Journal of Computer and Communications, 2016, 4, 39-50
Published Online January 2016 in SciRes. http://www.scirp.org/journal/jcc
http://dx.doi.org/10.4236/jcc.2016.41005

Scientific Research Publishing

A Secure DHCP Protocol to Mitigate LAN Attacks

Osama S. Younes[1,2]

[1]Faculty of Computers and Information, Technology Tabuk University, Tabuk, KSA
[2]Faculty of Computers and Information, Menoufia University, Al Minufya, Egypt
Email: usama.younas@ci.menofia.edu.eg

Received 1 November 2015; accepted 24 January 2016; published 28 January 2016

Abstract

Network security has become more of a concern with the rapid growth and expansion of the Internet. While there are several ways to provide security in the application, transport, or network layers of a network, the data link layer (Layer 2) security has not yet been adequately addressed. Data link layer protocols used in local area networks (LANs) are not designed with security features. Dynamic host configuration protocol (DHCP) is one of the most used network protocols for host configuration that works in data link layer. DHCP is vulnerable to a number of attacks, such as the DHCP rouge server attack, DHCP starvation attack, and malicious DHCP client attack. This work introduces a new scheme called Secure DHCP (S-DHCP) to secure DHCP protocol. The proposed solution consists of two techniques. The first is the authentication and key management technique that is used for entities authentication and management of security key. It is based on using Diffie-Hellman key exchange algorithm supported by the difficulty of Elliptic Curve Discrete Logarithm Problem (ECDLP) and a strong cryptographic one-way hash function. The second technique is the message authentication technique, which uses the digital signature to authenticate the DHCP messages exchanged between the clients and server.

Keywords

DHCP, Authentication, Data Link Layer Attacks, ECDLP

1. Introduction

Evolving of computer networks, and the variety of its services and applications, has increased the users need for local area networks (LANs) [1]. LAN technologies, such as Ethernet, are the infrastructure for the Internet that everybody uses without further thought.

How to cite this paper: Younes, O.S. (2016) A Secure DHCP Protocol to Mitigate LAN Attacks. *Journal of Computer and Communications*, **4**, 39-50. http://dx.doi.org/10.4236/jcc.2016.41005

Ethernet stands as the dominant networking technology in Local Area Network (LAN), which has been widely used in campus networks, enterprise networks and data centre networks due to its simplicity and auto-configuration characteristics. Its ease of use and low cost rely on broadcast-based service or resource discovery protocols, such as Dynamic Host Configuration Protocol (DHCP), Address Resolution Protocol, Network Time Protocol, and Network Basic Input/Output System.

Network security has become more of a concern with the rapid growth and expansion of the Internet. While there are several ways to provide security in the application, transport, or network layers of a network, the data link layer (Layer 2) security has not yet been adequately addressed [2]. In local networks, security weaknesses in the data link layer enable internal attacks. Although switches and routers have some built-in security features, they are not enough to fully ensure the security of LANs. Moreover, these features require network administrators' involvement and are prone to misconfiguration. In addition, data link layer protocols used in local area networks (LANs) are not designed with security features [2].

DHCP simplifies the access to a network. When a host connects to the network, DHCP [3] automates the assignment of TCP/IP stack configuration parameters such as IP addresses, subnet masks, and default gateway. It is an internet protocol that lets network administrators centrally manage the network. Without using DHCP, the IP address must be manually assigned for each host in a network and if the host moved to a new location in the network, the IP address must be manually configured.

DHCP is one of the most used network protocols for host configuration. It was designed since a long time ago [4]-[6] and it had not major changes, although it is vulnerable to a number of attacks, such as the DHCP rouge server attack, DHCP starvation attack, and malicious DHCP client attack. The main source of these attacks is given by the fact that DHCP does not use any authentication scheme for clients, servers, or exchanged messages.

Much research [7]-[17] has been done to make DHCP protocol more secure. Most of this research is based on the work introduced in [7], which proposed a standard for DHCP message authentication, using the option field in the DHCP message, based on a shared secret key between client and server. The main drawback of the related work introduced in literature is that managing the secrete keys between clients and server did not explain. In addition, some of this research uses a digital certificate to sign the exchanged message between the client and server; however, the size of the digital certificate exceeds the DHCP message and cannot be loaded into it. Few schemes [15] [16] were introduced for key management between clients and server. However, these schemes do not scale up with number of clients and make a central point of failure.

In this work, a new scheme called S-DHCP (Secure DHCP) is proposed to secure DHCP protocol. S-DHCP is an authentication technique, based on an extension of the DHCP protocol, which is used to mutually authenticate the DHCP client and DHCP server as well as to protect the integrity of the DHCP messages. The proposed solution is based on two techniques: entity authentication and key management (EAKM) technique, and message authentication (MA) technique. EAKM technique is based on using Diffie-Hellman key exchange algorithm supported by the difficulty of Elliptic Curve Discrete Logarithm Problem (ECDLP) and a strong cryptographic one-way hash function. It is used to mutually authenticate the client and the DHCP server, exchange the session key, and generate a digital signature for each client and the DHCP server. The MA technique uses the digital signatures generated using the EAKM technique to authenticate the DHCP messages exchanged between the client and server.

