IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 22, 2019, accepted May 20, 2019, date of publication May 30, 2019, date of current version June 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919966

Authentication and Privacy Approach for DHCPv6

AYMAN AL-ANI[1], MOHAMMED ANBAR[1], IZNAN HUSAINY HASBULLAH[1], ROSNI ABDULLAH[1,2], AND AHMED K. AL-ANI[1]

[1]National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Penang 11800, Malaysia
[2]School of Computer Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia

Corresponding author: Mohammed Anbar (anbar@nav6.usm.my)

This work was supported in part by the Bridging Research Grant, Universiti Sains Malaysia (USM), under Grant 304/PNAV/6316271.

ABSTRACT Dynamic Host Configuration Protocol for IPv6 (DHCPv6) is used to allocate and distribute IPv6 addresses and network configuration parameters to DHCPv6 clients. Two well-known issues of DHCPv6 are privacy concerns due to lack of protection of client information in transit, and lack of verification mechanism that allows attackers to inject fake network configuration parameters into the network undetected. This paper proposes DHCPv6 security (DHCPv6Sec) approach that is based on a hybrid cryptosystem to provide authentication for the DHCPv6 server messages and to protect the privacy of the DHCPv6 client. The DHCPv6Sec was evaluated and compared to the Secure-DHCPv6 in terms of processing time, traffic overhead, rogue DHCPv6 server prevention, privacy protection, and DHCPv6 message size limitation. The experiment results show that the DHCPv6Sec has 52% less processing time; 74% less traffic overhead; and remarkable superiority in all aspects measured.

INDEX TERMS IPv6 network, DHCPv6 server, rogue DHCPv6 server attack, DHCPv6, IPv6 privacy.

I. INTRODUCTION

Over the past few decades, there has been a significant increase in the number of Internet nodes [1]. Internet Protocol version 4 (IPv4) has been used since the beginning of the Internet era to uniquely identify each node on the Internet. However, the recent exponential increase in the number of Internet-facing devices has resulted in all Regional Internet Registries (RIRs) to run out of allocable IPv4 addresses. Internet Protocol version 6 (IPv6) is the upcoming Internet Protocol generation that will replace IPv4. Reports in 2018 of Internet trends, reveal a substantial increase in IPv6 usage [2]. Access of Google via IPv6 has surpassed 25% by April 2019 according to Google statistics [3]. IPv6 slightly improves security of the network, as well as the service quality. However, IPv6 remains vulnerable to security challenges like denial of service (DoS) and man-in-the-middle (MITM) attacks [4].

Dynamic Host Configuration Protocol for IPv6 (DHCPv6) and stateless auto-configuration (SLAAC) are two standard mechanisms that are used to configure IPv6 addresses of clients [5], [6]. However, DHCPv6 provides network administrator with more control on the network compared to SLAAC mechanism [7]. Also, DHCPv6 is utilized to distribute network configuration parameters to the clients in IPv6 network [8], [9]. DHCPv6 allows network administrators to configure more than 30 different network parameters [10] such as Network Time Protocol (NTP) server address, Session Initiation Protocol (SIP) server address, and Domain Name System (DNS) server address [10], [11] for DHCPv6 clients. DHCPv6 is, therefore, widely utilized in IPv6 networks that require such control. However, DHCPv6 has vulnerabilities which could threaten the security of IPv6 network. Several studies, such as [12] and [13], addressed and discussed the DHCPv6 security issues. Some of the vulnerabilities allow attackers to exploit DHCPv6 messages to provide incorrect configuration parameters to the client, either to divert the client's traffic towards a rogue server or to cause DoS [14]–[16].

Moreover, some critical client information is exposed by DHCPv6 messages [17]–[21] such as the type of device, the detail about the operating system, and hostname. Some of this information could be leveraged by attackers who have knowledge of the device or the specific vulnerabilities of the software to swiftly locate potential targets in IPv6 link-local network.

The privacy aspect of the client was not being considered in the original design of DHCPv6 [20]. Many approaches

The associate editor coordinating the review of this manuscript and approving it for publication was Quansheng Guan.

have been proposed such as Anonymity Profile [21] and Secure-DHCPv6 [22], to provide authentication and privacy for DHCPv6. However, these approaches are difficult to deploy in large-scale networks and reduces some of the functionalities of DHCPv6.

This paper proposes a new approach called DHCPv6 security (DHCPv6Sec) that provides anonymity of DHCPv6 client from attackers, but not to the DHCPv6 server. The proposed approach is designed to overcome the limitations and drawbacks of Secure-DHCPv6.

The organization of the paper is as follows: Section II provides background information about the DHCPv6 process and its threat model. Section III reviews related studies. The design of DHCPv6Sec is introduced in Section IV. The implementation of DHCPv6Sec is provided in Section V. The experiments, as well as the evaluation are provided in Section VI. A discussion of the results is provided in Section VII. Section VIII provides the conclusion, as well as further studies.

II. BACKGROUND

In IPv6 network, DHCPv6 server is typically deployed to assign IPv6 addresses and distribute network configuration parameters, such as DNS and NTP server addresses [10], [11] to DHCPv6 clients. DHCPv6 is similar to DHCPv4 in the IPv4 network in term of functionalities. However, the message formats are different, and both are vulnerable and susceptible to different types of attacks. For example, DHCPv4 is susceptible to starvation attack but not DHCPv6 because it has a huge amount of available IPv6 addresses at its disposal [21]. In addition, DHCPv6 treats client privacy differently than DHCPv4. DHCPv6 uses the client's DHCP unique identifier (DUID) which has potential privacy issues, whereas DHCPv4 does not [12]. Therefore, most of DHCPv4 security approaches cannot be applied directly to DHCPv6. As such, this paper focuses only on DHCPv6 and its security issues.

DHCPv6 has two modes of operation: stateful and stateless. The stateful mode is utilized to allocate and assign IPv6 addresses and distribute other network configuration parameters. Whereas, the stateless mode is used to only distribute the network configuration parameters.

A DHCPv6 server could be deployed in an IPv6 link-local network directly or remotely via a DHCPv6 relay agent. The DHCPv6 relay agent is only required in the case where the DHCPv6 client and the DHCPv6 server are not connected to the same link-local network. RFC 8213 document stated that the connection between DHCPv6 relay agent and DHCPv6 server could be secured using IPSec [23]. Therefore, this study will focus on the security issues of the DHCPv6 message exchange between DHCPv6 client and its first hop, which could be a DHCPv6 server or a DHCPv6 relay agent.

When a DHCPv6 client connects to a new IPv6 network, it first multicasts a Router Solicitations (RS) message, expecting a reply from a router with Router Advertisement (RA) message. Depending on the RA message it received, the client will operate either in stateful or stateless mode. During DHCPv6 stateful mode, the client should multicast DHCPv6 Solicit message to all DHCPv6 servers, which are located on the link-local network. The server will then respond through DHCPv6 Advertise message with related configuration information to the client. Next, the client will send DHCPv6 Request message to confirm the configuration. Finally, the server should send a DHCPv6 Reply message to confirm the configuration [24]. Figure 1 illustrates the basic process of DHCPv6 operation during the stateful mode.

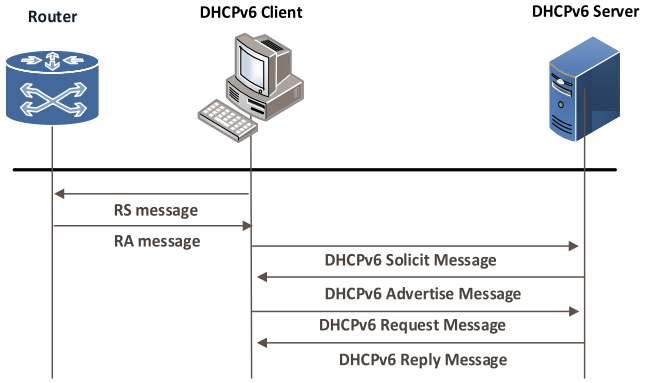


FIGURE 1. DHCPv6 operation during stateful mode.

On the other hand, in stateless mode, the client should multicast Information-request message and the server replies with a Reply message that contains the requested configuration parameters [25]. The server is the one that sends the configuration parameters to all local clients on the same network.

The DHCPv6 has two security issues which are rogue DHCPv6 server attack, and DHCPv6 privacy, that can be exploited by attackers. The following subsection illustrates these issues.

