Hindawi
Security and Communication Networks
Volume 2018, Article ID 1816462, 11 pages
https://doi.org/10.1155/2018/1816462

Research Article

Towards a New Algorithm to Optimize IPv6 Neighbor Discovery Security for Small Objects Networks

Ali El Ksimi and Cherkaoui Leghris

RTM Team, L@M, Faculty of Science and Technologies Mohammedia, University of Hassan II Casablanca, Morocco

Correspondence should be addressed to Ali El Ksimi; ali.elksimi@yahoo.fr

Received 28 December 2017; Revised 20 April 2018; Accepted 30 April 2018; Published 6 June 2018

Academic Editor: Wenjia Li

In order to verify the uniqueness of link-local or unicast addresses, nodes must perform a Duplicate Address Detection process before using them. However, this process is subject to many attacks and the security is willing to be the most important issues in Small Object Networks with IPv6. In this paper, we developed a new algorithm to optimize the security in IPv6-DAD process; this method is based on SHA-512 to verify the identity of the Neighbor Discovery messages transmitted in the link local. First, before sending the NS message, the new node uses the function SHA-512 to hash to the target address and use the last 64 bits in a new field and then encrypt the result with its private key. When receiving the secure message, the existing nodes decrypt it. Our algorithm is going to secure the DAD process by using a digital signature. Overall, this algorithm showed a significant effect in terms of the Address Configuration Success Probability (ACSP).

1. Introduction

The network protocol mainly used today for Internet communications is the Internet Protocol (IP). The IPv4 protocol suffers from many weaknesses such as the insufficient address space nowadays. Indeed, the IPv4 addresses are 32 bits long, which represents about 4,3 milliard of possible IPv4 addresses. Following the explosion of network growth Internet and wastage of addresses due to the class structure, the number of IPv4 addresses has become insufficient. Another problem is the saturation of the routing tables in the main routers of the Internet. Although since 1993, many emergency measures have been taken, this only allows delaying its deadline. So, the Internet Engineering Task Force (IETF) launched work in 1994 to specify the Internet Protocol that will replace IPv4: this protocol is IPv6 [1].

The Neighbor Discovery (NDP) [2] is the most important part in IPv6; it allows a node to integrate into the local network environment in which IPv6 packets are physically transmitted. Through to this protocol, it becomes possible to interact with the equipment connected to the same support (stations and routers). It is important to note that for a given node, the neighbors discovery does not consist in establishing an exhaustive list of the others connected to the link. Indeed, it is only to manage those with whom it dialogues. This protocol performs the following functions: Address Resolution, Neighbor Unreachability Detection, Autoconfiguration, and Redirect Indication. It uses five messages including Router Solicitation, Router Advertisement, Neighbor Solicitation, Neighbor Announcement, and Indication redirection. The IPv6 StateLess Address AutoConfiguration (SLAAC) [3] of IPv6 is primarily based on the NDP process. This mechanism uses Duplicate Address Detection (DAD) [4] to verify the uniqueness of the addresses on the same link. However, it is vulnerable to attack and many solutions have been standardized to minimize this vulnerability such as SEcure Neighbor Discovery (SEND) [5], but they are subject to certain limitations.

We base our study on SmObNet6 (Small Objects Network with IPv6) which is a generic term used to define either small or larger network to connect small communicating objects. The use of IPv6 protocol regarding communication, collecting, and exchanging data between objects represents a common point between these networks within the Internet infrastructure. This paper treats the SLAAC phases and explains the problems associated with them. So, we propose

a new algorithm based on SHA-512 [6] in order to optimize IPv6 security in SmObNet6,

This paper is organized as follows: Section 2 presents a related work to our field when Section 3 describes some IPv6 functionalities, in particular, the DAD process. Section 4 shows the parameters and methodology following in this work and we present our algorithm in Section 4. Section 5 includes the algorithm implementation with evaluation, after we conclude this paper and addresses some prospects.

2. Related Work

Attacks on the IPv6 operations, especially on DAD process, become one of the interesting research fields. Several proposals have been made by researchers to address security issues in IPv6 DAD. Many authors have treated this problem.

In [7], the authors have proposed a scheme to secure IPv6 address which includes the modifications to the RFC 3972 standard by reducing the granularity factor of a sec from 16 to 8 and replacing RSA with ECC and ECSDSA, using SHA-256 [6] hash function. This method improves the address configuration performance, but it does not eliminate the address conflict.

In [8], the authors have presented a new algorithm for address generation. This mechanism has a minimal computation cost as compared to CGA. Nevertheless, this mechanism uses SHA-1 hash encryption which is vulnerable to collisions attacks.

In [9], the authors have utilized a novel approach for securing IPv6 link-local communication. They have used an alternative approach for the CGA and SEND protocols which still represent a limitation to the security level.

Another approach such as secure IPv6 address configuration protocol for vehicular networks [10] was proposed to ensure security in IPv6 without DAD process. However, this method is used only when the distance between a vehicle and its serving AP is one-hop.

In [11], the authors have proposed a new method to secure Neighbor Discovery Protocol in IPv6. This mechanism is based on SDN controller to verify the source of NDP packets. However, this method is not efficient because it does not handle the detection of NDP attacks.

Another method was used in [12] to secure the DAD; it is called trust-ND. It is used to detect fake NA messages. However, the experiments show some limits of this method.

In [13], the authors have presented a technique for detecting Neighbor Solicitation spoofing and advertisement spoofing attacks in IPv6 NDP. However, this method can only detect NS spoofing, NA spoofing, and DoS attacks. The disadvantage of this method is that it does not detect other attacks like Duplicate Address Detection attacks.