The paper is organized as follows. Section 2 describes the functions of the DHCP protocol. DHCP security issues are discussed in Section 3. Related work is discussed in Section 4. The entity authentication technique is described in Section 5. Then, Section 6 shows how message authentication technique uses a digital signature to authenticate the DHCP messages. After that, Section 7 analyses the security problems of DHCP according to the proposed solution. Finally, section analyses the performance of the proposed techniques.

2. Dynamic Host Configuration Protocol

DHCP [3] Stands for Dynamic Host Configuration Protocol. It allows a host to obtain an IP address dynamically when it connects to the network. This service automates the assignment of IP addresses, subnet masks, gateway and other IP networking parameters. DHCP distributed addresses are not permanently assigned to hosts but are only leased for a period of time. If the host is powered down or taken off the network, the address is returned to the range for reuse. This is especially helpful with mobile users that come and go on a network.

The operation of DHCP protocol is shown in **Figure 1**. When a DHCP-configured device boots up or connects to the network, the client broadcasts a DHCPDISCOVER packet to identify any available DHCP servers

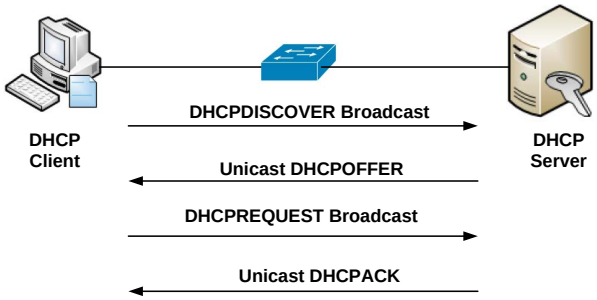

Figure 1. DHCP protocol operations.

on the network. A DHCP server replies with a DHCPOFFER, which is a lease offer message with an assigned IP address, subnet mask, DNS server, and default gateway information as well as the duration of the lease.

The client may receive multiple DHCPOFFER packets if there is more than one DHCP server on the local network, so it must choose between them, and broadcast a DHCPREQUEST packet that identifies the explicit server and lease offer that the client is accepting. A client may choose to request an address that it had previously been allocated by the server. Assuming that the IP address requested by the client or offered by the server is still valid, the server would return a DHCPACK message that acknowledges to the client the lease finalization.

3. DHCP Security Problems

3.1. DHCP Exhaustion Attack

The DHCP server has a pool of IP addresses that are being leased to hosts; but the IP address pool has always limited number of IP addresses. In DHCP exhaustion attack [2], the attacker exhausts the IP addresses in the DHCP address pool. The DHCP server happily hands out the entire set of addresses available to the client's network, because it has no way to differentiate between genuine host and spoofed one. If a legitimate client tries to obtain an IP address, it will have no IP connectivity because the entire of addresses have been allocated to the spoofed clients.

3.2. DHCP Rogue Server Attack

The rogue server is a DHCP server on a network which is not under the administrative control of the network staff. The DHCP rogue server attack [2] is a famous LAN attack in which a malicious user disguises itself as a DHCP server and responds to DHCP requests with a bogus IP address. When clients connect to the network, both the rogue and legal DHCP server receive the DHCP DISCOVER message; then both servers will offer them IP addresses as well as default gateway. The DHCP rogue server responds to DHCP requests with wrong configuration. The wrong information may be wrong default gateway, wrong DNS server, or wrong IP address. When the attacker host (a Rogue DHCP server) make itself a default gateway, it can receive the whole traffic of the network. So, he can analyse and modify all packets sent from the attacked machine and he can steal passwords and privacy information.

3.3. Malicious DHCP Client

A malicious client can gain unauthorized access in a network and then it can use the network services without being allowed to. More, it can launch a DHCP starvation attack, exhausting the DHCP server available IP addresses. An unauthorized client can also install and configure a rogue DHCP server and then realize the attacks specific to an illegitimate DHCP server.

4. Related Work

In RFC 3118 [7], two standard techniques used the option field for authentication of DHCP messages; configuration token and delayed authentication. The configuration token scheme is based on sharing a secret token between the client and server. This scheme can only protect DHCP server that has inadvertently been instantiated.

It does not support DHCP message authentication. The delayed authentication scheme uses the HMAC technique for DHCP message authentication. It uses a pre-shared secret key with MD5 message-digest algorithm to generate the MAC (Message Authentication Code) for DHCP messages. Although this scheme tried to solve the problem of DHCP entity authentication and message authentication using a shared secrete key, it did not show how the secrete key is managed, especially for a large number of clients. In addition, the security of the MD5 hash function is severely compromised. A collision attack exists that can find collisions in seconds using a desktop computer [8].

Using the option field for DHCP message authentication is limited because its maximum size is 255 bytes (the length field is one byte). To encode options larger than 255 bytes, a technique was introduced in [9]. The authors used *sname* and *file* fields to store long options using a technique called aggregate buffer.

In [10], the method introduced in [7] has been developed for securing DHCP using a digital certificate. The authors used a trusted server to distribute the digital certificates between the clients and server. Because of its size, the digital certificate cannot fit into one DHCP message. Therefore, the authors introduced a scheme depends on fragmentation of the DHCP message into many messages.

Based on authentication option specification introduced in [7], a method based on using X.509 digital certificate was introduced in [11]. The digital certificate is used to sign the DHCP messages transmitted between clients and server. A common trust server or authority is used to distribute the digital certificate. The authors showed that the digital certificate cannot be loaded in one DHCP message because of its large size. In addition, they indicated that the digital certificates revocation policies are hard to set up. Also, the authors recommended that using certificate based authentication with delay authentication.

