SEND hash agility such as the one based on the Supported Signature Algorithm option described in [5]. Based on the processing rules described in [5] nodes find the intersection between the sender's and the receiver's supported signature algorithms set, as well as the preferred signature algorithm. When producing and validating hashes in SEND, nodes must follow these rules [5]:

- In the Authorization Delegation Discovery (ADD) process during router authorization, if any of the certificates in the certification path uses the signature algorithm which is not one of the signature algorithms negotiated in the signature agility process through the use of the Supported Signature Algorithms option, nodes must reject the Router Authorization certificate.
- In order to produce the Digital Signature field, nodes must use the signature algorithm negotiated in the signature agility process through the use of the Supported Signature Algorithms option.
- In order to produce the Key Hash field, nodes must use the hash algorithm associated to the signature algorithm negotiated in the signature agility process through the use of the Supported Signature Algorithms option.

In order to support the hash agility in SEND we recommend the negotiation approach. Cryptographic negotiation can be done either in *a la carte* way or based on compromised suites. *A la carte* negotiation means that each algorithm, in our case the digital signature and the hash field in the SEND message, CGA and the digital signature in the certificate, is negotiated separately. Compromised suites on the other hand offer a certain number of complete cryptographic suites to be inputs for the negotiation procedure (Figure 1).

We also recommend some guidelines for the implementation of the SEND hash agility support. We suggest a couple of compromised suites (Figure 2) for cryptographic negotiation based on the new SEND option within the integrity protected part of the SEND message. The suggested principle has the following strong key aspects:

Fig. 1. SEND hash function option for the hash agility support



Type: NDP option type [RFC3971]
Length: SEND message length
Reserved: For future purposes
Suite: Compromised suite with the list of two
hash algorithms and two digital signatures.

Fig. 2. Compromised suite for the hash function option

Suite 1

Key Hash field:	SHA-256
Digital Signature:	RSASSA-PKCS1-v1_5
Certificate signature:	RSAwithSHA-1
CGA hash algorithm:	SHA-1

Suite 2

Key Hash field:	SHA-384
Digital Signature:	RSASSA-PSS
Certificate signature:	RSAwithSHA-1
CGA hash algorithm:	SHA-256

Suite 3

Key Hash field:	SHA-512
Signature field:	RSASSA-PSS
Certificate signature:	DSAwithSHA-1
CGA hash algorithm:	SHA-384

- It avoids the interoperability issues by offering multiple, different hash algorithms for each hash used in SEND fields, CGA or X.509 certificate. IPv6 hosts that are talking to each other would in such way negotiate each hash algorithm separately, as well as each digital signature algorithm.
- Even though the downgrade attacks always remain possible to some extent, placing the hash function option under the coverage of the integrity protection would lower the possibility for the downgrade attacks.
- Compromised suites avoid the exponential expansion of proposal that happen in case of *a la carte* cryptographic negotiation and make the solution easier to understand, configure and implement.
- This solution enables the specification of new suites in the future. This way new hash and signature algorithms can be easily introduced in the hash agility process, thus improving the security of SEND.

5. CONCLUSION

We analyzed uses of hash functions in Secure Neighbor Discovery (SEND). SEND uses both raw hashes, digital signatures, as well as X.509 certificates within the SEND message fields (RSA Signature option, Key Hash field), IPv6 CGA addresses and X.509 certificates used in router authorization process.

Our analysis concluded that there are only two uses of hashes that are affected by recent collision attacks - non-repudiable signatures and digital signatures in certificates. Since none of the hashes at the Internet layer is used in non-repudiation purposes, and all demonstrated collision attacks are only about non-repudiation, SEND is not affected by the reduced hash collision resistance. Attacks against certificates are lingered due to the use of non human readable fields in that context. However, collision attacks, are possible, the techniques of which can only be expected to improve, not worsen. Computational power is also rapidly increasing. Therefore, we provided the survey that analyzes hash attacks, the use of hashes within SEND and evaluates the impacts of hash attacks on hashes used in SEND. We also evaluated possible approaches to allowing hash agility and described a possible solution for SEND hash agility based on a negotiation approach.

6. ACKNOWLEDGMENTS

We would like to thank Suresh Krishnan and Sheng Jiang (the co-authors of RFC6273 [9]) and Marcelo Bangulo Braun for the valuable comments on supporting the hash agility in SEND.

REFERENCES

[1] J. Arkko, J. Kempf, B. Zill, and P. Nikander. SEcure Neighbor Discovery (SEND). RFC 3971, Mar. 2005.
[2] T. Aura. Cryptographically Generated Addresses. RFC 3972, Mar. 2005.
[3] M. Bagnulo and J. Arkko. Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs). RFC 4982, July 2007.
[4] S. Bellovin and E. Rescorla. Deploying a New Hash Algorithm, 2005. November 2005.
[5] T. Cheneau, M. Laurent, S. Shen, and M. Vanderveen. Signature Algorithm Agility in the Secure Neighbor Discovery (SEND) Protocol. draft-cheneau-csi-send-sig-agility-02, June 2010.
[6] D. Cooper, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List. RFC 5280, Mar. 2005.
[7] H. Dobbertin. Cryptanalysis of MD4, 1996. Third Workshop on Cryptographic Algoruthms, Cambridge 1996. Lecture Notes in Computer Science, pages 55-72.
[8] P. Hoffman and B. Schneier. Attacks on Cryptographic Hashes in Internet Protocols. RFC 4270, Nov. 2005.
[9] A. Kukec, S. Krishnan, and S. Jiang. The Secure Neighbor Discovery (SEND) Hash Threat Analysis. RFC 6273, June 2011.
[10] C. Lynn, S. Kent, and K. Seo. X.509 Extension for IP Addresses and AS Identifiers. RFC 3779, June 2004.
[11] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 4861, Sept. 2007.
[12] P. Nikander, J. Kempf, and E. Nordmark. IPv6 Neighbor Discovery (ND) Trust Models and Threats. RFC 3756, May 2004.
[13] B. Schneier. Applied Cryptrography, 1996. Second edition, John Wiley & Sons.
[14] M. Stevens, A. Lenstra, and B. Weger. Chosen-Prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identities, 2007. EUROCRYPT 2007: 1-22.
[15] X. Wang, Y. Yin, and H. Yu. Finding Collisions in the Full SHA-1, 2005. CRYPTO 2007: 17-36.