Sādhanā Vol. 42, No. 12, December 2017, pp. 2041–2053
https://doi.org/10.1007/s12046-017-0749-y

Securing ARP and DHCP for mitigating link layer attacks

OSAMA S YOUNES[1,2]

[1]Faculty of Computers and Information Technology, University of Tabuk, Tabuk, Saudi Arabia
[2]Faculty of Computers and Information, Menoufia University, Menoufia, Egypt
e-mail: usama.younas@ci.menofia.edu.eg

MS received 22 December 2016; revised 19 March 2017; accepted 4 May 2017; published online 24 November 2017

Abstract. Network security has become a concern with the rapid growth and expansion of the Internet. While there are several ways to provide security for communications at the application, transport, or network layers, the data link layer security has not yet been adequately addressed. Dynamic Host Configuration Protocol (DHCP) and Address Resolution Protocol (ARP) are link layer protocols that are essential for network operation. They were designed without any security features. Therefore, they are vulnerable to a number of attacks such as the rogue DHCP server, DHCP starvation, host impersonation, man-in-the-middle, and denial of service attacks. Vulnerabilities in ARP and DHCP threaten the operation of any network. The existing solutions to secure ARP and DHCP could not mitigate DHCP starvation and host impersonation attacks. This work introduces a new solution to secure ARP and DHCP for preventing and mitigating these LAN attacks. The proposed solution provides integrity and authenticity for ARP and DHCP messages. Security properties and performance of the proposed schemes are investigated and compared to other related schemes.

Keywords. Network security; DHCP security; ARP security; ARP spoofing; DHCP starvation; rogue DHCP server.

1. Introduction

Due to the rapid growth and expansion of the Internet, Local Area Network (LAN) security has attracted more concern. There are many schemes to provide security in different network layers, such as network layer, transport layer, and application layer. However, data link layer security was not sufficiently addressed [1]. Most data link layer protocols used to manage LANs are designed without security features. The security weaknesses of data link layer protocols in LANs enable many dangerous attacks. Although routers and switches used in LANs have many built-in security features, they do not fully assure the security for LANs. In addition, these features need to be configured by the network administrator, which makes them prone to misconfiguration.

Dynamic Host Configuration Protocol (DHCP) [2] and Address Resolution Protocol (ARP) [3] are link layer protocols that are essential for LAN operation. DHCP simplifies the access to a network. When a host connects to the network, DHCP automates the assignment of configuration parameters of TCP/IP stack, including the default gateway, subnet mask, and IP addresses. ARP is used to resolve the MAC address of a device connected to the network whose IP address is known. ARP and DHCP were designed previously [2, 3]. Although these protocols have many security vulnerabilities, they had not had major changes.

DHCP is vulnerable to a number of attacks, including the rogue DHCP server, DHCP starvation, and malicious DHCP client attacks. These attacks are mainly because DHCP cannot authenticate clients, servers, or exchanged messages. Due to the stateless property and lack of integrity and authenticity, ARP has some serious inherent security vulnerabilities, such as ARP spoofing, Man-in-the-Middle (MITM) attack [4], Denial of Service (DoS) attack [5], and host impersonation attack [6].

Several solutions were proposed to target security issues in ARP. These techniques can be classified into two categories: non-cryptographic [7–18] and cryptographic [19–22]. However, most of these techniques have many limitations. Most non-cryptographic techniques do not prevent ARP attacks, but they only detect them. Cryptographic techniques are more effective in mitigating ARP attacks than non-cryptographic techniques. However, these techniques are vulnerable to the host impersonation and MITM attacks, as explained in section 6. Moreover, some of them do not support dynamic IP addressing. In addition, security issues of DHCP affect the security of ARP. As explained in section 6, the cryptographic techniques proposed in the literature for securing ARP, which support dynamic IP addressing, are vulnerable to DoS attack for not considering DHCP security issues. Therefore, to secure ARP, DHCP must be protected.

Many studies to secure DHCP have been proposed in the literature [23–32]. Most of these studies are based on the work introduced in [23], which proposed a standard for DHCP message authentication, based on a shared secret key between clients and the DHCP server, using the option field in the DHCP message. The main drawback of these studies is that they cannot mitigate the DHCP starvation attack and the rogue DHCP server attack.

This work introduces a new solution to secure the communication in LANs, including DHCP and ARP. The proposed solution is based on two techniques: Protect DHCP (P-DHCP) and Protect ARP (P-ARP). P-DHCP is used to secure DHCP. It provides confidentiality and authentication for DHCP clients and the DHCP server and protects the DHCP messages. P-ARP technique provides integrity and authenticity for ARP messages to guarantee that they do not change and that an authentic user has sent them. In addition, it converts ARP from a stateless to stateful protocol.

The proposed solution depends on applying cryptography to communications at the data link layer, which is common in upper layers. IPsec [33] and Transport Layer Security (TLS) [34] are well-proven protocols that provide authentication and data integrity using cryptography for communications at the network and transport layer. The proposed solution mitigates many link layer attacks, including the rogue DHCP server, DHCP exhaustion, malicious user, host impersonation, ARP Spoofing, man-in-the-middle, and denial of service attacks. The proposed techniques are compatible with standard protocols, simple, and cost-effective.

The rest of the paper is organised as follows. Section 2 provides a background for ARP and DHCP and their vulnerabilities. Section 3 discusses the pros and cons of different schemes introduced in the literature to secure ARP and DHCP. Sections 4 and 5 introduce the architecture and operation of P-DHCP and P-ARP, respectively. Section 6 presents the security analysis of the proposed schemes. Section 7 discusses the performance of the proposed schemes. Finally, section 8 provides the conclusion and future work.

2. Background

2.1 *Dynamic host configuration protocol*

DHCP allows a host to obtain an IP address dynamically when it connects to the network. This service automates the assignment of the IP address, subnet mask, 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 for mobile users that come and go on a network.

Figure 1 shows the operation of DHCP. When a host configured with DHCP is connected to the network, it sends

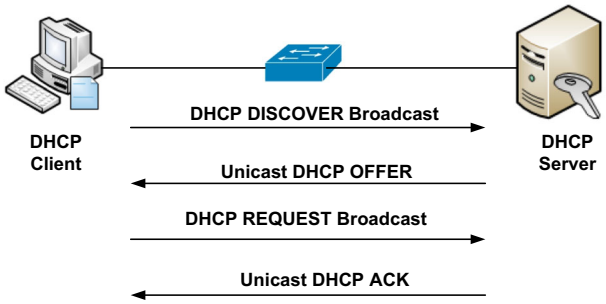


Figure 1. DHCP operations.

a broadcast message, called DHCP DISCOVER message, to discover the availability of the DHCP server on the network. If a DHCP server is available, it replies with a DHCP OFFER including the offered parameters; IP address, DNS server, subnet mask, and default gateway, as well as the leased period.