Two methods were proposed in [12] to make DHCP more secure; Secure DHCP with Digital Certificates (SDDC) and Secure DHCP with Shared Secrets (SDSS). The SDDC method depends on using digital certificates to authenticate the DHCP messages and entities. However, this method did not take into account the size of digital certificate may exceed the size of the DHCP message. The SDSS method uses a secrete key shared between entities to authenticate the DHCP messages.

Another mechanism called Challenge Handshake Authentication Protocol (CHAP) was introduced in [13] for authenticating the entities using DHCP. This mechanism lets the server generates an encrypted challenge response and attach it to the DHCPOFFER message when it receives the DHCPDISCOVER message. The challenge response is computed using a hash function and a secret key that must be shared between the DHCP clients and server. Also, Other DHCP messages; DHCPREQUEST and DHCPACK, are attached with the challenge response to authenticate the messages and entities. Although, this work did not change the state machine of the client, it introduced a new server; called authentication server, with additional communication between it and the DHCP server. In addition, the authors did not show how the secret key is managed between the servers and clients.

In [14], a method for authentication of clients and DHCP message was proposed based on Kerberos V [15]. The proposed method used the authentication option specification defined in [7]. Kerberos V protocol uses an authentication server and key distribution centre, which issues tickets for the clients and server that are encrypted using secret keys shared between the clients and server. To authenticate the DHCP request and response, the ticket of the client or server is included in the DHCP message. The main disadvantages of this method are that the Kerberos server represents a single point of failure. In addition, the tickets use the timestamp to prevent replay attack, which means that the clock of clients and servers should be synchronized.

To provide mutual authentication for DHCP clients and server and for DHCP message authentication, a scheme was proposed in [16]. This scheme is similar to schemes introduced in [7] [9] [10]. However, this scheme presented a methodology for sharing secrete keys between entities, which is used to compute MAC (Message Authentication Code) for DHCP messages. The scheme supposes that there is a shared secret key between entities, which is used to generate a secret key for each DHCP message using a generated random number. Unfortunately, the authors did not show how the random number is generated and how the new secrete key is generated from the old secret key and the random number.

The authors in [17] proposed a security system that consists of eight units that make only authorized clients to obtain IP addresses. Before connecting to the system, the client must get an ID and a password from the authentication server. This system uses additional message and is not compatible with the standard. In addition, the proposed scheme for managing the shared secret key between clients and the authentication server does not scale up with number of clients and represents central point of failure.

5. Entity Authentication and Key Management Technique (EAKM)

This section explains the proposed authentication and key management scheme. It is based on Diffie-Hellman key exchange algorithm supported by the difficulty of Elliptic Curve Discrete Logarithm Problem (ECDLP) and based on using a strong cryptographic one-way hash function. The significant advantage of ECDLP is the faster calculations compared to other public key cryptosystems with the same security levels. The EAKM technique is inspired by the work introduced in [18]. **Table 1** shows all notations used for the proposed scheme. The overall scheme is illustrated in **Figure 2** and is explained as follows. The proposed scheme consists of three phases: the setup phase, the registration phase, and the authentication phase.

5.1. Setup Phase

In this phase, the clients and server agree on the used elliptic curve parameters. The sever chooses a secrete key

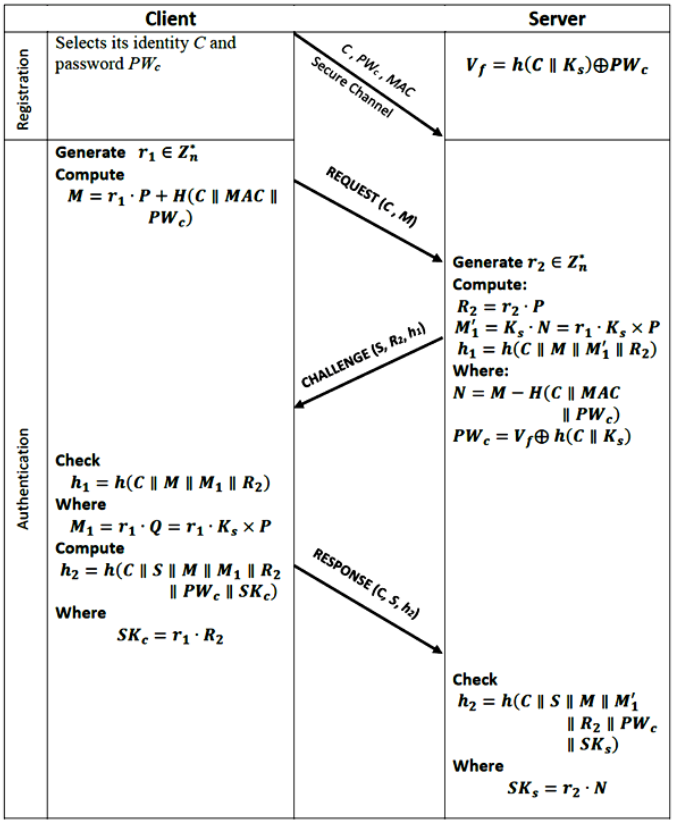

Figure 2. Flow diagram for EAKM technique.

Table 1. The notations used for proposed authentication scheme.