A. ROGUE DHCPv6 SERVER ATTACK

A DHCPv6 client configures its IP address and other network parameters based on the information contained in DHCPv6 server messages it received. However, the client does so without first verifying the legitimacy of the message source [26]. Therefore, attackers that are connected to the same link-local network, could masquerade as legitimate server and inject fake messages into the traffic to fool the client. The attack occurs when the client sends a Solicit message asking the server to reply. An attacker on the network will respond back with a fake Advertise message containing wrong network configuration parameters. Since the client does not have at its disposal a mechanism to verify the source of this message, it will readily accept the message and configure its IP address, as well as other network parameters with incorrect information. Hence, the client is under attack such as DoS or redirect the user to rogue servers as shown in Figure 2 [22], [27], [28]. Accordingly, authentication of the DHCPv6 server message is considered essential in IPv6 networks.

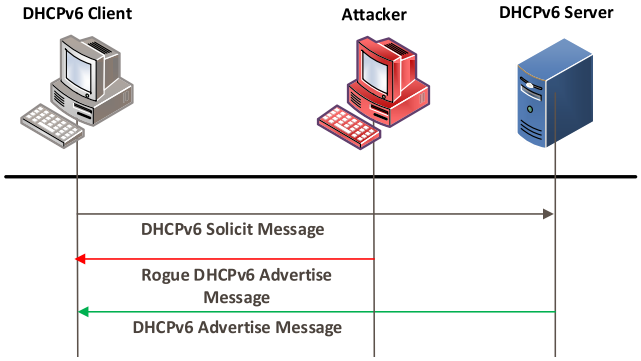


FIGURE 2. DHCPv6 security challenge.

B. DHCPv6 PRIVACY

DHCPv6 messages may reveal crucial information about the client. This information includes several identifiers such as Client's DUID and hostname. These identifiers could be used as a stable identifier to the DHCPv6 client for tracking and profiling users and their activities over time. A stable identifier is a unique information that does not change over time and it can be used to distinguish one client from another.

Moreover, this information can be used to digitally fingerprint a client as it reveals the device type, the vendor name or the operating system type and specific version. This information could be exploited by attackers to monitor clients and to know the potential vulnerabilities of the device vendor or the operating system. In addition, attackers that monitor the DHCPv6 traffic through passive monitoring can obtain the hostname, operating system, and vendor name of the DHCPv6 clients. They could correlate such information with other information, such as from those extracted from traffic analysis and other information sources; they could potentially identify the device, device properties, and its user. Additionally, the DHCPv6 message can be used to know previously visited networks of the device. Therefore, the clients' privacy is disclosed due to the DHCPv6 information [12].

DHCPv6 is considered one of the main components in the IPv6 network and, accordingly, its protection is required.

III. RELATED WORK

There are many approaches that have been proposed to provide authentication for DHCPv6 server message and privacy for the DHCPv6 client. This paper classified these approaches into two groups: authentication approach, and privacy approach as shown in Figure 3.

A. AUTHENTICATION APPROACH

Authentication approach group includes approaches that prevent rogue DHCPv6 server in IPv6 link-local network.

1) DELAYED AUTHENTICATION APPROACH (DAA)

RFC 3315 [29] introduced Delayed Authentication Approach (DAA) as an approach to secure DHCPv6 by authenticating DHCPv6 server messages. DAA is a symmetric

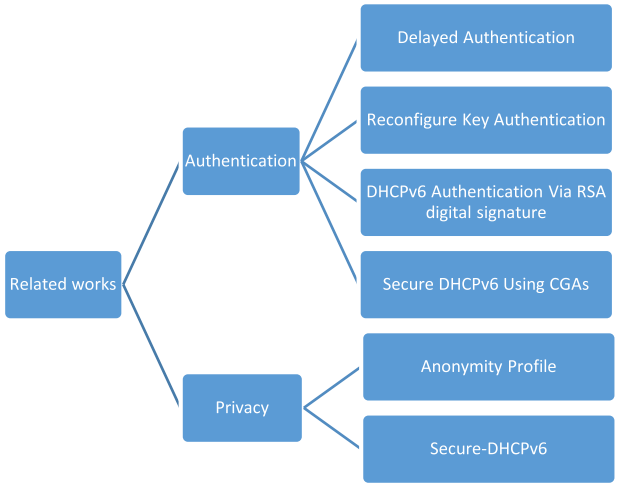


FIGURE 3. Classification of related work.

authentication approach based on Hash Message Authentication Code (HMAC) using MD5. It requires preconfiguring clients and servers with a secret key. Every message exchanged between a client and a server is digested using HMAC and the resulting hash value is appended to the DHCPv6 message. The receiver can verify the message by comparing the hash value using the same secret key. Even though this approach could prevent rogue DHCPv6 server attacks, the clients and servers must be preconfigured with the key. Since this approach does not provide any key distribution method, the key is usually distributed manually, which makes it difficult and impractical to be managed and deployed in a large-scale network.

2) RECONFIGURE KEY AUTHENTICATION APPROACH

RFC 8415 [24] specifies Reconfigure Key Authentication (RKAP) approach to secure DHCPv6 server messages. RKAP is designed to prevent spoofing of Reconfigure message by rogue DHCPv6 server. This approach is similar to DAA as it uses HMAC with MD5. The secret key is sent by the server in cleartext to the client during initial message exchange. Since the key is transmitted as cleartext, attackers could hijack the initial message containing the key and use the key to authenticate a malicious Reconfigure message. Furthermore, this approach is limited to securing the Reconfigure message only and not designed to provide authentication for all DHCPv6 messages.

3) DHCPv6 AUTHENTICATION VIA RSA DIGITAL SIGNATURE

Based on [30], a solution for DHCPv6 message authentication is provided using an RSA (Rivest–Shamir–Adleman) digital signature. A digital signature is an asymmetric authentication approach, which uses a private key for signing the messages and a public key for verifying the messages [31]. Client must keep its own private key in addition to the public key of the DHCPv6 server. On the other hand, the server must keep its own private key besides the public keys of the clients. The public keys are distributed manually between

clients and servers. The client and DHCPv6 server must sign every DHCPv6 message with their private key and append a signature to the DHCPv6 message. The receiver (i.e., client or server) uses the public key of the sender to verify the signature of the message. By doing so, the receiver can authenticate the DHCPv6 message source [30]. Even though this approach does provide authentication and integrity protection for DHCPv6 messages, the manual distribution of the key makes it difficult and impractical to be managed in a large-scale network.

4) SECURE-DHCPv6 USING CGA

Sheng Jiang and Sean Shen proposed Secure-DHCPv6 using Cryptographically Generated Address (CGA) approach to provide authentication for DHCPv6 server messages. The CGA is a security approach that proves source address ownership and provides data integrity protection [32]. In this approach, the client should be manually configured with DHCPv6 server's IP address. Therefore, the DHCPv6 client can easily verify the server message it received by comparing the source address with the pre-configured server's IPv6 address. However, the configuration of the server IP address on each client has to be done manually [33].

B. PRIVACY APPROACH

The security approaches that attempted to protect the privacy of DHCPv6 clients in IPv6 link-local network are grouped under the privacy classification.

1) ANONYMITY PROFILE APPROACH

Anonymity Profile approach attempts to anonymize the DHCPv6 client to the network and DHCPv6 server. It avoids using any options such as Class Vendor option, and User Class Option that may expose information about the client. Furthermore, it randomizes the DUID of the client and, therefore, the attacker will be prevented from correlating the client activities. Anonymity Profile approach avoids using authentication, such as those in Section III.A because it can possibly be used to fingerprint a client based on the unique identifiers used by the authentication approach that could potentially be linked back to a client. Although Anonymity Profile does protect the client privacy, the Anonymity Profile limits the use of some DHCPv6 functionalities and renders it unusable with the authentication mechanism, which makes it unsuitable in many situations [21].

2) SECURE-DHCPv6 APPROACH

Due to the limitations and drawbacks of Anonymity profile, Li, and et al. (2018) proposed Secure-DHCPv6 to provide authentication and to protect the privacy of DHCPv6 clients [22] by anonymizing the DHCPv6 clients to the attackers. This approach has three components: a digital certificate (DC) to authenticate DHCPv6 messages; a digital signature to ensure the integrity of DHCPv6 messages; and an Asymmetric Key Cryptography (AKC) to protect the privacy of DHCPv6 clients [34]. Every DHCPv6 client and server has their own DC, which is used to sign and encrypt DHCPv6 messages. The client starts the communication with the server by exchanging DC and negotiating for an algorithm to be used, which by default is the RSA algorithm, then begins to exchange DHCPv6 messages. Secure-DHCPv6 uses AKC, DC and digital signature, as well as requires two extra messages to be sent before DHCPv6 messages can be exchanged, which makes this approach complicated. In addition, this approach does not provide a mechanism to distribute the DC, thus the client and server must be manually configured with a trusted DC. Therefore, it is difficult and impractical to be deployed in a large network.