The client may receive multiple DHCP OFFER packets if there is more than one DHCP server on the local network. Therefore, it should choose one of these offers. Then, it broadcasts a message that identifies the accepted DHCP server and the details of lease offer, which is called DHCP REQUEST. After the server receives the DHCP REQUEST, if the IP address offered by the server is still available, it sends back a DHCP ACK message to the host to confirm the end of the lease process.

When DHCP was designed several years ago, security issues were not considered. Hence, the DHCP lacks some security mechanisms and is vulnerable to many attacks, such as the rogue DHCP server, DHCP starvation, and malicious DHCP client attacks [1]. The main source of DHCP vulnerabilities is that it cannot authenticate entities (the DHCP server and clients) or exchanged messages. Without authentication, a rogue DHCP server can send wrong information to hosts, which allows it to perform more complex attacks, such as MITM, sniffing, and phishing. Without using a mechanism for client authentication, a malicious user can access the network and use its services. Also, he or she can perform any other attack, such as the DoS attack, which may prevent legitimate users to access the network or use its services [1]. Section 6 discusses DHCP attacks in detail and explains how the proposed solution mitigates them.

2.2 *Address resolution protocol*

ARP is one of the protocols that are essential for the operation of LANs. It is used to resolve the Medium Access Control (MAC) address of a device connected to the network whose IP address is known. ARP receives the IP datagram from the network layer, and then it maps the IP address to the corresponding MAC address. After that, ARP passes the IP datagram to the data link layer. The resulting

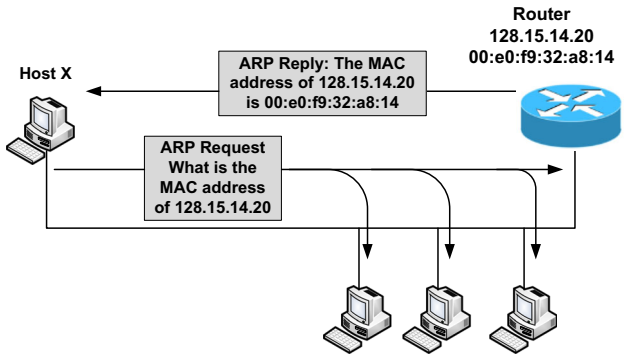


Figure 2. ARP operations.

address mapping is used to deliver the packet within the LAN. If the IP address is for a device located outside the LAN, the packet is directed to the gateway.

Anytime a device or host in a LAN has a packet to send to another device or host, it has to find the destination MAC address. Therefore, it broadcasts an ARP request that includes the known IP address of the destination, and the MAC and IP addresses of the sender, as shown in figure 2. Every device or host on the LAN receives the ARP request message and then it checks the destination IP address. If the intended receiver recognises its IP addresses, it sends back a unicast ARP reply to the source. The ARP reply contains the receiver MAC and IP addresses.

On receiving the ARP reply, the sender temporarily stores the IP/MAC pair in the ARP cache, where a list of IP/MAC pairs is stored. Any host must check the ARP cache before sending a discovery packet (ARP Request) to determine if there is an entry for the destination IP address. The ARP cache is updated periodically, where the old and unused entries are removed [35].

The ARP protocol was designed without any security features. It does not support any scheme for message authentication or integrity. In addition, ARP is a stateless protocol because it processes the ARP reply even if it did not send any ARP request. This makes ARP vulnerable to many attacks such as ARP spoofing attack. ARP spoofing is a technique by which an attacker transmits a spoofed ARP request or reply in a LAN. In this attack, the attacker associates its MAC address with the IP address of the victim. So, any traffic sent to the IP address of the victim will instead be sent to the attacker. The ARP spoofing attack is a serious security problem that threats LANs [6]. It is usually used as a start to perform other attacks, such as MITM attack [4], DoS attack [5], host impersonation attack, and connection hijacking attack [6].

3. Related work

Several studies were introduced in the literature to address the security issues in DHCP. In RFC 3118 [23], two standard techniques are 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 a DHCP server that has inadvertently been instantiated. It does not support DHCP message authentication. The delayed authentication scheme uses the HMAC (keyed-hash message authentication code) technique for DHCP message authentication. It uses a pre-shared secret key with MD5 message-digest algorithm to generate the message authentication code for DHCP messages. However, the security of the MD5 hash function used for message authentication is compromised [24].

In [25], the method introduced in [23] was developed for securing DHCP using a digital certificate. The authors used a trusted server to distribute the digital certificates between the server and clients. Because of its size, the digital certificate cannot fit into one DHCP message. Therefore, the authors introduced a scheme that depends on the fragmentation of the DHCP message into many messages.

Based on the authentication option specification introduced in [23], a method was introduced in [26] based on using X.509 digital certificate. The digital certificate was used to sign the DHCP messages transmitted between clients and the DHCP server. A common trust server or authority was used to distribute the digital certificate. The authors showed that the digital certificate could not be loaded in one DHCP message because of its large size. In addition, they indicated that the digital certificate revocation policies are hard to set up. Moreover, the authors recommended using certificate-based authentication with delay authentication.

A scheme called Secure DHCP with Digital Certificates (SDDC) was proposed in [27] make DHCP more secure. The SDDC method depends on using digital certificates to authenticate the DHCP messages and entities. However, this method did not consider that the size of the digital certificate might exceed the size of the DHCP message.

Another mechanism called Challenge Handshake Authentication Protocol (CHAP) was introduced in [28] for authenticating entities that use DHCP. This mechanism lets the server generate an encrypted challenge-response and attaches it to the DHCP OFFER message when it receives the DHCP DISCOVER message. The challenge-response is computed using a hash function and a secret key that must be shared between the DHCP server and clients. In addition, other DHCP messages, such as DHCP REQUEST and DHCP ACK, are attached with the challenge-response to authenticate the messages and entities.