C and *S*	The ID of the client and server
PW_c	The password of the client
$h(\cdot)$	Strong cryptographic one-way hash function
$H(\cdot)$	Function which makes a point map to another point on elliptic curve
$\|$	Concatenation operation
$\oplus$	Exclusive-or operation
K_s	Server password
SK	Session key
Q	Server public key

Ks, then it selects a point *P* in the elliptic curve to compute the public key $Q = Ks \cdot P$. The server publishes all parameters; *P*, *a*, *b*, *n*, *h*, and *Q*, except the secret key *Ks*.

5.2. Registration Phase

As shown in **Figure 2**, the registration phase consists of two steps:

1) The user or the network administrator chooses the identity *C* and password PW_c for the client. Then, *C*, PW_c, and MAC address of the client are sent to the server over a secure channel, such as a VPN (Virtual Private Network) or SSL (Secure Sockets Layer), or the network administrator insert them manually to the server database.

2) The server computes the password verification code $V_f = h(C \| K_s) \oplus PW_c$ using a strong cryptographic hash function *h* (.) and the server password *Ks*; then it stores the binding (C, V_f, MAC) in its database.

5.3. Authentication Phase

When any client *C* tries to connect to the local area network managed by the authentication server *S*, the authentication procedure between *C* and *S* proceeds as shown in **Figure 2**, which is illustrated as follows:

Step 1: The client generates the random number r_1, and then it computes the parameter $M = r_1 \cdot P + H(C \| PW_c)$ using the parameters *P* and *H* published by the server. The parameter *M*, the client ID (*C*), and the client MAC address are sent to the server in the request message REQUEST (*C*, *MAC*, *M*).

Step 2: If the server received the request message, it checks that the client identifier *C* and the client *MAC* address are stored in its database. If the server did not find *C* and *MAC* address the database, it ignores the request message. Otherwise, the server generates a random number r_2 and use the password verification code V_f, bended with *C* and stored in the database, and the sent parameter *M* to compute the client password PW_c and parameter *N*, M_1' and R_2, which are used to compute the parameter h_1, as shown in **Figure 2**. After calculating the parameter R_2 and h_1, the serve sends them with its identity *S* in a challenge message CHALLENGE (*S*, R_2, h_1).

Step 3: After receiving the challenge message, the client compute the value of $M_1 = r_1 \cdot Q = r_1 \cdot K_s \times P$, which is used to check the value of $h_1 = h(C \| M \| M_1 \| R_2)$. If the computed value for h_1 does not match with the received value, the session between the client and server is terminated. Otherwise, the client computes SK_c, which is used to compute h_2, as shown in **Figure 2**. At the end, the client sends the response message RESPONSE (*C*, *S*, h_2) to the server.

Step 4: In the same way, once the server receives the response message, it verifies the value h_2. First, it computes SK_s. Then it computes $h_2 = h(C \| S \| M \| M_1' \| R_2 \| PW_c \| SK_s)$. If the calculated value for h_2 is equal to the received one in the response message, the server authenticate the client *C* with the common and unique session key $SK = SK_c = SK_s = r_1 \cdot r_2 \cdot P = r_1 \cdot R_2 = r_2 \cdot N$.

As clear from **Figure 2**, a one way hash function with XOR operation is used for encrypting or decrypting all communication messages. Therefore, the computation cost (or computational overhead) is quite low, as explained in Section 8. At the end of the EAKM technique, the digital signature of each client and the server are generated and stored in each of them to be used by the S-DHCP technique, which is explained in the next section.

6. Message Authentication Technique

S-DHCP uses the message authentication technique, which extends DHCP with an integrity/authentication technique for DHCP packets, to mitigate different DHCP attacks. Because S-DHCP is built on top of DHCP, its specification (as for message exchange, timeout, cache) follows the original one for DHCP [4]. Therefore, network clients that do not use S-DHCP module can process the S-DHCP messages. However, in secure LAN, all clients should use S-DHCP. This section describes the message authentication technique.

6.1. Protocol Design Constraints

In order to make the proposed MA technique integrated and compatible with the current DHCP implementation, the following design constraints are adopted:

- The proposed scheme must follow the authentication option format introduced in [7];
- The DHCP client and server state machine must not be changed, *i.e.* any new state must not be introduced;

- Introducing any new DHCP message is not allowed;
- The clients that do not use proposed protocol must be able to obtain configuration information for the server.

In addition to the above, one of the main constrains is that DHCP specification does not permit the fragmentation of DHCP messages. This means that the maximum size of DHCP packet is 1500 bytes; the MTU for Ethernet. By subtracting the length of IP header, UDP header, and DHCP header, the maximum size of option filed in DHCP message is 1236 bytes, which is too short. It cannot allow the transmission of a long encryption key or a digital certificate.

6.2. Protocol Description

Figure 3 shows the DHCP message exchange when S-DHCP is enabled in clients and server. Each DHCP message sent by client or server are processed by the MA scheme before being sent through the network. In the initial state, the DHCP client broadcasts a request, which is DISCOVERY message, searching for DHCP server. The client indicates the use of S-DHCP in option filed in the message. To use the option field in DHCP messages, the format defined in RFC 3118 is followed, which was proposed for entity and message authentication for DHCP protocol. The option field is used as follows:

- For the Protocol field the new value 4 is used which indicates that the authentication information is attached to message;
- For the Algorithm field a new value 5 is used to mark the using of S-DHCP;
- For RDM (Replay Detection Method) field the same value is used as in the specification in RFC 3118;
- For the Replay Detection (RD) field, it contains a monotone strictly increasing counter containing the current time and date which can reduce the danger of replay attacks.