In [14], the authors have proposed a new method to secure NDP attacks; this method is based on the digital signature. It detects the messages NS and NA spoofing and DoS attacks, router redirection, and Duplicate Address Detection, but this mechanism is not complete.

In [15], the authors describe and review some of the fundamental attacks on NDP, prevention mechanisms, and current detection mechanisms for NDP-based attacks.

In this paper, we propose to study and evaluate the security in the NDP within the network based on IPv6 protocol. Indeed, we suggest a new algorithm which could secure the attacks in the DAD process. The results showed that DAD process could be optimized by introducing a new field in the NS and NA messages; the hash of the new node's target address. Overall, this method showed a significant effect in terms of time and computation.

3. Features of IPv6 NDP: Duplicate Address Detection

The Neighbor Discovery Protocol (NDP) mechanism provides IPv6 with some number of features essential for the proper IPv6 protocol functioning. The best known is the address resolution feature that matches what is ARP in IPv4. This protocol also offers other features. The one that will interest in our paper, Duplicate Address Detection (DAD), allows detect when two nodes want to use the same address and avoids the future collision by refusing the assignment of the address. This is equivalent to "gratuitous ARP" in IPv4. This feature is even more important, that, in IPv6, new nodes can use the "stateless autoconfiguration" and assign themselves an address (self-generated).

3.1. DAD Process. The Duplicate Address Detection mechanism applies to all type addresses unicast before they are assigned to network interfaces, regardless of whether they are manual, stateless, or stateful. This feature can still be disabled by system administrators.

The Neighbor Discovery Protocol mechanism uses ICMPv6 type messages [2]. Under the DAD mechanism, we are only interested in two types of messages, the Neighbor Solicitation (NS) and the Neighbor Advertisement (NA). When resolving an address, the message Neighbor Solicitation is used to request the physical address of a node (e.g., MAC address) it wants to communicate by contacting it via IPv6 address. This message contains a target field that is populated with the node's IPv6 address that we want to contact. If this target exists, it responds with a message to the request sending node and contains, in one of its fields, an option with the physical address of this node regarding the network interface concerned. This association between the logical address and the physical address will then be kept in the neighbor cache table.

For the DAD mechanism, this request/response exchange is used more finely. The node does not appropriate the address it desires until the procedure has been completed satisfactorily; during this procedure, this address will be called "tentative". To be more precise, if the node receives traffic destined for a "temporary" address, it must not process it or respond to it. The procedure is to issue a Neighbor Solicitation message with as target its "temporary" address and in source address, the address with type "unspecified" (: :). If someone answers, to this NS message with a Neighbor Advertisement message means that the address is already taken and a node already has this address, it is considered that the attempt to obtain an address fails: the node cannot

get this address. There is no other attempt to get this address; the administrator must intervene on the node to configure it with another address. There is another case where we cannot get the address: when the node receives a message NS with as target address the "temporary" address that it wants to use. This means that another node also performs a DAD procedure for the same address. In this case, neither of the two nodes performing the DAD mechanism on the same address will be able to obtain it.

The DAD mechanism is not infallible, especially if it occurs during the time when several nodes of the same network are temporarily "separated" (loss of connecting or dropping a link between the nodes) and that one or more of the nodes perform a DAD procedure. They can assign the same address without the collision detection procedure.

3.2. The Algorithm of DAD Process. For the node, the procedure starts by listening to the multicast group "all-nodes multicast" and the multicast group of the solicited-node ("solicited-node multicast"). The first allows it to receive address resolution requests ("Address Resolution") for this address and the second will allow it to receive the messages sent by other nodes also making a DAD on this address. In order to listen to these, the node must send a Multicast Listener Discovery (MLD) [16] request; when a node triggers the DAD procedure, it sends a Neighbor Solicitation message, an ICMPv6 type message.

The header IPv6 contains the following fields:

(i) The source address of the IPv6 packet is the unspecified address (: :).

(ii) The destination address is the multicast address of the solicited-node ("Solicited-Node Multicast Address") of the "tentative" address, that is the last three octets of the provisional address concatenated with the prefix FF02:: 1: FF00: 0/104.

When sending this NS message, we observe for the ICMPv6 header:

(i) The target address field is filled with the "tentative" address.

(ii) The link layer option of the source is not used. So, two nodes can send the same identical NS message.

With stateless autoconfiguration, it is important to note that if the DAD mechanism fails, then there is no further testing and a new address will have to be assigned otherwise, in particular, for addresses that will have been built automatically via the modified EUI-64 format.

The algorithm DAD is described as follows:

(i) The first step is to generate an IPv6 address with either autoconfiguration or other methods.

(ii) In the second step, the node will be subscribed in multicast groups: all multicast nodes and solicited multicast node.

(iii) After, there are three cases:

(a) A NA message is received: the tentative address is used as a valid address by another node. The tentative address is not unique and cannot be retained.

(b) A NS message from a neighbor is received as part of a DAD procedure; the tentative address is also a tentative address for another node. The tentative address cannot be used by any other node.

(c) Nothing is received after one second (default value): the tentative address is unique, it passes from the provisional state to a valid one, and it is assigned to the interface.

Figure 1 shows the DAD algorithm.