In [29], a method for authentication of clients and DHCP message was proposed based on Kerberos V [30]. The proposed method used the authentication option specification defined in [23]. Kerberos V protocol uses an authentication server and key distribution centre, which generated tickets for clients and servers. Tickets are encrypted using secret keys shared between the server and clients. To authenticate the DHCP REQUEST and response, the ticket

of the client or server is included in the DHCP message. The main disadvantage of this method is that its computational cost is very high.

In [31], Dinu and Togan proposed a technique to protect the DHCP server. This technique authenticates the DHCP messages and the DHCP server using digital certificates and public key cryptography, which can help to prevent the rogue DHCP server attack. However, this technique is vulnerable to DHCP starvation attack, which can be developed to establish a rogue DHCP server. This is explained in section 6 in detail.

In [32], a scheme called DHCPAuth was introduced to authenticate DHCP messages according to two trust models: PGP (Pretty Good Privacy) and PKI (Public Key Infrastructure). PKI is a framework that uses a certificate authority for generating, checking, and revocation of public key certificates, which prove the identity of entities and the ownership of a private key. PGP is a software used for encryption, decryption, signing and verification of signatures. These trust models have some drawbacks as pointed in [36, 37], which make them totally unsecure.

As explained earlier, most schemes introduced in the literature use an authenticator to authenticate the DHCP messages and entities. The authenticator is attached to the DHCP messages, which takes different forms such as a token, ticket, certificate, and message digest. Unfortunately, using these types of authenticator does not prevent a DHCP starvation attack, which can be developed to the rogue DHCP server attack, as explained in section 6.

Several schemes were proposed to target security issues in ARP. These techniques can be classified into two categories: non-cryptographic [7–18] and cryptographic [19–22]. Non-cryptographic techniques are software solutions used to analyse the traffic in the network to detect ARP spoofing. Non-cryptographic techniques have many drawbacks, which are summarised as follows:

- These techniques inspect each packet at the link layer, which has a negative effect on the delay and throughput.
- They do not prevent the attack, but it waits until the attack occurs and try to resolve it.
- They cannot be used with DHCP addressing, where the static ARP cache entries should be manually configured.
- They cannot be applied in large-scale networks.
- They are not cost-effective.
- Some of them are not feasible.

Compared with non-cryptographic techniques, cryptographic techniques are more efficient and effective in mitigating ARP attacks. Many cryptographic techniques have been proposed in the literature. To the best of our knowledge, the most significant techniques were introduced in [19–22], which are discussed later.

In [19], a technique called Secure Address Resolution Protocol was proposed. In this technique, a secure server is used to distribute secret keys to all clients in the network. Then, the server periodically sends an invite message to each client. Every client receives an invite message, it should send the server an accept message, including its IP/MAC address pair. The server stores IP/MAC address pair in a local database. When a client tries to resolve an IP address into its corresponding MAC address, it sends a request to the server, which sends a reply message with the resolved MAC address. However, this technique cannot prevent man-in-the-middle attack and replay attack. In addition, the overhead incurred by the invite message sent from the server to clients is high and may cause network congestion.

An ARP security scheme called S-ARP was proposed in [20], which is an extension to the ARP protocol. S-ARP depends on public key cryptography for providing authentication for ARP reply messages. To obtain the public key, the host must contact an authentication authority called Authoritative Key Distributor (AKD) server. During the bootstrapping process, the MAC address and the public key of the server are securely distributed to all clients. The request message of S-ARP is similar to that of ARP. However, the reply of S-ARP is authenticated by appending the signature of the sender to the message obtained from AKD. Using the public key of the sender, the client that receives the reply message verifies the signature of the sender. However, the main drawback of this technique is that it does not support dynamic assignment of IP addresses. Also, this technique needs to generate and verify at least a signature for each address resolution process, which deteriorates the performance of the protocol.

A technique was presented in [21] called Secure Link Layer (SSL). This technique encrypts traffic in data link layer using the public key cryptography. SSL can authenticate entities and protect message integrity. However, it does not prevent authorised hosts from transmitting malicious messages, which can be used to perform many attacks.

Based on S-ARP, Lootah *et al* proposed a technique to secure ARP called the Ticket-based Address Resolution Protocol (TARP) [22]. Using a security server, secure attestations including MAC/IP address mapping, called tickets, are distributed to hosts when they are connected to the network. The distributed tickets are signed by a Local Ticket Agent (LTA) and are used to authenticate the MAC/IP addresses binding. Each ticket is valid for a specified period. For address resolution, a client broadcasts the ARP request. Then, the required client sends an ARP reply attached with its ticket. On receiving the ARP reply, the requesting host validates the attached ticket using the public key of the LTA.

The performance of TARP is better than that of S-ARP because TARP needs only one public key decryption operation per process for address resolution. However,

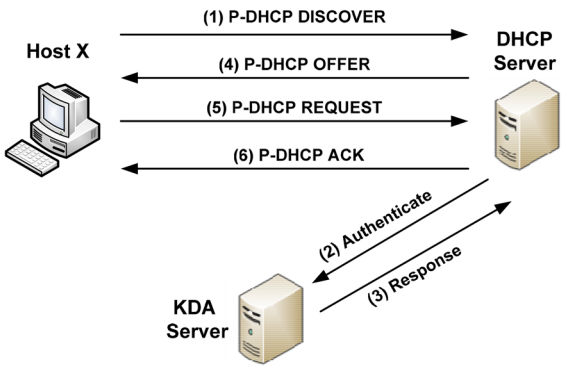
TARP is vulnerable to host impersonation attack. As the authors explained, an attacker can impersonate an offline host by replying to its ticket. In addition, using a ticket with a short lifetime greatly increases the traffic load on the network because clients need to renew their tickets more frequently, and thus increases the computational cost for ticket generation. On the other hand, increasing the ticket lifetime increases the possibilities for a replay attack.

4. Protect DHCP

As explained earlier, DHCP is vulnerable to many attacks. The main weaknesses of DHCP lie in the fact that it does not authenticate DHCP messages and entities, and does not consider integrity of the message. This work introduces a new scheme to secure DHCP, called P-DHCP (Protect DHCP), which overcomes these weaknesses. This section explains P-DHCP.

Figure 3 shows the operation of the P-DHCP protocol, which illustrates the messages exchanged between a host X, the DHCP server, and the KDA (Key Distribution and Authentication) server. The KDA server is connected to a network to distribute security keys and to authenticate the connecting entities (a client or server).