If the DHCP server receives a DHCPDISCOVER message from a client, it passes the message to the DHCP engine if the message's option filed indicates that the protocol S-DHCP is not enabled. Otherwise, the MA scheme processes the message before passing it to DHCP engine. It verifies the value of Replay Detection field. If the verification process fails, the message is dropped. Otherwise, the server starts to prepare the DHCPOFFER message. The server reserves an IP address for the client and makes a lease offer by sending a DHCPOFFER message to the client. This message contains the following:

- The MAC address of the client;
- The offered IP address to the client (*IPc*);
- The IP address of the DHCP server (*IPs*);
- Lease duration;
- The gate way (GW).

In addition, to authenticate and authorize the DHCPOFFER message, the server signature is attached to DHCPOFFER message in the option filed. As shown in **Figure 3**, the server signature attached to DHCPDISCOVER message includes the secrete key shared with the client, the MAC address of the client, *IPc*, *IPs*, GW, and RD. Also, to mitigate the replay attack, the Replay Detection filed is updated with the correct value.

When the client receives the DHCPOFFER message, the MA scheme passes it to the DHCP engine or process it, depending on the authentication option filed is set or not. If this option is set, the MA scheme checks the RD

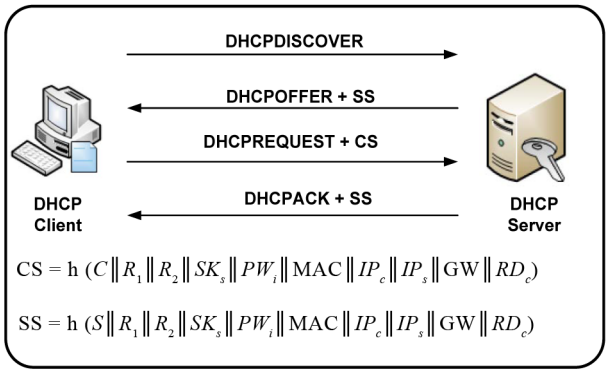

Figure 3. The DHCP message exchange suing MA technique.

filed. If the value of RD is strictly higher than the old value, the processing continues. Otherwise, the message is discarded.

After checking the RD field, the MA scheme verifies the signature of the server and the integrity of the message. First, it extracts the fields from the message required for signature verification, which are MAC, *IPs*, *IPc*, GW, and RD. Then, it calculates *SS*, as shown in **Figure 3**. If the calculated *SS* equals to the sever signature attached to the DHCPOFFER message, and MAC and *IPc* equal to that of the client, it starts to construct the DHCPREQUEST. Otherwise, the message is discarded.

The DHCPREQUEST message is a response to the DHCPOFFER message and it is used to request the offered IP address. Most fields of DHCPREQUEST messages are the same as that of the DHCPOFFER message. The client chooses a new value for Replay Detection filed and update it in the message. To authenticate and authorize the DHCPREQUEST message, the signature of the client is attached to this message, which is shown in **Figure 3**. The CS is generated using the new value for the RD field.

After receiving the DHCPREQUEST message, the DHCP server verifies the RD field. Then it validates the client signature. It extracts the required fields for validation; RD, *IPc*, *IPs*, *GW*, and MAC. Then it generate the signature CS and compare it with the one attached to the DHCPREQUEST message. If the RD field and signature of the client passes the validation procedure, the server starts to prepare the DHCPACK message in the same way as DHCPOFFER message explained above. The DHCPACK message includes all configuration information that the client have requested. When the client receives the DHCPACK message, it must validate it in the same way as explained for the DHCPOFFER message. After validation of the DHCPACK message, the DHCP client starts to configure its network interface with the negotiated parameters.

7. Analysis of Security Threats

This section discusses how the proposed authentication and key management scheme, and MA scheme mitigate main security vulnerabilities that a DHCP server or client can encounter in wireless or wired LAN.

7.1. DHCP Rouge Server Attack

The rogue DHCP server is a DHCP server, set up on a network by an attacker, which is not under the control of network administrators. By placing a rogue DHCP server on the network, the attacker can supply its own system as the default gateway and DNS server resulting in a man-in-the-middle attack.

The proposed S-DHCP scheme mitigates the DHCP rouge server attack because the DHCP server authenticate itself to the client during the authentication process and during DHCP message exchange. During the authentication process, the server is authenticated using the parameter h_1 that depends on v_f and M_1'. To generate M_1', the attacker must know the client password PW_c, *Ks*, and r_1. Even if the attacker compromised the password of the client, he cannot generate V_f without compromising the secret K_s (which is a high entropy number) and guessing r_1 which are difficult especially with maintaining the secret K_s periodically. On the other hand, if the attacker steals the secret K_s, he needs the password verifier to access the client password.

After the authentication process and during exchanging DHCP messages, if an authenticated client tries to impersonate the DHCP server, it must have the signature of the legitimate DHCP server. To do that, the attacker must compromise PW_c and SK_s. Also, he must guesses r_1 and r_2 assuming that he has R_1 and R_2. To compute SK_s, given the public key *Q*, the attacker must break the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is very difficult problem. Also, given R_1, R_2 and *P*, guessing r_1 and r_2 is difficult and cannot be done in polynomial time. In addition, online password guessing cannot succeed, since *Ks* is a high entropy number that cannot be guessed and the password verifier V_f is not available to attacker.