3.3. The Attack on DAD Process. An attack on the DAD mechanism was identified in [10]. The attack is composed as follows: the attacker will deceive the DAD mechanism and make it succeed in one of the two cases where it fails so that the victim cannot claim an address. Since there is a finite number of tries to get an address, the DAD always ends up failing; it is a DoS attack [11]. For the attack to be feasible, the attacker must be able to listen on the network to any query necessary to perform the DAD procedure, e.g., the NS messages with the unspecified address as the source address is characteristics of the DAD procedure; this implies being able to join the multicast group "Solicited-Node". He then has two choices; he can send a NS message with, as source address, the unspecified address and, as the target address, the address of the victim or an NA message with, as the target address, the "tentative" address of the victim. It can thus prevent the arrival of new nodes having no address yet. The attack effectiveness depends strongly on the type of links, because it is necessary that the attacker can receive the first NS sent by the victim and that he can answer them. Indeed, the attacker must to be able to join the multicast group "Solicited-Node", which is not easy in the case of a level 2 point-to-point technology, for example, ADSL.

3.4. Vulnerabilities of Multicast Communications. In IPv6 multicast (DAD process), groups are identified by a group address and any node in the network can join or leave the group when it wishes. This simplicity, which is the power of multipoint routing, presents however many vulnerabilities such as

(i) IPv6 multicast does not support the notion of the closed group. Indeed, multicast addresses are public: joining a group or leaving a group is an operation that does not require special permissions. This allows any node to join a group and receive messages for it.

(ii) Access to the group is not controlled: an intruder can send data to the group without being part of it, disrupt the multipoint session, and possibly cause congestion in the network.

(iii) The data intended for the group can cross several unsecured channels before reaching all members of

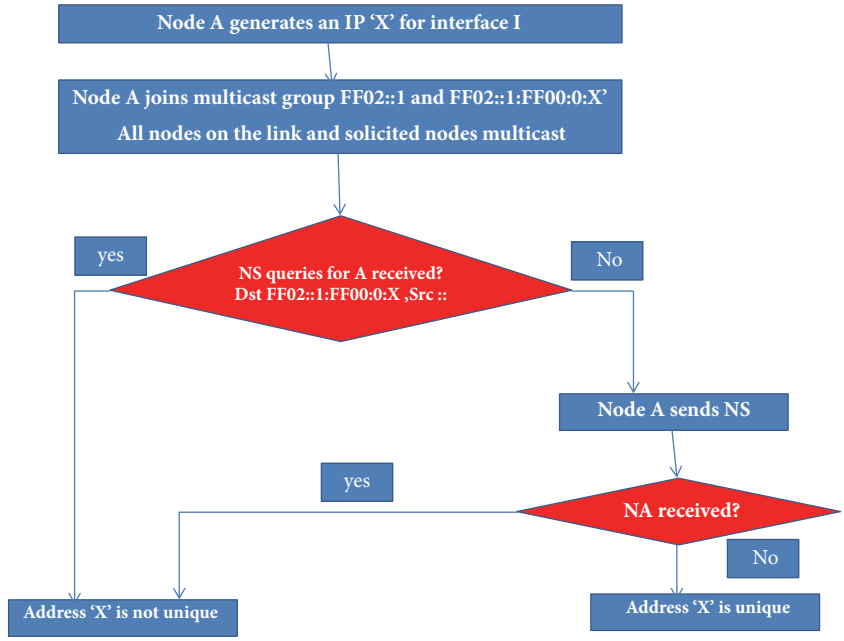

Figure 1: The flowchart of the DAD process.

the group. This increases listening opportunities to potential intruders.

(iv) Group communications offer more opportunities for intercepting communications, proportional to the number of participants.

(v) A vulnerable point in the group implicates the safety of all members of the group.

(vi) The large-scale publication of the group's identity and address helps intruders focus their attacks.

(vii) Attackers can impersonate the legitimate members of the group.

To counteract these attacks, group communication requires security services such as authentication, data privacy, and confidentiality of the traffic flow.

3.5. Security Needs in Multicast. Multicast requires the set of security mechanisms in a unicast communication in addition to some needs inherent to its nature which is the group communication. These needs can be divided into three main parts.

3.5.1. Authentication. All participants in a multicast session must self-authenticate before joining the group. Authentication [17] may be restricted to group members such as sources and receivers or possibly extended to the routing infrastructure like designated routers.

Among other authentication mechanisms, the certification scheme with a third authority can be used.

3.5.2. Integrity. This ability ensures that the multicast stream reaches the recipients without falsification. This option is usually provided by cryptographic, hash, and digital signature mechanisms [18].

3.5.3. Confidentiality. This confidentiality [19] must be provided at several levels:

(i) Past privacy (backward confidentiality): we can imagine that a hacker can store the multicast stream for a time interval $[t_0, t]$ and join the group at time t to acquire the keys needed to decrypt this stream "past". Past privacy alters such a hacking scheme by (for example) modifying decryption keys for the stream, once a new member joins the group.

(ii) Forward confidentiality: a system with this ability prevents any member excluded from the multicast group at time t from having the keys necessary for decrypting the multicast stream at times $t + \mu$. This usually results in a modification of these keys and then their redistribution to the remaining members.

(iii) Group privacy: only authenticated members must have the keys to decrypt multicast messages.

4. Proposed Algorithm Model

In this section, we present the description of our algorithm which makes it possible to secure the target address used in NS message.

4.1. Digital Signature. A digital signature must prove the identity of the issuer of NS or NA messages and guarantee nonrepudiation. The RSA cryptosystem also allows the signing of a message. Indeed, by inverting the mechanisms that are to say that the decryption of the message, which is only

accessible to those who know the factorization of the module, becomes the signature process. On the other hand, since encryption is public, by encrypting the signature produced, everyone must fall back on the message. The ability to decrypt with RSA proves the knowledge of the private key.

4.2. Hash Function. A hash function [20] is a method for characterizing information, a data. By having a sequence of reproducible treatments at an input, it generates a fingerprint to identify the initial data.