When a host X is installed on the network, the KDA server generates a session key K_X for it. Then, the server shares the session key with X. This operation is performed manually or by using a secure channel. For this work, manual distribution of session keys is only considered. In addition, to register the host to the KDA server, the server stores the binding between the session key, ID, and MAC address of the host in a local database. To authenticate a host, we suppose that the KDA server uses an effective security protocol such as Kerberos [30], Challenge-Handshake Authentication Protocol (CHAP) [38], or Extensible Authentication Protocol (EAP) [39].



$$\textbf{DISCOVER} = \textbf{DHCP DISCOVER} \,||\, ID_X \,||\, MAC_X \,||\, \textbf{TS} \,||\, \textbf{LT}$$

$$\textbf{P-DHCP DISCOVER} = \textbf{DISCOVER} \,||\, \textbf{H}(\textbf{DISCOVER})$$

$$\textbf{Authenticate} = \textbf{E}(Ks, [\textbf{DHCP OFFER} \,||\, ID_S \,||\, MAC_S \,||\, ID_X \,||\, MAC_X \,||\, IP_X \,||\, \textbf{TS} \,||\, \textbf{LT}])$$

$$\textbf{OFFER} = \textbf{DHCP OFFER} \,||\, MAC_X \,||\, IP_X \,||\, ID_X$$

$$\textbf{P-DHCP OFFER} = \textbf{E}(K_X, [\textbf{OFFER} \,||\, K_{XS} \,||\, \textbf{TS} \,||\, \textbf{LT}])$$

$$\textbf{Response} = \textbf{E}(K_S, [\textbf{P-DHCP OFFER} \,||\, K_{XS} \,||\, \textbf{TS} \,||\, \textbf{LS}])$$

$$\textbf{P-DHCP REQUEST} = \textbf{E}(K_{XS}, [\textbf{DHCP REQUEST} \,||\, \textbf{TS} \,||\, \textbf{LS}])$$

$$\textbf{P-DHCP ACK} = \textbf{E}(K_{XS}, [\textbf{DHCP ACK} \,||\, \textbf{TS} \,||\, \textbf{LS}])$$

Figure 3. P-DHCP message exchange.

After authenticating the registered host, it starts to contact the DHCP server to get network settings. Host X broadcasts the P-DHCP DISCOVER message, which consists of the ordinary DHCP DISCOVER message appended with the ID and MAC address of X, the timestamp (*TS*), and the message lifetime (*LT*). To authenticate this message, it is also appended with the message digest produced using a one-way hashing function. The TS and LT fields are used to inform the receiver about the time at which this message was generated and the period of time at which the message will be valid. TS and LT are used to prevent the replay attack.

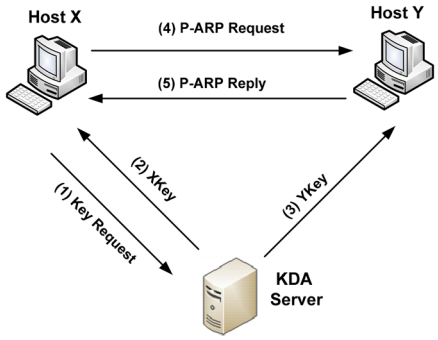
After receiving the P-DHCP DISCOVER message, the DHCP server checks the validity of the message. It generates the message authentication code and compares it with the hash code appended to the message. If the hash code is valid, the server continues to process the message. Otherwise, it ignores the message. After validating the P-DHCP DISCOVER message, the DHCP server checks that host X is registered with the KDA server. To do that, it generates the Authenticate message and sends it to the KDA server. The Authenticate message consists of the standard DHCP OFFER message appended with the ID and MAC address of the DHCP server; the ID, IP address, and MAC address of host X; TS; and LT. To protect the Authenticate message, it is encrypted using the DHCP server session key (K_S).

After receiving the Authenticate message, the KDA server decrypts it using the session key (K_S) shared with the DHCP server. Then, using the ID and MAC address of host X, it searches its database to check if host X is registered or not. If host X is a valid host, the KDA server generates the Response message and sends it to the KDA server. The Response message consists of the P-DHCP OFFER message, received from KDA server, appended with K_{XS}, TS, and LS. K_{XS} is the shared session key generated by the KDA server to be used to encode any further message exchanged between host X and the KDA server. The Response message is encrypted using K_S. On the other hand, the P-DHCP OFFER message is encrypted using K_X. As shown in figure 3, P-DHCP OFFER message includes the standard DHCP OFFER message received from the KDA server, ID_X, MAC_X, IP_X, K_{XS}, TS, and LS.

If host X is not a legitimate host (or is not registered with the KDA server), the KDA server does not send the Response message to the DHCP server. After a timeout period, the DHCP server sends another Authenticate message to the KDA server. This process is repeated for three

times. If the DHCP does not receive any response from the KDA server, the DHCP server ignores the P-DHCP DISCOVER messages sent by X.

When the DHCP server receives the Response message, it decrypts the message using the session key K_S and extracts the P-DHCP OFFER message from it, which is transmitted to host X. Host X decodes the message using its session key to get DHCP OFFER and the shared key K_{XS}. If host X accepted the offer sent by the DHCP server, it constructs the P-DHCP Request message and sends it to the DHCP server. The P-DHCP REQUEST message is encoded by the shared key K_{XS}, where it includes the standard DHCP REQUEST, TS, and LS. On receiving and decoding the P-DHCP REQUEST message, the DHCP server constructs and transmits the P-DHCP ACK message to host X. As shown in figure 3, the P-DHCP ACK message is the standard DHCP ACK message encoded by K_{XS}. Host can decode the P-DHCP ACK message, and then can read the DHCP ACK message, which acknowledges the leased offer to the host.



$$\textbf{Key Request} = \mathrm{E}(K_X, [\mathrm{ID}_X \,||\, IP_X \,||\, MAC_X \,||\, IP_Y \,||\, TS \,||\, LT])$$
$$\textbf{XKey} = \mathrm{E}(K_X, [ID_X \,||\, IP_X \,||\, MAC_X \,||\, ID_Y \,||\, IP_Y \,||\, K_{XY} \,||\, TS \,||\, LT])$$
$$\textbf{YKey} = \mathrm{E}(K_Y, [ID_Y \,||\, IP_Y \,||\, MAC_Y \,||\, ID_X \,||\, IP_X \,||\, K_{XY} \,||\, TS \,||\, LT])$$
$$\textbf{P-ARP Request} = \mathrm{E}(K_{XY}, [\textbf{ARP Request} \,||\, ID_X \,||\, IP_X \,||\, MAC_X \,||\, TS \,||\, LT])$$
$$\textbf{P-ARP Reply} = \mathrm{E}(K_{XY}, [\textbf{ARP Reply} \,||\, ID_Y \,||\, IP_Y \,||\, MAC_Y \,||\, TS \,||\, LT])$$

Figure 4. P-ARP message exchange.