Secure-DHCPv6 also puts a limit on the size of DHCPv6 message as this approach utilizes RSA for encryption, which is not designed to encrypt big messages [35]–[37]. DHCPv6 has more than 130 options defined and some of these options are used to transmit domain name and URL, in which the size may reach up to 64kb [38]. Further, Secure-DHCPv6 did not define certificates' types, such as anonymous DC, that the client should use. In Secure-DHCPv6 approach, the client exposes its DC every time it joins a new network, thus the client's DC can be used as a stable identifier to track and profile the client.

IV. PROPOSED DHCPV6SEC

The main goal of this research is to propose DHCPv6Sec approach to overcome the limitations of the Secure-DHCPv6 by utilizing hybrid cryptosystem and RA message. The hybrid cryptosystem is utilized to reduce the complexity of the approach and to remove the message size limitation that was put by Secure-DHCPv6. The RA message is used to provide a distribution mechanism for the server public key. In addition, the DHCPv6Sec is designed to prevent the exposure of DHCPv6 client's information.

Hybrid cryptosystem uses two separate cryptosystems: Symmetric Key Cryptography (SKC) algorithm [39] and AKC algorithm [40]. SKC algorithm is used to encrypt and decrypt DHCPv6 message option using a secret key. AKC algorithm is used to distribute the secret key of the client to the server. AKC algorithm uses a public and private key pair. The public key will be used by the client to encrypt the secret key to be sent to the server; the server decrypts the secret key by using its own private key. The private key will be generated manually at the DHCPv6 server. Whereas, to distribute the public key to the clients, DHCPv6Sec utilizes RA message since it is already being used in IPv6 link-local network to provide IPv6 clients with network configuration information such as stateless or stateful modes. In this research, it is assumed that the RA message is secured by a third party mechanism such as Secure Neighbor Discovery (SEND) [41] or RA guard [42]. By using RA message as key distribution mechanism, the public key no longer has to be configured manually on each client.

DHCPv6Sec utilizes RSA 2048 and Advanced Encryption Standard with Galois Counter Mode (AES-GCM) 128. RSA was selected to be used for AKC [43] due to its strong

TABLE 1. Proposed option and its fields.

Option Name	Option Fields	Used By	Option Format
DHCPv6 Public Key (DPK)	▪ Public Key	Router	RA Option (RFC 4861)
Encrypted Key (EK)	▪ Encrypted Secret Key	DHCPv6 client	DHCPv6 Option (RFC 7227)
Key ID (KI)	▪ Key ID ▪ Key Lifetime	DHCPv6 server	
Replay Detection (RD)	▪ Replay Detection	DHCPv6 client and server	
Client Encrypted (CE)	▪ Key ID ▪ Nonce ▪ Tag ▪ Cyphertext	DHCPv6 client	
Server Encrypted (SE)	▪ Nonce ▪ Tag ▪ Cyphertext	DHCPv6 server	

cryptographic features. Moreover, AES-GCM was selected for use with SKC [44] because of its impressive performance on multiple platforms to provide privacy and authentication. Several new messages and options have been proposed to allow the DHCPv6 message to be encrypted and to allow the use of RA message to convey the server's public key to DHCPv6 clients. Section IV.A describes the newly proposed DHCPv6Sec options and Section IV.B illustrates AES-GCM operation. DHCPv6Sec also provides a mechanism to prevent replay attack, which is detailed in Section IV.C.

A. THE DHCPv6Sec OPTIONS

Several new options are introduced in DHCPv6Sec to protect client privacy and to provide authentication for DHCPv6 messages. These options are designed according to the RFC 4861 specification for RA option [45] and the RFC 7227 specification for DHCPv6 options [38]. The proposed options and their fields are tabulated in Table 1.

- *DHCPv6 Public Key (DPK):* allows RA message to convey public key to DHCPv6 client. The client uses the public key to encrypt a secret key. The DPK option should be appended to each RA message that is sent out by the router.
- *Encrypted Key (EK):* is designed to convey encrypted secret key that will be used to encrypt DHCPv6 messages. This option will be used by DHCPv6 client.
- *Key ID (KI):* is designed to convey Key Lifetime indicating for how long the client should keep the key; and Key ID to be used to correlate between Ciphertext and secret key. This option will be used by DHCPv6 server.
- *Replay Detection (RD):* is designed to convey strictly monotonically increasing timestamp to prevent replay attack. This option will be used by both DHCPv6 client and server.
- *Client Encrypted (CE):* is designed to convey the encrypted DHCPv6 client message, which includes Key ID to identify the secret key that is used for encryption; Ciphertext, which is encrypted standard DHCPv6 client message options; nonce; and tag. The nonce and tag are used by AES-GCM to provide authentication. This option will be used by DHCPv6 client.
- *Server Encrypted (SE):* is designed to convey the encrypted DHCPv6 server message including Ciphertext, which is encrypted DHCPv6 server message; Cyphertext; nonce; and tag. This option will be used by DHCPv6 server.

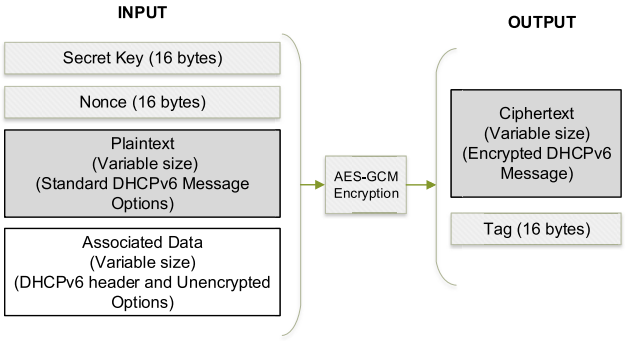


FIGURE 4. The input/output operation of AES-GCM encryption.

B. AES-GCM OPERATION

The AES-GCM is used by DHCPv6Sec to (i) ensure the integrity for both unencrypted DHCPv6Sec message header and DHCPv6 options (RD, KI and EK options), which must be checked before the verification and decryption process and (ii) ensure the integrity of DHCPv6 message option and to provide privacy for DHCPv6 client by protecting the information inside the DHCPv6 message options such as Client Identifier and Option Request options.

The DHCPv6Sec sender (i.e., client or server) encrypts DHCPv6 messages following the operation as shown in Figure 4.

- *Encryption Process Input:* Secret Key, Nonce, Plaintext, Associated Data; where Secret Key (16 bytes) is the key that is used for encryption; Nonce (16 bytes) is a random value used by the AES-GCM; Plaintext (variable size) is the DHCPv6 message options to be encrypted; and Associated Data (variable size) is the unencrypted data part.
- *Encryption Process Output:* Ciphertext; Tag, where Tag (16 bytes) is used for the authentication process; and Ciphertext is the encrypted DHCPv6 message options.

Figure 4 shows the input/output operation of the AES-GCM encryption. The output of the encryption process should be inserted into CE or SE options together with the Nonce.

Furthermore, during the decryption process, the receiver should treat the DHCPv6 message as follows:

- *Decryption Process Input:* Secret Key, Nonce, Tag, Ciphertext, and Associated Data; where Secret Key (16 bytes) is the key that is used for encryption; the Nonce (16 bytes) is the same random value that is used in encryption process; Ciphertext (variable size) is the encrypted DHCPv6 message; Associated Data (variable size) which is the header and encrypted option that is

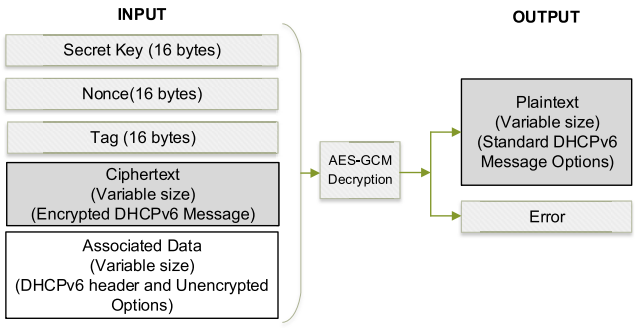


FIGURE 5. AES-GCM decryption input/output operation.

used in the encryption process data; and Tag (16 bytes) is the output of the encryption process.