7.2. DHCP Exhaustion Attack

DHCP starvation attack is designed to deplete all of the addresses within the address space allocated by the DHCP server. The attacker floods the DHCP server with DHCPREQUEST packets of spoofed MAC addresses. Therefore, clients of the victim network are denied an IP address requested via DHCP and thus is not able to access the network. Hence, DHCP starvation can be classified as a denial of service attack. The network attacker can then set up a rogue DHCP server on the network and perform the man in the middle attack, or simply set their machine as the default gateway and can sniff packets.

The proposed scheme can mitigate the DHCP starvation attack. For any received DHCPREQUEST message from any client, the DHCP server verifies the client signature attached to the message. If the DHCPREQUEST message does not contains the client signature, the server that uses S-DHCP discards the request. Otherwise, the server extract the parameters *C*, MAC, *IPc*, *IPs*, and *RDc* form the message header. Then, it uses the parameters *C*, R_1, R_2, *SKs*, and *PWc* bound with *C* and MAC address in its database to generate the signature of the client. After that, it compares between the attached client signature and the generated signature. If there any difference between them, the request message is discarded. Otherwise, the request is accepted.

If the attacker changed the MAC address in the header of the DHCPREQUEST message to initiate the DHCP starvation attack and attached any client signature to the message, the DHCP server will detect the difference between the attached client signature and generated signature. The attacker can do the DHCP starvation attack if and only if he has the signatures of many clients. This means he has to guess the parameters R_1, R_2, *SKs*, and *PWc*, supposing that he knows the parameters *C* and MAC. As explained above, guessing of all these parameters for one client is extremely difficulty. However, to do that for a large number of clients is nearly impossible.

7.3. Malicious User Attack

In the malicious user attack, also called spoofing or masquerading attack, the attacker impersonate the identity of a legitimate user in an illegal manner. The proposed authentication scheme resists against the malicious user attack. To impersonate the user *C*, the attacker must know the parameters *PWc*, *C*, MAC, and *M* of the client. Even if the attacker compromised the password of the user *PWc*, he cannot compute the correct value of *M* because he cannot guess the random number r_1.

7.4. Stolen-Verifier Attack

In the stolen-verifier attack, an attacker may attack the server and steal the database that contains clients' information such as the ID and password verification code. Then the attacker tries to use this information to impersonate the legitimate user. The proposed authentication scheme mitigates this attack, because the attacker cannot compute or guess the client's password even if he has the client ID and password verification code. This because he cannot get the server secret *Ks* which is not stored in the database.

7.5. Brute-Force Password-Cracking Attack

In the brute-force password-cracking attack, an opponent tries repeatedly to guess for the secret parameters such as the password of the client or the server and check them against an available cryptographic hash of the secret parameters. The proposed scheme mitigates this attack because, even if the opponent intercept all exchanged messages by a passive attack and correctly guesses the password, he cannot guess other secret parameters (*M*, R_2, M_1, M_1'). This is because these parameters are function mapped points to an EC point, which is a very difficult problem.

8. Performance Evaluation

This section shows the results of testing the proposed client and message authentication schemes. For performance evaluation of the proposed sachems, they have been implemented using C language with the open source cryptographic library (Open SSL) [19]. All performance tests are conducted using Ubuntu virtual machines with 3.6 GHz processor and 2 GB of RAM.

The client authentication time (CAT), DHCP message authentication time (MAT), and completion time (CT) are the metrics used for performance evaluation of the proposed schemes. CAT is the time taken to authenticate a client, which is the time needed for processing the request, challenges and response messages. The MAT time is the time needed for processing and exchanging the DHCPDISCOVER, DHCPOFFER, DHCPREQUEST, and DHCPACK messages between the client and server. The time needed for receiving the network configuration parameter from DHCP server is called the completion time. CT equals to the sum of CAT and MAT.

In all experiments, to measure MAT, CAT, and CT; the experiment is repeated many times until 95% confidence interval with 2% error is reached. For repeated experiments and for each metric, the average of the measured values is computed to be used in the analysis and comparisons.

In the first experiment, the number of clients (*N*) that try to access the local area network at the same instant is

increased from 5 to 40 clients. In addition, it is supposed that the network is under attacks and the number of attackers is *t* percent of all nodes, where *t* is set to 0%, 20%, 40%, and 60%. The attacker impersonates the identity of the legitimate user and tries to deceive the server to get an offer. In this experiment, the metrics CAT, MAT, and CT are measured and the results are shown in **Figures 4-6**, respectively.

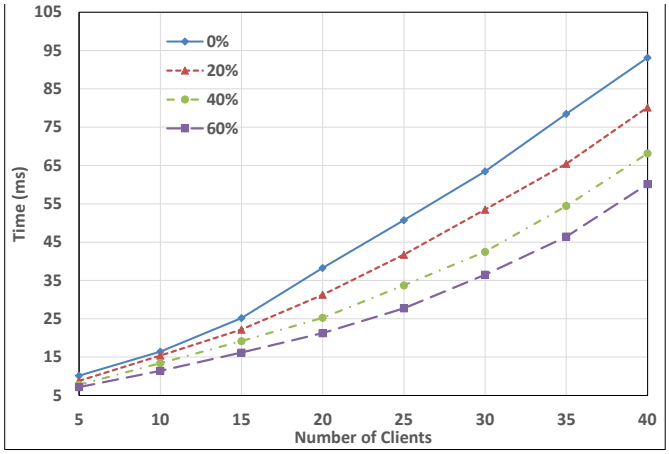

Figure 4. CAT versus *N* with increasing number of attackers.