5. Protect ARP

The main weakness of ARP is that its messages lack integrity and authenticity. In addition, as ARP is a stateless protocol, it does not keep any information regarding ARP requests sent out and does not differentiate between received messages. It blindly trusts any received ARP reply, even if it has a spoofed address. This section describes a new technique called P-ARP (Protect ARP) to secure address resolution protocol. The proposed technique provides integrity and authenticity for ARP messages to guarantee that the request and reply message has not been changed and that they were sent by the authentic user. In addition, it converts ARP from stateless to stateful protocol.

P-ARP consists of two phases: setup and protection. In the setup phase, the P-ARP client contacts the KDA server to get the session key that is used to encode messages transmitted to other clients in the network. After getting the session key, the protection phase starts. In this phase, the P-ARP messages are encoded with the session key and transmitted between clients. The following explains these two phases in detail.

5.1 *Setup phase*

Figure 4 illustrates the two phases of the P-ARP protocol. As explained in section 4, after registration with the KDA server, an authentic host, such as X, can get its IP address from the DHCP server using P-DHCP. If host X tries to contact another host Y, but it only knows the IP address of Y, it performs MAC address resolution using P-ARP protocol, as shown in figure 4. Host X sends the Key Request message to the KDA server. Host X encodes the Key Request message using its session key K_X. The message includes the following fields: the ID; IP; and MAC address of host X, the IP address of host Y, TS, and LT.

On receiving the Key Request message, the KDA server can interpret the message and know that host X and Y are trying to communicate and asking for a shared key. Therefore, the KDA server generates the session key K_{XY}. As shown in figure 4, the key K_{XY} is encoded into XKey message and YKey message, which are transmitted to host X and Y, respectively. Host X and Y can extract the shared session key K_{XY} from the XKey message and YKey message, respectively. After getting the shared session key, the setup phase ends up.

5.2 *Protection phase*

After obtaining the shared session key, host X broadcasts the P-ARP Request message, as shown figure 4. The P-ARP Request message consists of the standard ARP Request, TS, LT, and ID; the IP address; and the MAC address of host X. To allow only host Y to be able to access the P-ARP Request message, it is encrypted using the shared key between X and Y. Any other host than Y, running classic ARP or P-ARP client, cannot access the Request message.

On receiving the P-ARP Request message from host X, host Y decodes and validates the message. If the message is valid, it extracts the ARP Request. If the destination IP address in the ARP Request is for host Y, host Y adds the MAC and IP mapping of host X to its ARP cache and starts to prepare the P-ARP Reply message. The P-ARP Reply message includes the classic ARP Reply, TS, LT, and the IP address; the MAC address; and ID of host Y. Host Y encodes the P-ARP Request message using the key shared

with host X (K_{XY}). When host X receives the P-ARP reply message, it decodes and validates it. If the message is valid and the destination IP address is for host X, it adds the MAC and IP mapping of host Y to its ARP cache.

After sending any message by a client or a server, a timeout timer is set. During only this period of time, the client is allowed to receive the response to the message. Otherwise, any response received for this request is discarded. This feature enabled by the P-ARP client converts ARP from a stateless to a stateful protocol.

6. Analysis of security threats

This section discusses how the proposed schemes mitigate different vulnerabilities in data link layer in local area networks. In addition, security properties of the proposed schemes are compared with other related schemes.

6.1 *TARP security vulnerabilities*

As explained in section 3, TARP uses a ticket to authenticate the MAC/IP address binding. A trusted server called the Local Ticket Agent (LTA) manages the tickets. When a host sends a DHCP request to the DHCP server to obtain the IP address and other configuration information, the LTA server sends a ticket to this host. The ticket is used to protect the ARP Reply message. Any host receiving a reply message validates the ticket using the public key of the LTA server. If the ticket is valid, the MAC/IP association is accepted and used to update ARP cache; otherwise, the reply message is ignored. Compared to other schemes introduced in the literature, TARP is more effective and has better performance. However, TARP is vulnerable to host impersonation attack and DoS attack.

The host impersonation attack can be performed for TARP in two steps: (1) launching the MAC spoofing attack, (2) capturing the victim's ticket. To launch the MAC spoofing attack, the attacker does the following as shown in figure 5:

1. To launch this attack, the victims machine should be disconnected from the network. The attacker may attack the victim's machine using the DoS attack to make it offline, or force the victim to restart the machine.
2. The attacker sends a message to the network switch with a forged MAC address of the victim (MAC_B).
3. When the network switch receives the message, it looks in the MAC address table (called CAM table) for the source and destination MAC address. The message is forwarded to the port associated with the destination MAC address. In addition, it updates the port number associated with the source MAC address. The switch assigns the MAC address of the victim to the port connected to the attacker. Therefore, it forwards frames, destined to the victim, to the attacker.

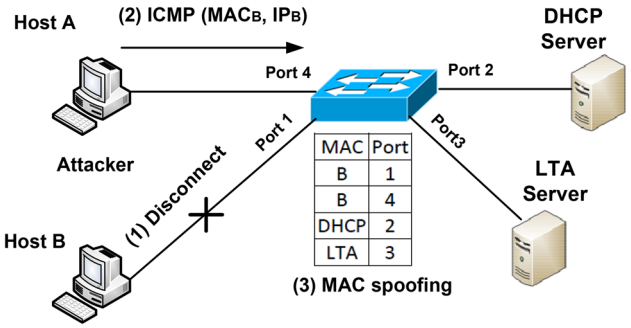


Figure 5. MAC spoofing attack.