- *Decryption Process Output:* Plaintext or Error, where Plaintext is the decrypted DHCPv6 message, Error indicates the DHCPv6 message that has been modified.

Figure 5 shows the input/output operation of the AES-GCM decryption. If the decryption process is successful (i.e., Plaintext), the receiver processes the decrypted message options as per RFC 8415; else the entire DHCPv6 message should be discarded.

C. PREVENTING REPLAY ATTACK

Replay attack is very common in secure network communication [46]. In replay attack, an attacker makes use of stale authentication message to trick the victim to use old configuration parameters which leads to DoS attack or MITM. To prevent replay attack, DHCPv6Sec utilizes RD option and Transaction-Id. Transaction-Id should be generated and verified according to the RFC 8415 specification. Furthermore, the RD option should be appended to all DHCPv6Sec messages and should always be verified by the receiver. The sender (DHCPv6 client or server) should set the RD option with a monotonically increasing timestamp. The client or server that receives RD option compares its RD value with the previously recorded value from the same sender; if the value of the latter is greater, then the receiver accepts the message; else the message should be discarded. On the other hand, if it is the first time a receiver receives a message with an RD option, the receiver skips the replay detection; but only records the RD value in its cache to be used in replay detection process later as shown in Figure 6.

D. DHCPv6Sec WORKFLOW

The DHCPv6Sec workflow consists of three main stages: deploying public and private keys, sharing the secret key, and exchanging DHCPv6 messages. This section describes these stages.

1) DEPLOYING PUBLIC AND PRIVATE KEYS (STAGE 1)

The purposes of this stage are (i) to generate a public and private key pair for the DHCPv6 server, and (ii) to distribute the server's public key to clients in the network. DHCPv6Sec

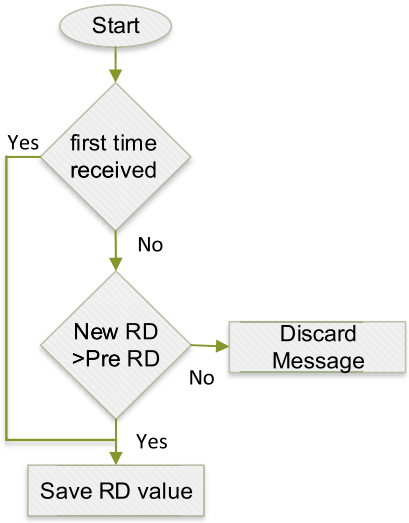


FIGURE 6. Process RD option.

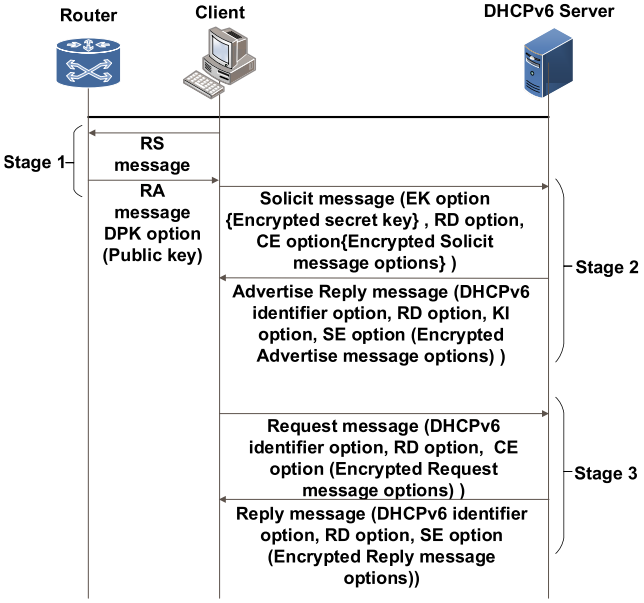


FIGURE 7. DHCPv6Sec message exchange.

operation begins by generating RSA public and private keys for the DHCPv6 server. The private key is kept by the server and the public key is manually deployed in the router for distribution to clients in the network.

Whenever a new DHCPv6 client joins a network, the client must first obtain the server's public key from the router by multicasting RS message (multicast IPv6 address FF02::2). The router replies by multicasting back RA message (multicast IPv6 address FF02::1) with DPK option to convey the server's public key to the client as shown in Figure 7 (Stage 1). The public key is not manually configured or manually distributed to the clients, which makes DHCPv6Sec more practical and manageable in large-scale networks compared to other authentication approaches.

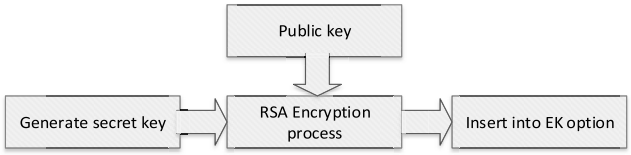


FIGURE 8. Encrypt secret key process.

Message Type	Transaction Id
RD Option	
EK Option (Encrypted Secret Key)	
CE Option (Key ID, Nonce, Tag, Ciphertext)	

FIGURE 9. DHCPv6 message with RD, EK, and CE options.

2) SHARING THE SECRET KEY (STAGE 2)

The aim of this stage is to generate and share the secret key and Key ID of the AES-GCM. This stage is divided into four steps: generating secret key, decrypting secret key, sending Key ID, and storing Key ID.

a: GENERATING THE SECRET KEY

The purposes of this step are (i) to generate a secret key, and (ii) to encrypt DHCPv6 client message at the client's side. After receiving the server's public key from the router, the client generates a secret key and encrypts it with the server's RSA public key. The client sends the encrypted secret key by using EK option with a first exchanged message i.e. Solicit message as illustrated in Figure 8.

The secret key is required by the client and server to encrypt and decrypt DHCPv6Sec message using AES-GCM algorithm. As mentioned in Section IV.B, DHCPv6Sec does not encrypt the entire DHCPv6 message; only the portion of the standard DHCPv6 message options that contain client's information would be encrypted. The client generates a Solicit message containing a message type; Transaction-Id; standard Solicit message options such as Client Identifier and Option Request options; and the proposed DHCPv6Sec options such as RD, EK and CE. The standard Solicit message are encrypted and authenticated with the secret key using AES-GCM as detailed in Section IV.B; the encrypted value is inserted into CE option. The Key ID field of CE option should be set to 0 as the server will use the Key ID to correlate between the secret key and Ciphertext. Furthermore, EK option is used to convey the encrypted secret key. RD option and Transaction-Id will be processed as described in Section IV.C to prevent replay attack. Figure 9 shows the Solicit message with RD, EK, and CE options. In Figure 7, Stage 2 shows the client sends Solicit message to the DHCPv6 server.

b: DECRYPTING THE SECRET KEY

The aim of this step is to decrypt the secret key and the DHCPv6 client message at the server side. After receiving

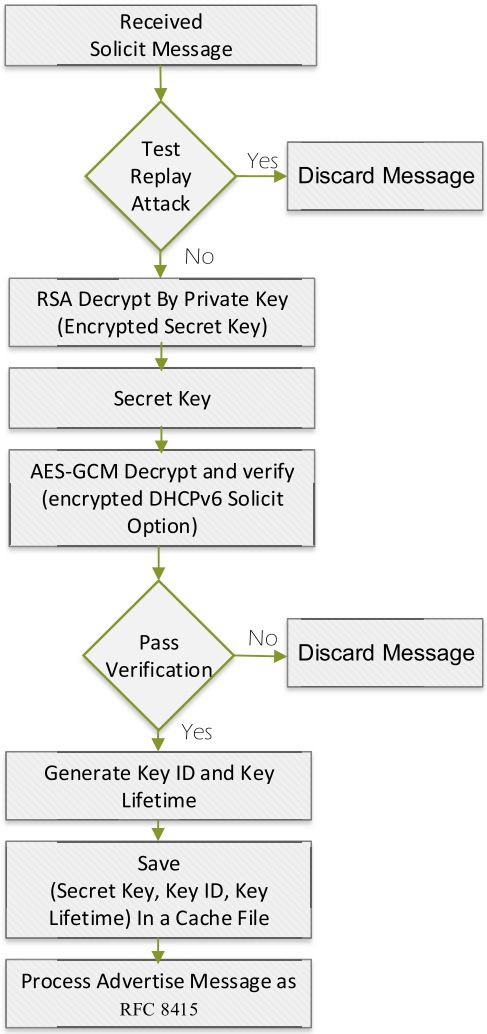


FIGURE 10. Decrypt secret key and save at the DHCPv6 server.