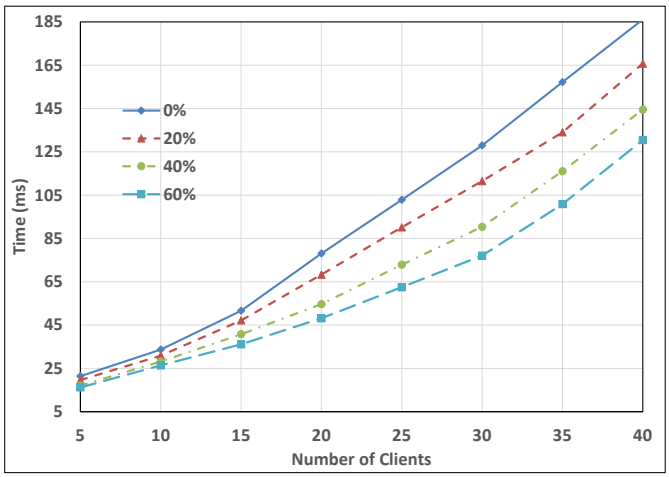

Figure 5. MAT versus *N* with increasing number of attackers.

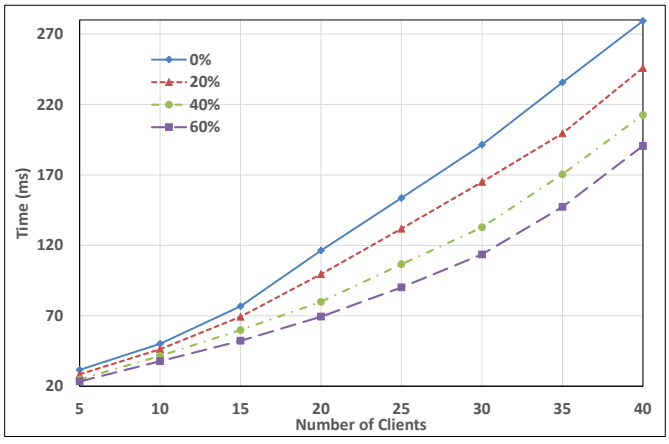

Figure 6. CT versus *N* with increasing number of attackers.

As shown in **Figures 4-6**, the higher the number of clients the greater the measured value for CAT, MAT, and CT. This is because increasing the number of client increases the time needed for processing messages in the server for either client authentication or message authentication. In addition, for the same number of authenticated clients, it is clear that increasing *t* decreases the value of CAT, MAT, and CT. This because increasing *t* increases the number of attackers and decreases the number of legitimate users. In addition, for attackers, the sever does not process the DHCPREQUEST and DHCPOFFER messages, as explained in the last section. This indicates the effectiveness of the proposed scheme because it prevents the attacker to complete the attack.

To measure the overhead of using the proposed scheme (S-DHCP), MAT is measured in the case of using S-DHCP and standard protocol (DHCP). As the first experiment, *N* is varied between 5 and 40, where there are no attackers in the network. **Figure 7** shows the results of this experiment. As shown in the figure, the overhead increased from 7.8 ms to 66.2 ms with increasing *N* from 5 to 40. Although the proposed scheme makes delay in the process of accessing the network by this overhead, it makes the network more secure by mitigating many attacks, as explained in Section 7. In addition, it is clear that overhead is very small.

Figure 8 compares between the proposed work and the work introduced in [12], which uses the digital certificates to authenticate the DHCP messages. For different number of nodes, the authentication time is measured, where the length of the key used to encrypt the digital certificate is 2048 and 3072 bits. As shown in **Figure 8**, the authentication time in the case of using digital certificate is much higher than that in the case of using the proposed scheme especially with a large number of clients.

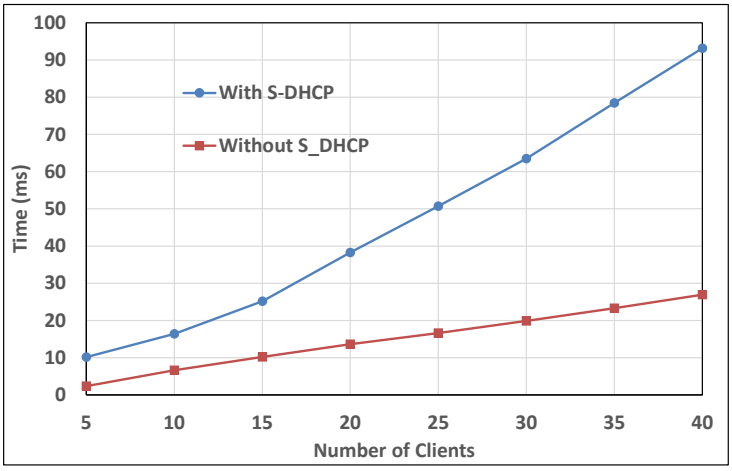

Figure 7. MAT versus *N* with in the case of using DHCP and S-DHCP.