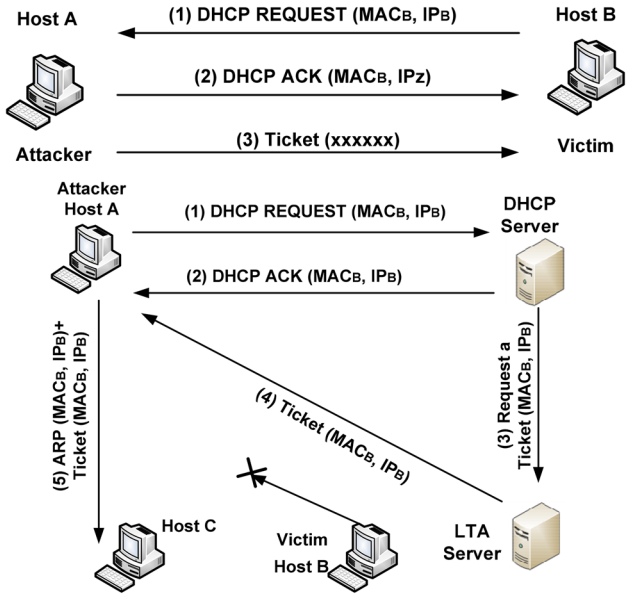


Figure 6. Host impersonation attack for TARP.

Next, the attacker captures the victim's ticket as follows, as shown in figure 6:

1. Send a DHCP request message with forged MAC and IP addresses (MAC_B and IP_B) to the DHCP server.
2. When the DHCP server receives the DHCP request, it renews the lease for host B. It sends a reply message to Host B (victim).
3. The DHCP server sends a request to the LTA server to issue a new ticket for host B (Ticket [MAC_B, IP_B]).
4. The LTA server sends the ticket to host B.
5. Due to spoofing of the switch MAC table, the attacker receives the ticket.
6. To renew the ARP table of other hosts, the attacker sends them an ARP Reply message with forged IP and MAC addresses and the ticket of the victim.

At the end, the attacker can impersonate the victim by using his or her ticket. He or she can then receive all traffic

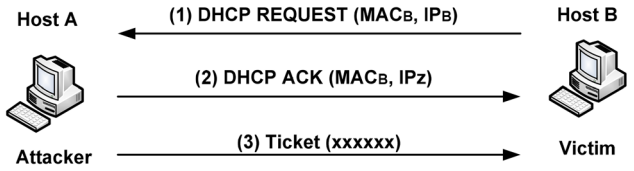


Figure 7. DoS attack for TARP.

intended for the victim. If the victim's machine is reconnected to the network, the attacker may redirect traffic to him after modification or eavesdropping, which is known as MITM attack. This attack can be very harmful if the attacker succeeds in impersonating the network gateway or the DHCP server.

To perform the DoS attack for TARP, the attacker does the following:

1. The attacker launches a MAC spoofing attack (as explained earlier) to associate the MAC address of the DHCP server with the port that his or her machine uses.
2. As shown in figure 7, when the victim (Host B) sends a DHCP request (DHCP Request (MAC_B, IP_B)) to the DHCP server to renew the leased IP address, the request will be forwarded to the attacker.
3. After receiving the DHCP request, the attacker sends the victim a DHCP ACK message offering a forged IP address (IP_Z).
4. The attacker generates a forged ticket and sends it to the victim.

Finally, due to the forged IP address, the traffic destined to the victim will not reach. In addition, other hosts will not trust the victim if he tries to update their ARP cache because of its forged ticket. Consequently, all network services of the victim will be disrupted.

Compared with TARP, the proposed scheme P-ARP mitigates the host impersonation attack and DoS attack. If the attacker performed the attacks as explained, he can get the P-DHCP messages or P-ARP messages. However, he cannot read these messages or replay them because they are encoded with a session key. The only way to impersonate a host (or send forged frames) is to get its security key by capturing it or by using brute force attack. This will be very difficult if we used an efficient authentication technique and chose a security key with a large size.

6.2 *DHCP authentication vulnerabilities*

As explained in section 3, to secure DHCP, most techniques proposed in the literature use an authenticator to authenticate entities and DHCP messages, as illustrated in figure 8. Common forms of the authenticator are the digital certificate, ticket, token or message-digest. Most techniques use the public key cryptography to protect the authenticator. However, most of these techniques cannot mitigate the DHCP starvation attack. This attack was designed to

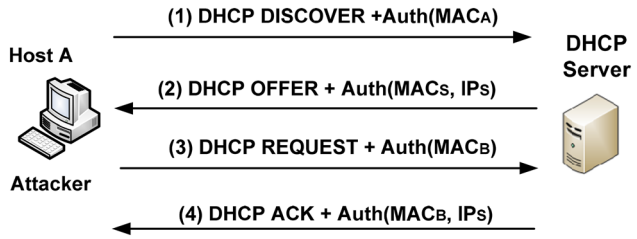


Figure 8. Protecting entities and DHCP messages using an authenticator.

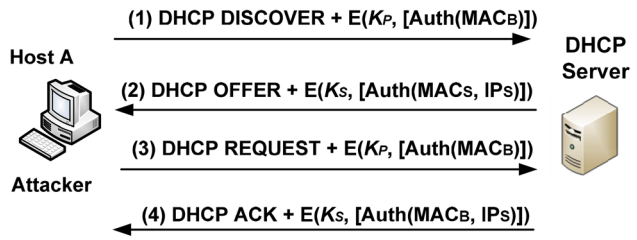


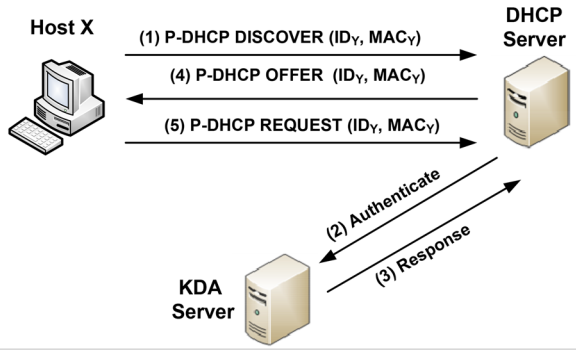
Figure 9. DHCP attack against the DHCP authentication technique.

deplete all of the addresses within the address space allocated by the DHCP server. It can be performed by flooding the DHCP server with DHCP requests with spoofed MAC addresses. As a result, the address space available for the DHCP server is exhausted. Therefore, the DHCP server denies any IP address request, which prevents clients to connect to the network. Hence, the DHCP starvation attack can be classified as a denial-of-service attack.