A hash function, therefore, takes as input a message of any size, applies a series of transformations, and reduces this data. We get at the output a string of hexadecimal characters, the condensed, which summarizes somehow the file.

We define a hash function as an application:

$$\mathbf{h} : \{\mathbf{0}, \mathbf{1}\}^{*} \rightarrow \{\mathbf{0}, \mathbf{1}\}^{\mathbf{n}}, \quad n \in N. \tag{1}$$

A hash function is considered safe if the following three properties are satisfied:

(a) Resistance to a preimage attack (one-way). For any given output y, finding an x, which makes h(x) = y, is computationally infeasible.

(b) Resistance to a second preimage attack. For any given input x, finding an input x' that is unequal to x, which makes h(x) = h(x'), is computationally infeasible.

(c) Resistance to a collision attack. Finding two unequal inputs x and x0, such that

h(x) = h (x'), is computationally infeasible.

The SHA Secure Hash Algorithm [21] is a hash algorithm used by certificate authorities to sign certificates and CRL (certificate revocation list). Introduced in 1993 by the NSA with the SHA0, it is used to generate unique condensates (thus for "chopping") of files.

4.3. The Structure of SHA1 and SHA512. Table 1 shows the characteristics of SHA-1 and SHA-512.

4.4. Hash-TargetAdd-DAD. Hash-TargetAdd-DAD (Hash target address) is a new definition of the ICMPv6 packet (for NS and NA).

Since the standard DAD is not secure, in order to fulfill such security requirement, a "Hash Secure Target" can be applied on NS and NA messages to ensure that only nodes which possess this hash are able to communicate in the IPv6 local network.

Figure 2 shows the message format of Hash-TargetAdd-DAD.

The message format of Hash-TargetAdd-DAD is illustrated in Figure 3 Hash-TargetAdd-DAD uses two new message types, namely, $NS_{\text{hash-targetAdd-DAD}}$ and $NA_{\text{hash-targetAdd-DAD}}$, and its "Type" fields are 138 and 139, respectively. Compared with the NDP packet, Hash-TargetAdd-DAD adds a new field "Hash_target_64", which stores the last 64 bits of the SHA-512 result.

The hash_target_64 calculation method is illustrated in Figure 3.

Table 1: The characteristics of SHA-1 and SHA-512.

	SHA1	SHA512
Message size	2^{64} bits maximum	2^{128} bits maximum
Block size	512 bits	1024 bits
Word size	32 bits	64 bits
Size of digest	160 bits	512 bits
Security level	Collision in 263 operations	2^{128} bits

4.5. Secure Target Generation and Matching Process. Each node including the new one begins by generating two public and private keys. Before sending the message, it will encrypt the NS with the private key and then multicasts its public key. When receiving an encrypted message, the receivers will decrypt the NS message with the received public key.

In the case where the receiving node wants to send an NA message, it will encrypt it with its private key and send the encrypted message and the public key to the new node. The new node will decrypt the NA message with the public key sent.

Public and private keys are generated using the RSA algorithm [22].

Rule. If the secure target is the same, then the NS is authentic; else drop the message.

(i) NS message sending step:

Before the new node sends the NS message, it proceeds as follows:

(ii) First, it generates the target address fingerprint using an SHA-512 hash function and it extracts the 64 bits from the result of E_s:

E_s =SHA-512 (target address), where E_s is target address fingerprint

E_{64} = Hash_target_64.

(iii) Then, it encrypts E_s with its private key:

Signature (Target address) = C (E_{64}), where C is an RSA encryption function using the new node private key.

Figure 4 shows the mechanism of the secure NS message sending encrypted with the private key using RSA signature.

(i) NS message receiving step:

First, upon receipt of the secure NS, the existing will decrypt it with the public key of the new node (generated by the RSA algorithm). Then, it generates the fingerprint of its IPv6 address, using the same hash function as the new node (SHA-512). Finally, it compares the generated fingerprint and that resulting from the signature.

If both fingerprints are identical, the signature is validated. We are therefore sure that

(ii) This is the new node that sent the NS message.

(iii) The NS message has not changed since the new node signed it.

Ethernet header	Dest MAC Src MAC Type
IPv6 header	Src address Dest address Next header
ICMPv6 header	Type Target address Options icmpv6 Hash_target_64

Figure 2: The message format of Hash-TargetAdd-DAD.

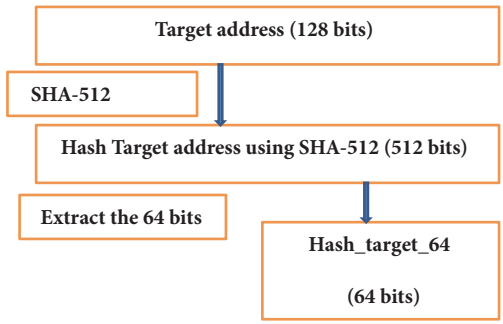

Figure 3: Message format of Hash-TargetAdd-DAD.

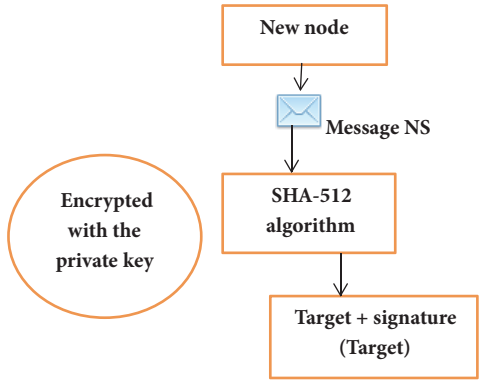

Figure 4: The secure NS sent.