the DHCPv6 Solicit message, the server first needs to test for replay attack as mentioned in Section IV.C. Then, the server decrypts the encrypted secret key (EK option) by using its own RSA private key. Next, the server decrypts and verifies CE option with AES-GCM by using the secret key as illustrated in Section IV.B to ensure the message integrity and to obtain the DHCPv6 Solicit message options. After that, the server should generate a Key Lifetime and Key ID for the secret key. The Key ID is used later to correlate the secret key with client messages. The Key Lifetime is the validity period of the key. The Key ID, Key Lifetime, and secret key need to be cached by the server to be used later for encryption and decryption of DHCPv6Sec message options as illustrated in Figure 10. The server processes the Solicit message with the decrypted options based on RFC 8415.

c: SENDING KEY ID

The purpose of this step is to send Key ID and encrypt DHCPv6 server message at the server side. After processing the Solicit message, the server will reply to the client by sending a DHPCv6 Advertise message with RD option,

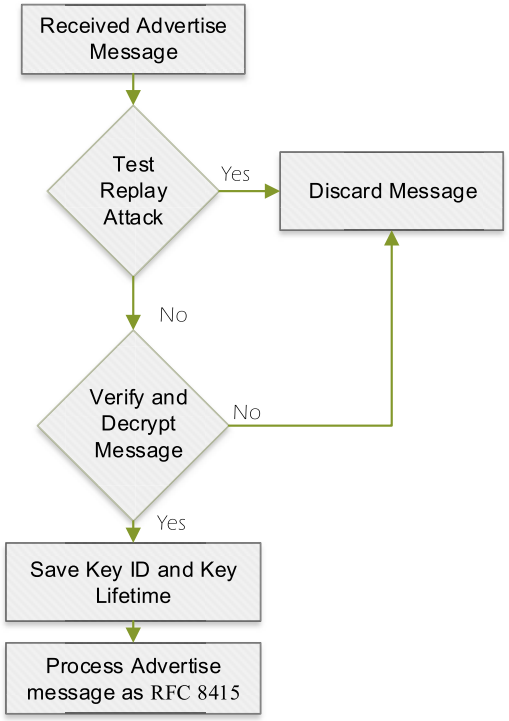


FIGURE 11. DHCPv6 client process encrypted Reply message.

KI option, and SE option that contains the encrypted part of the DHCPv6 Advertise message options. Similar to the Solicit message option encryption, the original Advertise message options are encrypted and inserted into the SE option. The KI option will be used to convey the Key ID and lifetime of the secret key to the client. Figure 7 illustrates the Advertise message exchange between the server and the client in Stage 2.

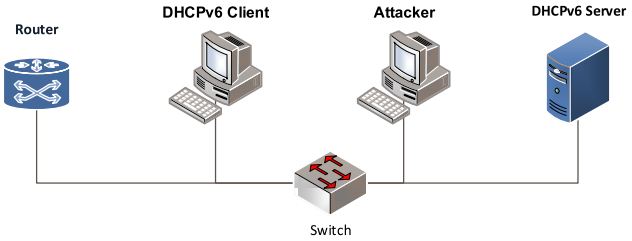
d: STORING KEY ID

In this step, the client stores the Key ID which is needed to decrypt DHCPv6 server message. After receiving the DHCPv6 Advertise message, the client should test for replay attack as stated in Section IV.C. The client decrypts and verifies SE option with AES-GCM using the secret key as described in Section IV.B. Then, the client stores the Key ID and Key Lifetime. The key ID will be appended to CE option later; and the Key Lifetime indicates the validity period of the keys (secret key and Key ID) before they need to be regenerated. By doing so, the client and server share the same secret key, Key ID and Key Lifetime. The client and server use the secret key to encrypt and decrypt subsequent DHCPv6 message exchanges in Stage 3. After that, the client processes the Advertise message as per RFC 8415 [15] as illustrated in Figure 11.

3) EXCHANGING DHCPv6 MESSAGES (STAGE 3)

Stage 3 involves encrypting and decrypting DHCPv6Sec messages that have been exchanged between the client and the server. Once the server obtains the secret key from the client in Stage 2, the subsequent DHCPv6Sec message exchanges between the client and server will be encrypted and decrypted using AES-GCM with the secret key and Key ID. Figure 7 illustrates the encrypted message exchanges between the client and the server in Stage 3. When the lifetime of the key expires, the client is required to regenerate a new secret key and repeat the process of Stage 2. Furthermore, whenever verification or test replay attack fails in any stage of the operation, which indicates that the message is not authentic, the message should be discarded.

By doing so, DHCPv6Sec prevents rogue DHCPv6 server attack and protects the DHCPv6 message exchanges between DHCPv6 server and clients. DHCPv6Sec could be used with randomized MAC addresses to provide full privacy protection for DHCPv6 client.



Device Type	Operating System	CPU Type	Memory
DHCPv6 Server	Ubuntu 16.04	Intel Core™ i7-2640M CPU @ 2.80GHz × 4	4 Gb
DHCPv6 Client	Ubuntu 16.04	Intel Core™ i7-3770 CPU @ 3.40GHz × 8	8Gb
Attacker	Kali Linux 2018.4	Intel Core™ i7-2640M CPU @ 2.30GHz × 4	4Gb
Router	Ubuntu 16.04	Intel Core™ i7-2640M CPU @ 2.30GHz × 4	4 Gb

FIGURE 12. Network topology and device specification.

V. IMPLEMENTATION OF THE PROPOSED APPROACH DHCPV6SEC

To evaluate the proposed approach for DHCPv6Sec, experiments were performed in a local network. Figure 12 illustrates the network topology and specifications of the devices used in the experiment, which consist of a DHCPv6 server, a DHCPv6 client, a router and an attacker. An open-source DHCPv6 server application, DHCPket server was modified to support DHCPv6Sec. DHCPket is written in python programming language [47]. A python-based DHCPv6 client was modified to use DHCPv6Sec as well as to receive and process RA message with public key. The Pycryptodome library was utilized for AES-GCM and RSA algorithms [48] in both DHCPv6 client and server. The attacker ran tools on Kali Linux, which was specifically made and used for penetration testing. Scapy and Wireshark tools were used to sniff server messages [49], [50]. The RA guard was utilized to prevent rogue RA message [47].

VI. EXPERIMENTS AND EVALUATION

The proposed DHCPv6Sec is designed to overcome the limitation of Secure-DHCPv6 approach. The main differences between the two approaches are (i) DHCPv6Sec used hybrid cryptosystem to provide authentication and privacy, whereas Secure-DHCPv6 used DC and AKC; (ii) DHCPv6Sec used RA message to automatically distribute the server's

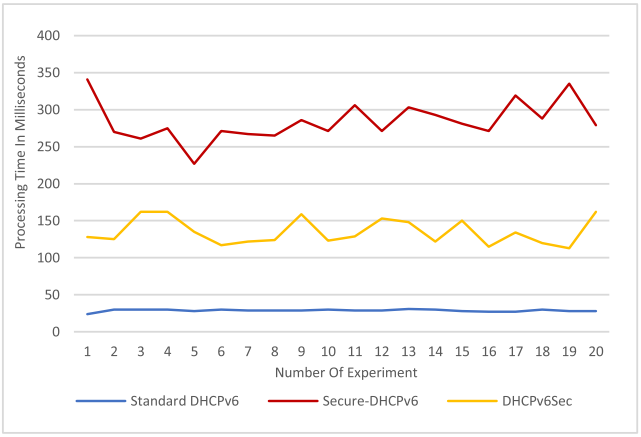


FIGURE 13. The total processing time of various approaches (in Milliseconds).

public key, whereas Secure-DHCPv6 approach used two extra DHCPv6 messages to distribute the server's DC. Accordingly, the DHCPv6Sec was evaluated and compared against Secure-DHCPv6 in term of processing time, traffic overhead, rogue DHCPv6 server attack prevention, privacy issue and message size limitation. This section describes the experiments.