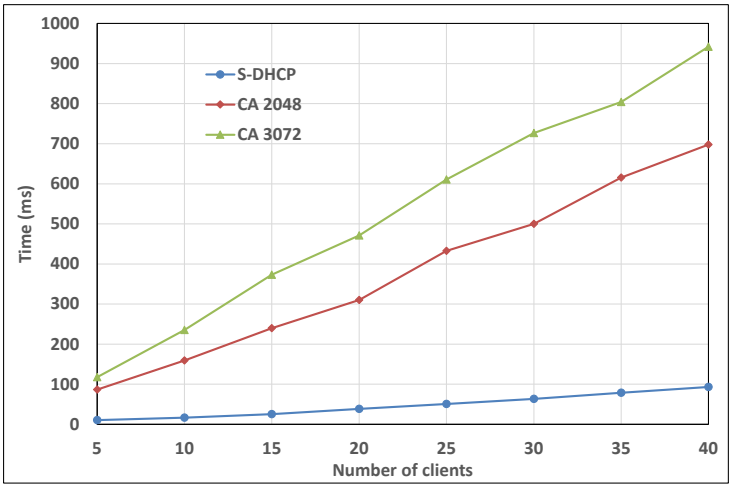

Figure 8. MAT versus *N* with in the case of using digital certificate and S-DHCP.

9. Conclusion

Dynamic host configuration protocol (DHCP) is one of the most used network protocols for host configuration that works in data link layer. DHCP is vulnerable to a number of attacks, such as the DHCP rouge server attack, DHCP starvation attack, and malicious DHCP client attack. This work introduces a new scheme called Secure DHCP (S-DHCP) to secure DHCP protocol. The proposed solution consists of two techniques. The first is the authentication and key management technique that is used for entities authentication and management of security key. It is based on using Diffie-Hellman key exchange algorithm supported by the difficulty of Elliptic Curve Discrete Logarithm Problem (ECDLP) and a strong cryptographic one-way hash function. The second technique is the message authentication technique, which uses the digital signature to authenticate the DHCP messages exchanged between the clients and server. Security analysis of the proposed scheme showed that it mitigates main security vulnerabilities that a DHCP server or client can encounter in wireless or wired LAN. In addition, performance analysis of the proposed scheme indicated that its performance is better compared to the digital certificate scheme introduced in literature.

References

[1] Hu, X.D., Gao, Z. and Li, W. (2009) Research on the Switched LAN Monitor Mechanism and its Implementation Method Based on ARP Spoofing. *International Conference on Management and Service Science*, Wuhan, 20-22 September 2009, 1-4. http://dx.doi.org/10.1109/icmss.2009.5304207

[2] Altunbasak, H.C. (2006) Layer 2 Security Inter-Layering in Networks. Georgia Institute of Technology, Atlanta.

[3] The TCP/IP Guide. DHCP Security Issues. http://www.tcpipguide.com/free/t_DHCPSecurityIssues.htm

[4] Droms, R. (1997) Dynamic Host Configuration Protocol. RFC 2131.

[5] Droms, R. and Lemon, T. (2002) The DHCP Handbook. 2nd Edition, Sams Publishing, Carmel, Indiana.

[6] Alexander, S. and Droms, R. (1997) DHCP Options and BOOTP Vendor Extensions. RFC 2132.

[7] Droms, R. and Arbaugh, W. (2001) Authentication for DHCP Messages. RFC 3118.

[8] Stevens, M.M.J. (2007) On Collisions for MD5. Master Thesis, Eindhoven University of Technology, Eindhoven.

[9] Lemon, T. and Cheshire, S. (2002) Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4). RFC 3396.

[10] Xu, Y., Manning, S. and Wong, M. (2011) An Authentication Method Based on Certificate for DHCP. DHC Internet Draft.

[11] Glazer, G., Hussey, C. and Shea, R. (2003) Certificate-Based Authentication for DHCP.

[12] Duangphasuk, S., Kungpisdan, S. and Hankla, S. (2011) Design and Implementation of Improved Security Protocols for DHCP Using Digital Certificates. ICON, Singapore.

[13] De Graaf, K., Liddy, J., Raison, P., Scano, J.C. and Wadhwa, S. (2011) Dynamic Host Configuration Protocol (DHCP) Authentication Using Challenge Handshake Authentication Protocol (CHAP) Challenge. United States Patent Application 20110154440.

[14] Hornstein, K., Lemon, T., Adoba, B. and Trostle, J. (2001) DHCP Authentication via Kerberos V. IETF DHC Working Group.

[15] Ricciardi, F. (2007) Kerberos Protocol Tutorial. The National Institute of Nuclear Physics Computing and Network Services, LECCE, Italy.

[16] Ju, H. and Han, J.W. (2005) DHCP Message Authentication with an Effective Key Management. *Proceedings of the World Academy of Science, Engineering and Technology*, **8**.

[17] Komori, T. and Saito, T. (2002) The Secure DHCP System with User Authentication. *Proceedings of the* 27*th Annual IEEE Conference on Local Computer Networks*, Tampa, 6-8 November 2002. http://dx.doi.org/10.1109/lcn.2002.1181774

[18] Yang, C., Wang, R. and Liu, W. (2005) Secure Authentication Scheme for Session Initiation Protocol. *Computers & Security*, **24**, 381-386. http://dx.doi.org/10.1016/j.cose.2004.10.007

[19] Open SSL. Open SSL Project. http://www.openssl.org/