Figure 9 illustrates the DHCP starvation attack against the DHCP authentication techniques. This attack can be performed as follows:

1. The attacker (host A) broadcasts a DHCP DISCOVER message with a spoofed MAC address (MAC_B). To authenticate the message, it is attached with the authenticator that is encrypted using the public key of the DHCP server (K_P).
2. The DHCP server replies with a DHCP OFFER attached with the authenticator signed by the server private key (K_S). The authenticator includes the MAC address (MAC_S) and IP address (IP_S) of the DHCP server.
3. On receiving the DHCP OFFER message, the attacker encodes the authenticator using the public key of the DHCP server and attaches it to the DHCP REQUEST message that is sent to the server.
4. At the end, the server sends the DHCP ACK message that acknowledges to the client the lease finalisation.

By repeating this process for different forged MAC addresses, the attacker can exhaust all IP addresses available for the server. For techniques that use the message digest as an authenticator, the attacker can set up a rogue DHCP server on the network and set his machine as the default gateway and can sniff network traffic.



$$\textbf{DISCOVER} = \textbf{DHCP DISCOVER} \,||\, ID_Y \,||\, MAC_Y ||\, \textbf{TS} \,||\, \textbf{LT}$$

$$\textbf{P-DHCP DISCOVER} = \textbf{DISCOVER}||\, \textbf{H}(\textbf{DISCOVER})$$

$$\textbf{Authenticate} = \textbf{E}(Ks, [\textbf{DHCP OFFER} \,||\, ID_S \,||\, MAC_S \,||\, ID_Y \,||\, MAC_Y \,||\, IP_Y \,||\, \textbf{TS} \,||\, \textbf{LT}])$$

$$\textbf{OFFER} = \textbf{DHCP OFFER} \,||\, MAC_Y \,||\, IP_Y \,||\, ID_Y$$

$$\textbf{P-DHCP OFFER} = \textbf{E}(K_Y, [\textbf{OFFER} \,||\, K_{YS} \,||\, \textbf{TS} \,||\, \textbf{LT}])$$

$$\textbf{Response} = \textbf{E}(K_S, [\textbf{P-DHCP OFFER}||\, K_{YS} \,||\, \textbf{TS} \,||\, \textbf{LS}])$$

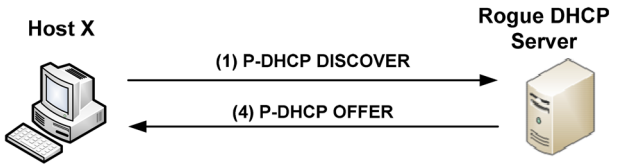
$$\textbf{P-DHCP REQUEST} = \textbf{E}(K_{XX}, [\textbf{DHCP REQUEST} \,||\, \textbf{TS} \,||\, \textbf{LS}])$$

Figure 10. DHCP starvation attack against P-DHCP.

The proposed scheme (P-DHCP) can effectively mitigate the DHCP starvation attack. As shown in figure 10, to perform this attack with P-DHCP, the attacker (host X) broadcasts a P-DHCP DISCOVER message with a spoofed ID (ID_Y) and MAC address (MAC_Y). After receiving the P-DHCP DISCOVER message, the DHCP server contacts the KDA server to check the authenticity of the host with ID_Y and MAC_Y. If host Y is not a legitimate host, the KDA and DHCP servers do not send any response to host X. Otherwise, the DHCP server and KDA server exchange Authenticate and Response messages, as shown in figure 10. Then, the DHCP server sends the attacker the P-DHCP OFFER, which is encrypted using the session key of host Y (K_Y). Upon receiving the DHCP OFFER message, the attacker tries to encrypt it, which is not possible because he does not have the session key K_Y. If the attacker continues the attack, he sends the DHCP server the P-DHCP REQUEST message encrypted using any key other than K_{YS}, such as *KXX*, as shown in figure 10. Consequently, the DHCP server cannot decrypt and read the P-DHCP REQUEST message because it decrypts the message using the session key K_{YS} provided by the KDA server. Therefore, the DHCP server does not assign an IP address for the attacker.

6.3 *Rogue DHCP 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



$$\textbf{DISCOVER} = \textbf{DHCP DISCOVER} \,||\, ID_X \,||\, MAC_X ||\, \textbf{TS} \,||\, \textbf{LT}$$

$$\textbf{P-DHCP DISCOVER} = \textbf{DISCOVER}||\, \textbf{H}(\textbf{DISCOVER})$$

$$\textbf{OFFER} = \textbf{DHCP OFFER} \,||\, MAC_X \,||\, IP_X \,||\, ID_X$$

$$\textbf{P-DHCP OFFER} = \textbf{E}(K_N, [\textbf{OFFER} \,||\, K_M \,||\, \textbf{TS} \,||\, \textbf{LT}])$$

Figure 11. Rogue DHCP server attack against P-DHCP.

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.

P-DHCP can resist the rogue DHCP server attack. Suppose that an attacker installed a rogue DHCP server on the network, as shown in figure 11. When a host X broadcasts a P-DHCP DISCOVER message, an attacker may send an offer for X. The attacker encrypts the DHCP OFFER message using an unknown key K_N, as shown in figure 11. On receiving the DHCP OFFER message, host X decrypts it using its session key K_X, which was securely shared with the KDA server. Therefore, X cannot recognise the offer message and does not continue the process. As a result, host X does not accept the DHCP OFFER sent by the rogue DHCP server. By using P-DHCP, to impersonate the DHCP server, the attacker must have the session key of the victim.

6.4 *ARP spoofing attack*

In the ARP spoofing attack, the attacker sends falsified ARP messages to a host on the LAN to associate his MAC address with the victim's IP address. This results in poisoning the ARP cache with a wrong binding between the attacker's MAC address and the victim's IP address. As a result, the attacker can receive all packets that are intended for the victim's IP address.

By using the P-ARP scheme, to perform an ARP spoofing attack, the attacker must spoof the Key Request message, P-ARP Request message, or P-ARP Reply message. This is possible only if the attacker knows the session key of the victim, which is used to encode the communication with the KDA server or any other client. As explained earlier, this is not an easy task, especially with using session keys with a large size.

6.5 *Man-in-the-middle attack*

MITM attack can be established when an attacker spoofs the ARP cache of two victims with his MAC address. Once the ARP cache of the victims has been poisoned, the victim

machines send all of the packets intended for the other machine to the attacker. The attacker can easily monitor all communication between the victim machines. To perform the MITM attack, the attacker must compromise the security key of the two victims to spoof their ARP cache. As explained earlier, this is a very difficult task, especially if the size of security keys is large.