In other words, if D ($C(E_{64})$) == E_{64} (IPv6-existing node)

Where D is an RSA decryption function using the new node public key.

Figure 5 shows the mechanism of the secure NS message receiving decrypted with the public key of the new node using RSA signature.

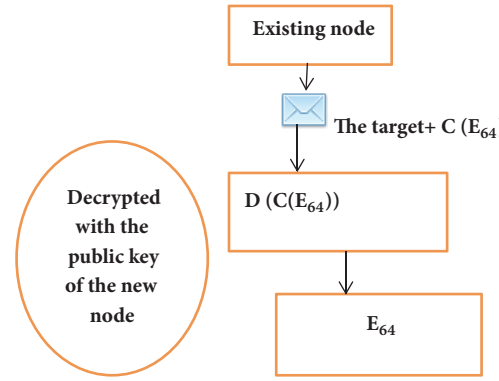

Figure 5: The secure NS received.

4.6. Algorithm HSEC-Target-DAD (HASH Secure Target Address in DAD Process). The steps of our algorithm are defined as follows:

(i) First, the new node generates an IPv6 (the tentative address).

(ii) The new node uses a hash function to hash the target address with SHA-512.

(iii) It extracts the last 64 bits from the hash.

(iv) The hash-64 will be appended to NS message.

(v) The hash-64 is encrypted by the new node private key and it is sent to multicast address FF02::8 instead of all-nodes solicited multicast group FF02::1 to exclude an attacker who can join the group FF02:: 1 (all nodes of the network).

(vi) Existing nodes will decrypt the received NS message with public key new node and match its generated hash secure target with sender's hash secure target.

(vii) If the sender and a receiver hash secure target match, then the verification of the IP address will take place; otherwise, if no match of hashTag is found the receiver

will discard the NS message and add the MAC address into the blacklist.

(viii) The existing node will check its IPv6 address with the new node target address if the match of hash target.

(ix) If the match of duplicate IPv6 address is found, receiving node will reply with NA message appended with the hash target and encrypted by the private key of the existing node. However, if the target address is found unique it creates an entry in its neighbor cache table in order to maintain and update the table for future communication.

(x) When receiving the NA message, the new node will decrypt it with the public key of existing node. If the match of the hash secure target is found then it performs new DAD process, else it will simply discard the message and add the MAC address to the blacklist.

Our algorithm (Figure 6) is based on target address hashing using the SHA-512 function, then we extract the 64 bits of the result, and we encrypt the NS message with a private key new node; this key is generated from the RSA algorithm [23]. Upon receipt of the secure NS message, the receiver uses the public key new node also generated from the RSA algorithm, to decrypt the received message.

Figure 6 shows the flowchart of our algorithm.

5. Implementation and Evaluation

5.1. Network Topology. The network environment includes a gateway router, an Ethernet switch, a new node (MN1), two existing nodes (MN2 and MN3), and an attacker. Figure 7 shows the network topology. The simulated network is a LAN network.

Each node can have several addresses and centralized random address space to increase the probability of address conflict.

5.2. Simulation Results and Evaluation. In our simulation, we define the following performances:

(i) Address Configuration Failure Probability (ACFP): when a mobile node uses DAD process to configure its address in the presence of an attack. If a DAD process (DAD-P) is performed y times, and x times have failed, then the ACFP of DAD-P is

$$ACFP = \left(\frac{x}{y}\right) \quad (2)$$

So, since Address Configuration Success Probability (ACSP) is the complement [24] of ACFP then it is defined as follows:

$$ACSP = 1 - \left(\frac{x}{y}\right) \quad (3)$$

From the definition of ACSP, we can conclude that if ACSP is equal to 0, it means that the DAD-P is failed y times, then the attack is fully functional in DAD-P. Thus, we can use the ACSP to measure a DAD-P.

The simulation includes two scenarios. Scenario 1 simulates DAD and HSEC-Target-DAD with the occurrence of an attack node.

(i) *Scenario 1*: simulation of DAD and HSEC-Target-DAD with the occurrence of an attack node.

(a) Results analysis: the simulation results are presented in Figure 8. The results of the simulation show when there is an attacker in the network, with the standard DAD, the configuration of the address generated to the new node fails, which shows that ACSP tends to 0. However, with our algorithm, the attacker cannot decrypt the sent message because he does not have the private key, which shows that ACSP tends to 1.

We can see in the figure that ACSP of HSEC-Target-DAD is higher than DAD.

(ii) *Scenario 2*: simulation of pseudocollision attacks and SLAAC attacks against HSEC-Target-DAD.

(a) Pseudocollision attacks: before using a pseudocollision attack [25], we first need to define how a hash function internally works.

Most hash functions are basically composed out of four functions:

(i) The first function is called an initialization function; it just sets a bunch of start values for the state: $I:\emptyset \rightarrow \{0,1\}^k$

(ii) The second function is called an input preprocessing function; it computes some values based on the message and possibly hidden context: $P:\{0,1\}^l \rightarrow \{0,1\}^q$

(iii) The third function is called a state-update function, sometimes also called "compression function"; it takes the current message block, the associated preprocessing, and the current state and outputs a new state: $U:\{0,1\}^l \times \{0,1\}^q \times \{0,1\}^k \rightarrow \{0,1\}^k$

(iv) The fourth function is called an output function; it takes the state and outputs the hash digest: $O:\{0,1\}^k \rightarrow \{0,1\}^o$

Now a normal collision attack takes the standard composition of these functions and tries to find a collision.

A pseudocollision attack on the other hand just tries to find a collision on the state-update function. So an attacker is interested in finding two triples $x= (m,p,h), x' = (m',p',h')$ such that $U(x)=U(x')$ with $x \neq x'$.