A. PROCESSING TIME

The aim of this experiment is to measure the total processing time for standard DHCPv6, Secure-DHCPv6, and the proposed DHCPv6Sec approaches. The total processing time to generate and verify the DHCPv6 messages were calculated. The messages which were measured include Solicit, Advertise, Request, and Reply; as well as Information-request and Reply message that were used by Secure-DHCPv6 to discover and obtain DHCPv6 server's DC prior to establishing client-server communication. The total processing time (Pt) was calculated by using Equation (1):

$$Pt = \sum_{DHCPv6messages}^{i=0} \left(Et_{(Gp)i} - St_{(Gp)i} + Et_{(Vp)i} - St_{(Vp)i}\right) \quad (1)$$

where, Pt is total processing time, Et is end time, St is start time, Gp is generation process and Vp is verification process.

Since the processing time could be affected by other non-related operations of the operating system, the experiment was repeated 20 times in order to obtain more reliable results. Figure 13 shows a line chart of the processing time.

Based on the experiment result, the total processing time of DHCPv6Sec is about 52.41% less than Secure-DHCPv6. This is due to the use of AKC and DC by Secure-DHCPv6 for authentication and protection of the privacy of the client. Table 2 shows the mean and standard deviation (STDVE) of the processing time as well as the corresponding overhead to generate and verify various types of DHCPv6 messages. The overhead is calculated by using the standard DHCPv6 average processing time as a baseline.

TABLE 2. Processing time for generating and verifying DHCPv6 messages (in milliseconds).

		Standard DHCPv6		Secure-DHCPv6			DHCPv6Sec		
		Mean	STDVE	Mean	STDVE	Overhead	Mean	STDVE	Overhead
Generation	Solicit	7	1	111	16	105	90	16	83
	Advertise	5	0	8	2	3	6	1	1
	Request	8	0	13	4	6	10	1	2
	Reply	6	0	7	0	2	6	1	0
	Information-request	0	0	1	0	1	0	0	0
	Reply (Information-request)	0	0	10	0	10	0	0	0
Verification	Solicit	1	0	33	7	32	21	2	20
	Advertise	1	0	31	3	31	3	1	2
	Request	1	0	31	8	30	4	0	3
	Reply	1	0	31	3	30	3	0	2
	Information-request	0	0	0	1	0	0	0	0
	Reply (Information-request)	0	0	4	14	4	0	0	0

Based on the result in Table 2, the processing time for Secure-DHCPv6 is considerably higher compared to DHCPv6Sec because Secure-DHCPv6 is more complex than the proposed approach. In addition, two extra messages are required which resulted in increased processing time and complexity of Secure-DHCPv6.

B. TRAFFIC OVERHEAD

The aim of this experiment is to measure the traffic overhead for standard DHCPv6, Secure-DHCPv6, and the proposed DHCPv6Sec approaches. The total message size during DHCPv6 stateful mode to obtain IPv6 address was calculated by summing the size of all messages that were exchanged. The messages which were measured include RA, RS, Solicit, Advertise, Request, and Reply; as well as Information-request and Reply message that were used by Secure-DHCPv6 to discover and obtain DHCPv6 server's DC prior to establishing client-server communication. The traffic overhead is benchmarked against the total Standard DHCPv6 message size which serves as a baseline.

From Table 3, Secure-DHCPv6's traffic overhead is 2600 bytes higher than DHCPv6Sec. This overhead is due to two reasons. The first reason is because Secure-DHCPv6 requires two extra messages to be transmitted before DHCPv6 messages could be exchanged; and the second reason is because it uses RSA algorithm for encryption which adds additional 256 bytes of data to the messages.

TABLE 3. Message size and traffic overhead (bytes).

Message name	Standard DHCPv6	Secure-DHCPv6	DHCPv6Sec
RS	70	70	70
RA	150	150	414
Solicit	145	1499	457
Advertise	148	346	208
Request	161	346	213
Reply	148	346	176
Information Request (Secure-DHCPv6)	0	86	0
Reply (Secure-DHCPv6)	0	1295	0
Total	752	4068	1468
Traffic Overhead	0	3316	716

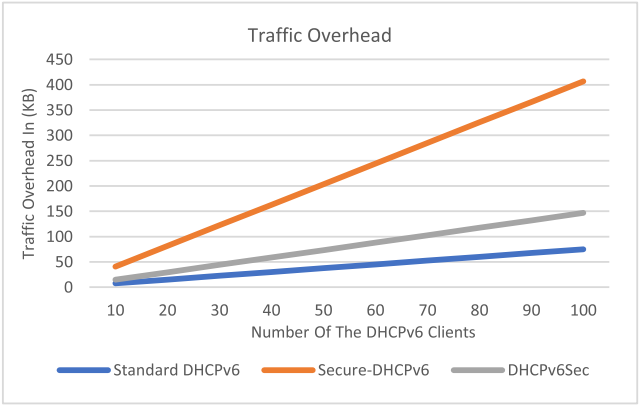


FIGURE 14. Traffic overhead (KB).

Further, the traffic overhead is affected by the number of DHCPv6 clients in the network. Figure 14 shows that the traffic overhead is directly proportional to the number of clients. Theoretically, DHCPv6Sec has 260 KB (260,000 bytes) less traffic overhead than Secure-DHCPv6 when there are 100 clients in the network. The figure of 260,000 bytes is obtained from the multiplication of the traffic overhead of a single Secure-DHCPv6 client (2600 bytes) and the total number of clients (100). Thus, the proposed approach clearly has less traffic overhead compared to Secure-DHCPv6.

C. ROGUE DHCPv6 SERVER ATTACK PREVENTION

This experiment measures the ability of different approaches to prevent rogue DHCPv6 server attack. IPv6 Attack Toolkit and Scapy were used to generate the rogue DHCPv6 server attack. The IPv6 Attack Toolkit and Scapy are open-source tools for penetration testing and attacking IPv6 network. IPv6 Attack Toolkit provides a tool called "fake_dhcps6.c" that behaves as rogue DHCPv6 server. In addition, Scapy was designed to sniff client messages; crafts network packets; and transmits rogue server DHCPv6 messages. The experiment begins with the rogue DHCPv6 server tools running, followed by a client joining the network. If the client configured itself with parameters from rogue DHCPv6 message, the attack would be considered successful, else the attack failed. Table 4 shows the results of the experiment.

TABLE 4. Comparison of rogue DHCPv6 server attack on the various approaches.

Mechanism Name	Success Rogue DHCPv6 Server Attack		Success Rate (Attack)
	fake_dhcps6	Scapy	
Secure-DHCPv6	×	×	0%
DHCPv6Sec	×	×	0%

TABLE 5. Privacy comparison for authentication approaches.

Approach Name	Privacy Issue	Success Rate (Providing Privacy)
Secure-DHCPv6	×	0%
DHCPv6Sec	✓	100%

Based on the results, both Secure-DHCPv6 and DHCPv6Sec managed to prevent rogue DHCPv6 server attack; however, only DHCPv6Sec provided a mechanism to automatically distribute the public key. Thus, the proposed approach is more manageable and practical to be deployed in large-scale network.

D. PRIVACY ISSUE

This experiment tests the ability of Secure-DHCPv6 and DHCPv6Sec to protect the privacy of DHCPv6 client. As mentioned in Section II.B, DUID and hostname could be used as stable identifier for the clients; and it could be correlated with client's activities over time. In Secure-DHCPv6, the client exposes its DC every time it connects to a network. Since DC contains a unique public key, DC has been used as a stable identifier for DHCPv6 client instead of the DUID and hostname. Therefore, client activities can be tracked and profiled. In this experiment, a monitoring tool was connected to the network to sniff the traffic for Secure-DHCPv6 messages. The tool was able to capture and display the DC of all clients whenever they connect to the network. The tool was written in Python language. Table 5 shows the results of the experiment.

The result proves that Secure-DHCPv6 does not protect the privacy of DHCPv6 client because the attacker is able to follow the client across different networks by tracking the DC of the client. Whereas, DHCPv6Sec successfully protected the privacy of client by not exposing any identifiable information that could be used to track and profile the client's online activities.

E. ENCRYPTED MESSAGE SIZE LIMITATION

This experiment examines the ability of Secure-DHCPv6 and DHCPv6Sec to send large DHCPv6 message. As described earlier, Secure-DHCPv6 uses ASK that puts a limit on the size

TABLE 6. Encrypted message size limitation.