Pseudocollision attacks: this method attempts to search for one or more collision addresses (the IPv6 address with a hash value whose last 64 bits are the same as that in the "Hash_target-64" field) after the attack node receives NS. Then, a number of $NA_{\text{hash-targetAdd-DAD}}$ is sent to increase the probability of a successful attack.

SLAAC: in SLAAC attack [26], the node can obtain an IPv6 address by combining its own MAC address and network prefix according to "EUI-64." Thus, the attack node can

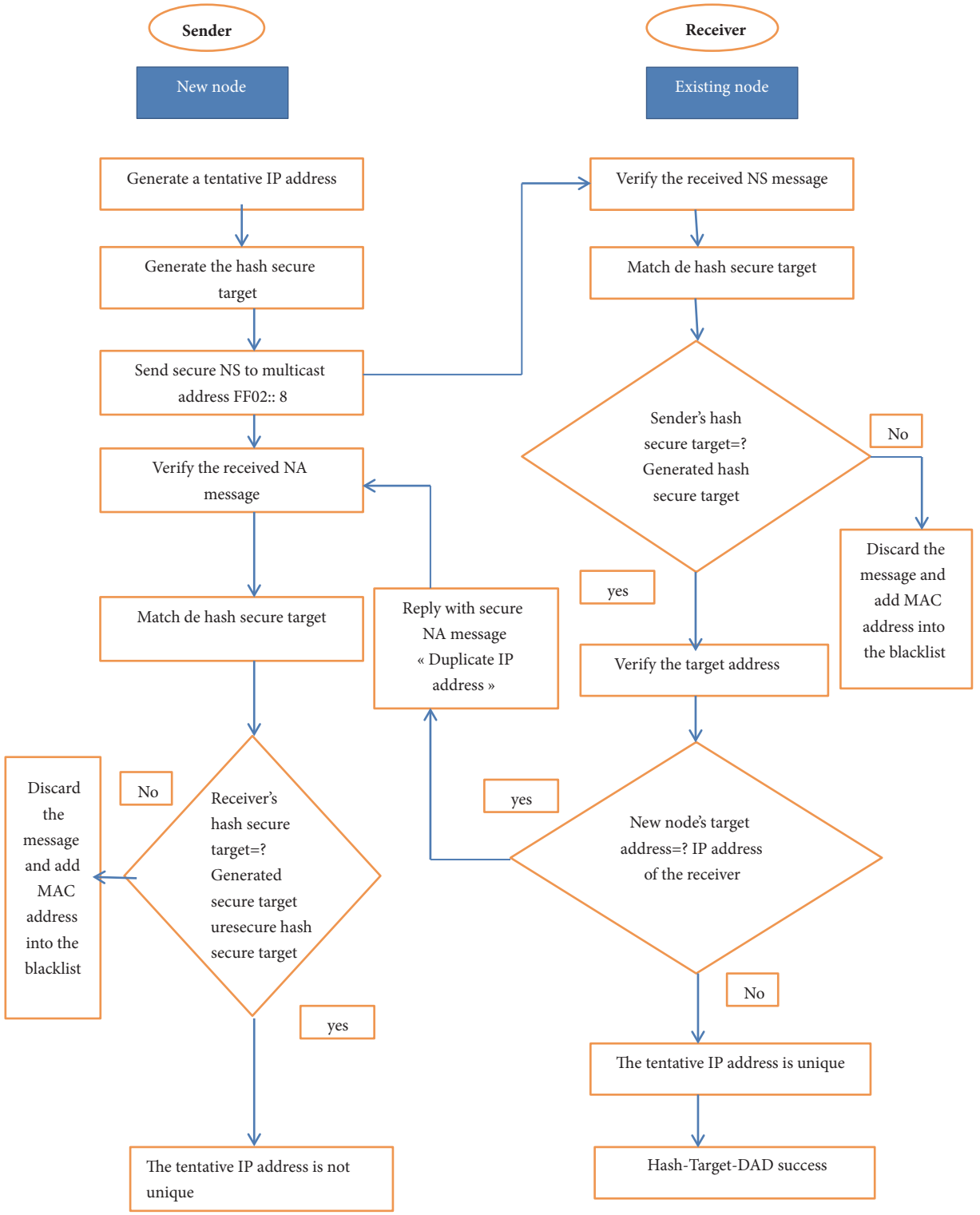

Figure 6: The flowchart of the proposed algorithm.

use the characteristics of SLAAC by combining the network prefix and source MAC address in the $NS_{hash\text{-}targetAdd\text{-}DAD}$ to infer the destination address of DAD.

We set DAD process to 10 seconds. The simulation results are shown in Figure 9. For pseudocollision attack, although the address space is 2^{32} and the attack node has 10 seconds to seek all collisions, the preimage is difficult to find by the attacker as shown in Figure 9. For a SLAAC attack, the address is formed by EUI-64 method [27]. Using this method, the attack node can attack the network; thus, the ACSP is

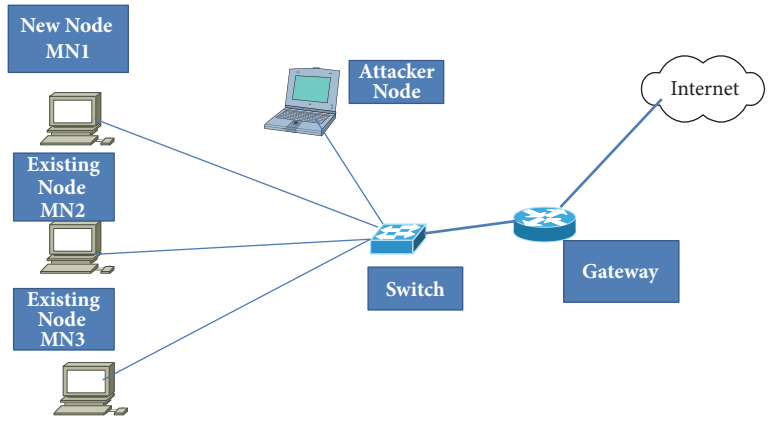

Figure 7: The network topology.

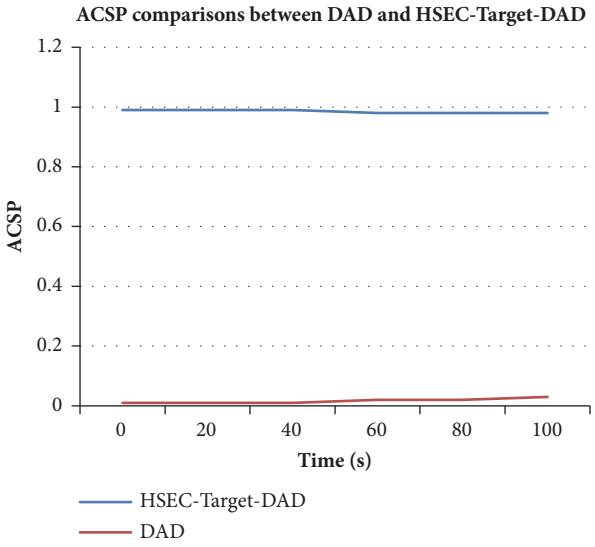

Figure 8: ACSP comparisons between DAD and HSEC-Target-DAD.

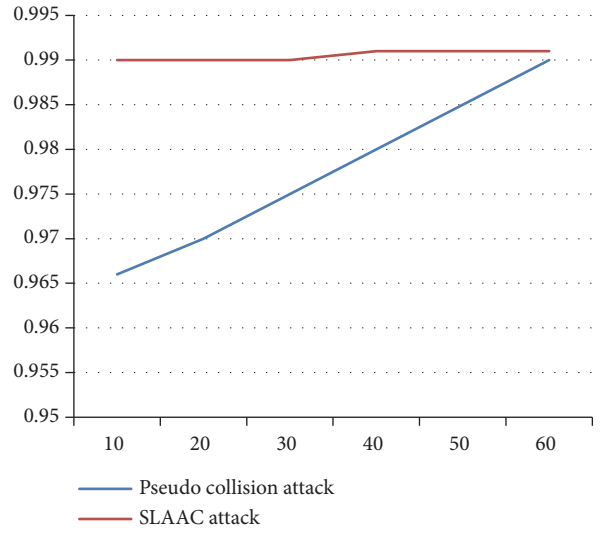

Figure 9: ACSP comparisons between pseudocollision attack and SLAAC attack.

considerably low during the early stage of the simulation. Then, the implementation of the blacklist in the algorithm will have its effect.

When DAD process is failed, the construction of the ID (64 bits) is done randomly. Thus, SLAAC attack does not work anymore, and the ACSP of the subsequent HSEC-Target-DAD process gradually increases and approaches to the ACSP of pseudocollision attack.

The effectiveness of our algorithm:

(i) CGA use SHA1 as a hash function; however in this paper, we use SHA512.

(ii) SHA512 is faster when the size of input data is large, in our case; the size of the target address is 128 bits.

(iii) Another effectiveness of our algorithm is that it uses asymmetric encryption to sign messages.

(iv) Hash_target_64 field can effectively prevent attacks.

6. Conclusion and Perspectives

In order to ensure that all configured addresses are likely to be unique on a given IPv6 link, the nodes execute a Duplicate Address Detection algorithm. Nodes must execute the algorithm before assigning addresses to an interface.

For security reasons, the uniqueness of all addresses must be verified prior to their assignment to an interface. The situation is different for addresses created by stateless automatic configuration. The uniqueness of an address is determined primarily by the portion of the address formed from an interface ID. Therefore, if a node has already verified the uniqueness of a link-local address, we do not need to test the additional addresses individually. The addresses must be created from the same interface ID. All manually obtained addresses must be individually tested to ensure their uniqueness. System administrators at some sites believe that the benefits of Duplicate Address Detection are not worth the overhead they use. For these sites, the use of Duplicate

Address Detection can be disabled by setting an interface configuration flag.

In this paper, we have developed a new algorithm to secure the DAD process in IPv6 network for the small objects in an IPv6 network. This method is based on the security of NS and NA messages. First, before sending the NS message, the new node uses the hash function SHA-512 to hash to the target address and extracts the last 64 bits and then encrypts the result with the public key sent by the initiator of the multicast group FF02::8. When receiving the secure message, the existing nodes decrypt it with its private key.

Then, a hash check must be done; if the hashes are the same, the verification of the IP addresses can be done; otherwise, the message will be deleted.

The underlying cryptosystem, used to generate the public and private key, is RSA algorithm. We used this algorithm for signing the sent message.

The simulation results show that our algorithm has a higher Address Configuration Success Probability than the standard DAD process.

Although IPv6 node communications are limited to NDP and DAD protocols when IPv6 is not officially deployed, there are still attacks that can affect network performance by exploiting only these two protocols as we have been able to study. Our future work will be focalized on router discovery security.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification," RFC Editor RFC8200, 2017.

[2] A. S. A. M. S. Ahmed, R. Hassan, and N. E. Othman, "IPv6 neighbor discovery protocol specifications, threats and countermeasures: a survey," *IEEE Access*, vol. 5, pp. 18187–18210, 2017.