Approach Name	Success Sending Message (Size in bytes)			Success Rate (Sending Message)
	1st Scenario (166 bytes)	2nd Scenario (261 bytes)	3rd Scenario (292 bytes)	
Secure-DHCPv6 (RSA2048)	✓	×	×	33%
DHCPv6Sec (RSA2048)	✓	✓	✓	100%

of DHCPv6 message. On the other hand, DHCPv6Sec uses hybrid cryptosystem that does not limit the DHCPv6 message size. The experiment was carried out by transmitting DHCPv6 Advertise messages with multiple options. These options are the standard Advertise DHCPv6 message option such as Client Identifier option and Server Identifier option in addition to options such as DNS server option, and NIS server option. The message sizes for the first, second and third scenarios are 166 bytes, 261 bytes, and 292 respectively. Table 6 shows the results of the experiment.

Based on the result, Secure-DHCPv6 approach failed to send DHCPv6 messages that were bigger than 256 bytes because it used RSA for encryption; and the RSA Plaintext size were based on RSA key size (256 bytes). On the other hand, DHCPv6Sec was successful in sending all messages regardless of their sizes because the hybrid cryptosystem did not put any limitation on the size of DHCPv6 messages.

VII. DISCUSSION

Based on the results of the experiments, DHCPv6Sec reduces the complexities of DHCPv6 processes compared to Secure-DHCPv6. DHCPv6Sec is 52% faster than Secure-DHCPv6. Consequently, this could limit the use of Secure-DHCPv6 in resource constrained devices.

Besides reducing the complexity, DHCPv6Sec also reduces the traffic overhead by around 78% compared to Secure-DHCPv6. Hence, DHCPv6Sec has significantly lower communication cost and bandwidth utilization.

DHCPv6Sec and Secure-DHCPv6 similarly prevent rogue DHCPv6 server attack; however, Secure-DHCPv6 does not provide any mechanism to distribute the key which makes their usage limited and cannot be used in public access Internets such as coffee outlets, airports, and hotels as well as in bring-your-own-device (BYOD) environment. Thus, DHCPv6Sec approach was designed to solve the key distribution issue by providing a distribution mechanism for the public key through the RA message.

Moreover, in the experiment to measure the ability to protect the privacy of DHCPv6 clients, the results showed that DHCPv6Sec outperformed Secure-DHCPv6 because of the use of DC by Secure-DHCPv6 that can be used as a stable identifier to trace and profile DHCPv6 client activities over time. Whereas, DHCPv6Sec is designed to protect the privacy of DHCPv6 clients. Thus, using DHCPv6Sec can ensure the clients remain hidden and secure from potential attackers.

TABLE 7. Experiment results summary and comparison.

Approach Name	Secure-DHCPv6	DHCPv6Sec
Processing Time	High	Medium
Traffic Overhead	High	Low
Authentication	Yes	Yes
Privacy	No	Yes
Operational Limitation	Limit DHCPv6 Message Size	None

Additionally, the experiment proved that DHCPv6Sec did not put a limit on the DHCPv6 message size, while Secure-DHCPv6 did. Therefore, DHCPv6Sec could be used without worrying about the size of the DHCPv6 message that may exceed the maximum message size limit.

The summary of the results of the experiments, as well as the comparisons, are tabulated in Table 7.

VIII. CONCLUSION AND FUTURE WORK

Preventing rogue DHCPv6 server attack and protecting the privacy of IPv6 clients are important security goals for any security approaches in securing IPv6 link-local networks. The proposed DHCPv6Sec is an attempt to provide a sophisticated and modern approach to achieve those goals and at the same time provides an approach to overcome the shortcomings of Secure-DHCPv6. The results of the experiments show clear advantages for DHCPv6Sec in all aspects that were measured side by side with Secure-DHCPv6. Therefore, DHCPv6Sec can be considered as a better alternative to Secure-DHCPv6.

Some potential future works include; (i) using the Elliptic-curve Diffie–Hellman (ECDH) [51] algorithm to distribute secret key instead of RSA algorithm; (ii) hybridization of DHCPv6Sec with other mechanisms such as [52] and [53] to prevent DoS attack on IPv6 duplicate address detection (DAD) process. DHCPv6Sec allows those mechanisms to hide the offered IPv6 address in transit from potential attacker; and (iii) support standardized passive snooping techniques (e.g., SAVI) [54].

ACKNOWLEDGMENT

This research was supported by the Bridging Research Grant, Universiti Sains Malaysia (USM) No: 304/PNAV/6316271.

REFERENCES

[1] S. Groat, M. Dunlop, W. Urbanksi, R. Marchany, and J. Tront, "Using an IPv6 moving target defense to protect the smart grid," in *Proc. IEEE PES Innov. Smart Grid Technol. (ISGT)*, Jan. 2012, pp. 1–7.
[2] *State of IPv6 Deployment 2018*, Internet Society, Reston, VA, USA, 2018.
[3] (2019). *IPv6—Google*. Accessed: Mar. 24, 2019. [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html
[4] S. Yan, X. Huang, M. Ma, P. Zhang, and Y. Ma, "A novel efficient address mutation scheme for IPv6 networks," *IEEE Access*, vol. 5, pp. 7724–7736, 2017.
[5] J. M. V. Ruiz, C. S. Cardenas, and J. L. M. Tapia, "Implementation and testing of IPv6 transition mechanisms," in *Proc. IEEE 9th Latin-Amer. Conf. Commun. (LATINCOM)*, Nov. 2017, pp. 1–6.
[6] G. Yousheng, Y. Lingyun, and H. Lijing, "Addressing scheme based on three-dimensional space over 6LoWPAN for Internet of Things," in *Proc. IEEE 13th Int. Conf. Electron. Meas. Instrum. (ICEMI)*, Oct. 2017, pp. 59–64.

[7] M. Anbar, R. Abdullah, B. N. Al-Tamimi, and A. Hussain, "A machine learning approach to detect router advertisement flooding attacks in next-generation IPv6 networks," *Cogn. Comput.*, vol. 10, no. 2, pp. 201–214, 2018.
[8] L. Tirkkonen, "Utilising configuration management node data for network infrastructure management; Konfiguraationhallinnan datan käyttö verkkoinfrastruktuurin hallintaan," Aalto Univ., Espoo, Finland, 2016.
[9] N. Tripathi and N. Hubballi, "Detecting stealth DHCP starvation attack using machine learning approach," *J. Comput. Virology Hacking Techn.*, vol. 14, no. 3, pp. 233–244, 2018.
[10] J. Brzozowski and G. Van de Velde, *Unique IPv6 Prefix per Host*, document RFC 8273, 2017.
[11] E. Horley, "IPv6 and DHCP," in *Practical IPv6 for Windows Administrators*. Berkeley, CA, USA: Springer, 2014, pp. 191–207.
[12] S. Krishnan, T. Mrugalski, and S. Jiang, *Privacy Considerations for DHCPv6*, RFC Editor, 2016
[13] L. Hendriks, A. Sperotto, and A. Pras, "Characterizing the IPv6 security landscape by large-scale measurements," in *Proc. IFIP Int. Conf. Auton. Infrastruct., Manage. Secur.*, 2015, pp. 145–149.
[14] A. Atlasis and E. Rey, "IPv6 router advertisement flags, RDNSS and DHCPv6 conflicting configurations," Enno Rey Netzwerke (ERNW) Providing Security, 2015.
[15] S. P. Naidu and A. Patcha, "IPv6: Threats posed by multicast packets, extension headers and their counter measures," in *Proc. IJCSNS*, 2015, vol. 15, no. 10, p. 70.
[16] S. Sarma, "Securing IPv6's neighbour and router discovery using locally authentication process," *IOSR J. Comput. Eng.*, vol. 16, no. 3, pp. 22–31, 2014.
[17] F. Gont and W. Liu, "A method for generating semantically opaque interface identifiers (IIDs) with the dynamic host configuration protocol for IPv6 (DHCPv6)," Tech. Rep., 2016.
[18] S. Groat, M. Dunlop, R. Marchany, and J. Tront, "The privacy implications of stateless IPv6 addressing," in *Proc. 6th Annu. Workshop Cyber Secur. Inf. Intell. Res.*, 2010, p. 52.
[19] J. Tront, S. Groat, M. Dunlop, and R. Marchany, "Security and privacy produced by DHCP unique identifiers," in *Proc. 16th North-East Asia Symp. Nano, Inf. Technol. Rel. (NASNIT)*, Oct. 2011, pp. 170–179.
[20] S. Krishnan, T. Mrugalski, and S. Jiang, *Privacy Considerations for DHCPv6*, document RFC 7824, 2016.
[21] C. Huitema, T. Mrugalski, and S. Krishnan, *Anonymity Profiles for DHCP Clients*, RFC Editor, 2016.
[22] L. Li, G. Ren, Y. Liu, and J. Wu, "Secure DHCPv6 mechanism for DHCPv6 security and privacy protection," *Tsinghua Sci. Technol.*, vol. 23, no. 1, pp. 13–21, Feb. 2018.
[23] Y. Pal and B. Volz, *Security of Messages Exchanged Between Servers and Relay Agents*, RFC Editor, 2017.
[24] T. Mrugalski, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, document RFC 8415, 2018.
[25] R. Droms, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*, document RFC 3736, 2004.
[26] S. Shen, X. Lee, Z. Sun, and S. Jiang, "Enhance IPv6 dynamic host configuration with cryptographically generated addresses," in *Proc. 5th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput. (IMIS)*, Jun./Jul. 2011, pp. 487–490.
[27] F. Gont, W. Liu, and G. Van de Velde, *DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers*, RFC Editor, 2015.
[28] V. Alangar and A. Swaminathan, "IPv6 security?: Issue of anonymity," *J. Eng. Comput. Sci.*, vol. 2, no. 8, pp. 2486–2493, 2013.
[29] R. Droms, J. Bound, T. Lemon, B. Volz, C. Perkins, and M. Carney, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, document RFC 3315, 2003.
[30] Z. Su, H. Ma, X. Zhang, and B. Zhang, "Secure DHCPv6 that uses RSA authentication integrated with self-certified address," in *Proc. 3rd Int. Workshop Cyberspace Saf. Secur. (CSS)*, Sep. 2011, pp. 39–44.
[31] N. S. Kumar, G. V. R. Lakshmi, and B. Balamurugan, "Enhanced attribute based encryption for cloud computing," *Procedia Comput. Sci.*, vol. 46, pp. 689–696, Jan. 2015.
[32] S. Jiang and S. Shen, *Secure DHCPv6 Using CGAs*, document draft-ietf-dhc-secure-dhcpv6-07.txt, Work Progress, 2012.
[33] S. Jiang, *Configuring Cryptographically Generated Addresses (CGA) using DHCPv6*, document draft-ietf-dhc-cga-config-dhcpv6-02, 2012.
[34] W. Fangfang, W. Huazhong, C. Dongqing, and P. Yong, "Substation communication security research based on hybrid encryption of DES and RSA," in *Proc. 9th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Oct. 2013, pp. 437–441.
[35] K. H. Rahouma, "Securing software programs by applying security services with microsoft VB net programming," *Amer. J. Inf. Sci. Comput. Eng.*, vol. 2, no. 6, pp. 79–90, 2016.
[36] K. H. Rahouma, "Reviewing and applying security services with non-english letter coding to secure software applications in light of software trade-offs," *Int. J. Softw. Eng. Comput. Syst.*, vol. 3, no. 1, pp. 71–87, 2017.
[37] A. Asaduzzaman, D. Gummadi, and P. Waichal, "A promising parallel algorithm to manage the RSA decryption complexity," in *Proc. SoutheastCon*, Apr. 2015, pp. 1–5.
[38] S. Krishnan, T. Mrugalski, M. Siodelski, S. Jiang, and D. Hankins, *Guidelines for Creating New DHCPv6 Options*, document RFC 7227, 2014.
[39] Z. Pervez, M. Ahmad, A. M. Khattak, N. Ramzan, and W. A. Khan, "O S 2: Oblivious similarity based searching for encrypted data outsourced to an untrusted domain," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0179720.
[40] X. Di, J. Li, H. Qi, L. Cong, and H. Yang, "A semi-symmetric image encryption scheme based on the function projective synchronization of two hyperchaotic systems," *PLoS ONE*, vol. 12, no. 9, Sep. 2017, Art. no. e0184586.
[41] 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.
[42] F. Gont, *Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)*, document RFC 7113, 2014.
[43] K. Moriarty, B. Kaliski, J. Jonsson, and A. Rusch, *PKCS #1: RSA Cryptography Specifications Version 2.2*, document RFC 8017, 2016.
[44] R. Housley, *Using AES-CCM and AES-GCM Authenticated Encryption in the Cryptographic Message Syntax (CMS)*, document RFC 5084, 2007.
[45] R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, document RFC 8200, 2017.
[46] R. Maidhili and G. M. Karthik, "Energy efficient and secure multi-user broadcast authentication scheme in wireless sensor networks," in *Proc. Int. Conf. Comput. Commun. Inf. (ICCCI)*, 2018, pp. 1–6.
[47] S. Steffann. (2018). *DHCPKit*. GitHub. [Online]. Available: https://github.com/sjm-steffann/dhcpkit
[48] H. Eijs. (2018). *Pycryptodome*. GitHub. [Online]. Available: https://github.com/Legrandin/pycryptodome
[49] P. Biondi. (2011). *Scapy Packet Manipulator*. [Online]. Available: http//www.secdev.org/projects/scapy
[50] G. Combs. *Wireshark*. Wireshark Team. Accessed: 2017. [Online]. Available: https://www.wireshark.org/
[51] J. Cai, "A handshake protocol with unbalanced cost for wireless updating," *IEEE Access*, vol. 6, pp. 18570–18581, 2018.
[52] A. K. Al-Ani, M. Anbar, S. Manickam, and A. Al-Ani, "DAD-match; Security technique to prevent denial of service attack on duplicate address detection process in IPv6 link-local network," *PLoS ONE*, vol. 14, no. 4, 2019, Art. no. e0214518.
[53] A. K. Al-Ani, M. Anbar, S. Manickam, A. Al-Ani, and Y.-B. Leau, "Proposed DAD-match security technique based on hash function to secure duplicate address detection in IPv6 link-local network," in *Proc. Int. Conf. Inf. Technol.*, 2017, pp. 175–179.
[54] J. Bi, J. Wu, G. Yao, and F. Baker, *Source Address Validation Improvement (SAVI) Solution for DHCP*, RFC 7513, 2015.

AYMAN AL-ANI received the B.S. degree in computer engineering from the University of Technology and the M.Sc. degree in information technology from Universiti Utara Malaysia (UUM), in 2016. He is currently a Ph.D. Fellow with the School of National Advance IPv6 Center (NAv6), Universiti Sains Malaysia (USM). His research interests include computer networks, network security, software-defined networks, and the internet security.

MOHAMMED ANBAR received the Ph.D. degree in advance computer network from Universiti Sains Malaysia (USM), where he is currently a Senior Lecturer with the National Advance IPv6 Centre (IPv6). His current research interests include malware detection, web security, intrusion detection system (IDS), intrusion prevention system (IPS), network monitoring, the Internet of Things (IoT), and IPv6 security.

IZNAN HUSAINY HASBULLAH received the B.Sc. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA. He has experience working as Software Developer, Project Manager, R&D Consultant, CTO, and Security Auditor, prior to joining NAv6, in 2010. He is currently a Research Officer with the Next Generation Unified Communication Group, National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia (USM). His research interest includes unified communication, video conferencing, next generation networks, and graphical user interface (GUI), and lecture capture systems.

ROSNI ABDULLAH received the bachelor's degree in computer science and applied mathematics and the master's degree in computer science from Western Michigan University, Kalamazoo, MI, USA, in 1984 and 1986, respectively, and the Ph.D. degree in the area parallel algorithms from Loughborough University, U.K., in 1997. She is a Professor in computer science with the School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia. She is currently the Director of the National Advanced IPv6 Centre (Nav6), and also the Head of the Parallel and Distributed Processing Research Group. She has led over 20 research grants, and has published over 100 papers in journals and conference proceedings. Her current research interests are in the area of parallel algorithms on multicore and GPGPU architectures for bioinformatics applications.

AHMED K. AL-ANI received the B.S. degree in computer technique engineering from the University of Al-Ma'mun, in 2013, and the M.Sc. degree in information technology from Universiti Utara Malaysia (UUM), in 2016. He is currently pursuing the Ph.D. degree with the National Advance IPv6 Center (NAv6), Universiti Sains Malaysia (USM), Gelugor, Penang, Malaysia. He is also a Computer Engineer. His research interests include computer network security, the Internet security, network communication protocols (IPv6), and IPv6 security.