[3] F. Gont, A. Cooper, D. Thaler, and W. Liu, "Recommendation on stable IPv6 interface identifiers," RFC Editor RFC8064, 2017.

[4] F. Alisherov and T. Kim, "Duplicate address detection table in IPv6 mobile networks," in *Advanced Communication and Networking*, C. C. Chang, T. Vasilakos, P. Das, T. Kim, B. H. Kang, and M. K. Khan, Eds., vol. 77 of *Communications in Computer and Information Science*, pp. 109–115, Springer Berlin Heidelberg, Berlin, Germany, 2010.

[5] M. Moslehpour and S. Khorsandi, "A distributed cryptographically generated address computing algorithm for secure neighbor discovery protocol in IPv6," *International Journal of Computer and Information Engineering*, vol. 10, no. N6, 2016.

[6] C. Dobraunig, M. Eichlseder, and F. Mendel, "Analysis of SHA-512/224 and SHA-512/256," in *Proceedings of the 21st International Conference on the Theory and Application of Cryptology and Information Security*, Advances in Cryptology – ASIACRYPT 2015, Auckland, New Zealand.

[7] J. L. Shah and J. Parvez, "IPv6 cryptographically generated address: analysis and optimization," in *Proceedings of the AICTC '16 Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, vol. 13, 2016.

[8] J. L. Shah and J. Parvez, "Optimizing security and address configuration in IPv6 SLAAC," in *Proceedings of the 11th International Conference on Communication Networks, ICCN 2015*, pp. 177–185, ind, August 2015.

[9] J. L. Shah, "A novel approach for securing IPv6 link local communication," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 136–150, 2016.

[10] X. Wang, Y. Mu, G. Han, and D. Le, "A secure IPv6 address configuration protocol for vehicular networks," *Wireless Personal Communications*, vol. 79, no. 1, pp. 721–744, 2014.

[11] Y. Lu, M. Wang, and P. Huang, "An SDN-based authentication mechanism for securing neighbor discovery protocol in IPv6," *Security and Communication Networks*, vol. 2017, pp. 1–9, 2017.

[12] S. Praptodiyono, I. H. Hasbullah, M. M. Kadhum, R. K. Murugesan, C. Y. Wey, and A. Osman, "Improving security of duplicate address detection on IPv6 local network in public area," in *Proceedings of the 2015 9th Asia Modelling Symposium (AMS)*, pp. 123–128, Kuala Lumpur, Malaysia, September 2015.

[13] F. A. Barbhuiya, G. Bansal, N. Kumar, S. Biswas, and S. Nandi, "Detection of neighbor discovery protocol based attacks in IPv6 network," *Networking Science*, vol. 2, no. 3-4, pp. 91–113, 2013.

[14] R. Hassan, A. S. Ahmed, and N. E. Osman, "Enhancing security for IPv6 neighbor discovery protocol using cryptography," *American Journal of Applied Sciences*, vol. 11, no. 9, pp. 1472–1479, 2014.

[15] M. Anbar, R. Abdullah, R. M. A. Saad, E. Alomari, and S. Alsaleem, "Review of security vulnerabilities in the IPv6 neighbor discovery protocol," *Lecture Notes in Electrical Engineering*, vol. 376, pp. 603–612, 2016.

[16] Sridevi, "Implementation of multicast routing on IPv4 and IPv6 networks," *International Journal on Recent and Innovation Trends in Computing and Communication*, pp. 1455–1467, 2017.

[17] Y. Cunjiang, X. Dawei, and J. Li, "Authentication analysis in an IPV6-based environment," in *Proceedings of the 2013 3rd International Conference on Computer Science and Network Technology*, 2013.

[18] M. A. Nia, A. Sajedi, and A. Jamshidpey, "An introduction to digital signature schemes," in *Proceedings of the Telecommunications (IST)*, 2014.

[19] K. Chittimaneni, M. Kaeo, and M. Kaeo, "Operational security considerations for IPv6 networks," *Internet-Draft*, 2014.

[20] N. Abdoun, S. El Assad, M. A. Taha, R. Assaf, O. Deforges, and M. Khalil, "Secure hash algorithm based on efficient chaotic neural network," in *Proceedings of the 2016 International Conference on Communications (COMM)*, pp. 405–410, Bucharest, Romania, June 2016.

[21] S. Gupta, N. Goyal, and K. Aggarwal, "A review of comparative study of MD5 and SSH security algorithm," *International Journal of Computer Applications*, vol. 104, no. 14, pp. 1–4, 2014.

[22] Saranya, Vinothini, and Vasumathi, "A study on RSA algorithm for cryptography," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp. 5708-5709, 2014.

[23] R. Pir, "Security improvement and speed monitoring of RSA algorithm," *IJEDR*, vol. 4, no. 1, 2016.

[24] "Complementary event," https://en.wikipedia.org/wiki/Complementary_event.

[25] G. Wang and Y. Shen, "Preimage and pseudo-collision attacks on step-reduced SM3 hash function," *Information Processing Letters*, vol. 113, no. 8, pp. 301–306, 2013.

[26] F. J. Buenaventura, J. P. Gonzales, M. E. Lu, and A. V. Ong, "IPv6 stateless address autoconfiguration (SLAAC) attacks and

detection," in *Proceedings of the DLSU Research Congress*, vol. 3, 2015.

[27] P. Tayal, "IPv6 SLAAC related security issues and removal of those security issues," *International Journal of Engineering and Computer Science*, vol. 3, no. 9, pp. 8445–8459, 2014.