6.6 *Denial-of-service attack*

Denial of service (DoS) attack can be performed using ARP spoofing. In this attack, the attacker updates the ARP cache of a host or device with forged MAC addresses. As a result, the outbound traffic of a host or device is directed to the attacker machine, and the inbound traffic is blocked. The P-ARP scheme can mitigate this type of attack because it prevents spoofing of the ARP cache, as explained earlier.

6.7 *P-ARP and P-DHCP vulnerabilities*

The proposed schemes are vulnerable to the flooding attack. This attack can be performed by flooding the victim's host or device with fake messages. This overloads the victim's host or device and consumes its resources that may stop its function. To launch this attack, the adversary may send a large number of fake messages in a short time, such as the P-ARP Request, P-ARP reply, Key Request, P-DHCP Discover, and P-DHCP REQUEST messages. The best countermeasure to this attack is to use a switch that supports port security. This type of switches can analyse the rate of transmitting packet in each port of the switch and can limit or block the traffic on a certain port if it increased over a threshold.

7. Performances analysis and comparison

To assess the performance of the proposed solution, it has been implemented using the same architecture introduced in [20] and [22]. This architecture was implemented on Linux. It consists of two parts: the kernel module and userspace daemon. The kernel module prevents the kernel from processing DHCP and ARP messages. It drops these messages to let the userspace daemon process them according to the designed protocol. Most functionality of the security protocols was implemented in the userspace daemon. It can act as a KDA server, a DHCP server, or a client. The libpcap and libnet sockets were used for packet capturing and packet creation and injection, respectively. The netlink socket was used to manage ARP cache and DHCP table entries, whereas the encryption and decryption operations were performed using OpenSSL.

For experimental evaluation of the proposed solution, we used a test bed consisting of a set of PCs connected through Gigabit Ethernet switch with the following specifications:

- All clients are equipped with 2.4 GHZ core i5 processor and 4 GB of RAM.
- The DHCP server and KDA server is equipped with 3.6 GHZ core i7 processor and 8 GB of RAM.
- All machines run Ubuntu Linux 14.04.4.

To analyse the performance of the proposed schemes, two performance metrics are proposed: the Configuration Time (CT) and the Resolution Time (RT). The CT is the time that a client takes to get the DHCP configuration parameters from the DHCP server. It is the time taken to process and exchange the DHCP messages; DHCP DISCOVER, DHCP OFFER, DHCP REQUEST, and DHCP ACK messages. The resolution time is the time needed to resolve the MAC address of a client. RT includes the time of processing and exchanging the ARP request and ARP reply.

For the first experiment, CT is measured for a single client connected with the DHCP and KDA servers. For symmetric encryption, the Advanced Encryption Standard (AES) is used with a key of 128 bits, 192 bits, and 256 bits. We ran the test 1000 times to measure CT. The minimum, maximum, mean, and standard deviation of CT are calculated and reported in table 1. As this table shows, the higher the encryption key size, the higher the time required to get DHCP configuration parameters from the DHCP server. In addition, with a large key size, CT for P-DHCP scheme is very small (about 1.6 ms).

In the second experiment, in order to show the real overhead of the P-DHCP, it is compared with DHCP, where the overhead in CT is computed by taking the difference between P-DHCP and DHCP. In addition, P-DHCP is compared with DHCPAuth scheme introduced in [32], which is one of the latest solution presented in the literature. DHCPAuth adopts using asymmetric encryption techniques such as RSA for signing digital certificates. In this experiment, AES with a key of 256 bits and RSA with a key of 1024 bits are used with P-DHCP and DHCPAuth, respectively. Similar to the first experiment, CT is measured for only one client connected to the DHCP and KDA servers. The test is repeated for 1000 times, where the minimum, maximum, mean, standard deviation, and overhead are computed for CT. The results of this experiment are shown in table 2.

As shown in table 2, the overhead of DHCPAuth is about 6 times that of P-DHCP. In addition, compared with DHCP, the overhead of P-DHCP is very small. To get the network

Table 1. CT (μs) with different key sizes.

	128	192	256
Min	1016	1299	1603
Max	1530	1489	1475
Mean	1209	1556	1690
Stan. Dev.	77	84	85

Table 2. CT (μs) for DHCP, P-DHCP, and DHCPAuth.

	DHCP	DHCPAuth	P-DHCP
Min	650	5907	1603
Max	902	7828	1475
Mean	769	6821	1690
Stan. Dev.	36	366	85
Overhead	N/A	6052	921

settings from the DHCP server using P-DHCP and DHCPAuth, the number of exchanged messages is 6 and 4, respectively. In addition, the sizes of messages used by DHCPAuth and P-DHCP are nearly equal. Therefore, from the performance point of view, the main difference between P-DHCP and DHCPAuth is the adopted cryptographic technique. DHCPAuth adapts using the asymmetric cryptography, whereas P-DHCP adopts using the symmetric cryptography. The symmetric encryption is about 3 to 4 times faster than asymmetric encryption [40, 41]. In addition, the time needed for message decryption using the asymmetric cryptography is about 6 to 7 times larger than that needed for message encryption. Also, for the symmetric cryptography, the time required for encryption and decryption is nearly equal [40, 41]. As a result, the overhead of DHCPAuth is much greater than that of P-DHCP.

To evaluate how the overhead of DHCP, P-DHCP and DHCPAuth change with the number of clients, CT is measured for these techniques, where the number of clients varies between 2 and 16. Figure 12 shows the results of this experiment. As shown in figure 12, the overhead of DHCPAuth much increases with increasing the number of clients in the network that try to get DHCP configuration parameters at the same time. Increasing the number of clients increases the number of DHCP DISCOVER and DHCP REQUEST messages received and processed by the DHCP server. The messages are queued and processed one by one in arrival order, which induces queuing delay for DHCP requests at the DHCP server. The queuing delay significantly increases with increasing the processing time of each request. As explained earlier, for a single client, the overhead of DHCPAuth is larger than that of P-DHCP. Therefore, the message queuing delay induced by DHCPAuth greatly increases with increasing the number of clients compared with P-DHCP. As a result, compared with P-DHCP, the overhead induced by DHCPAuth greatly increases with increasing the number of clients.

To evaluate the performance of P-ARP, a system consisting of two clients and the KDA server is used. It is supposed that one of the clients tries to resolve the MAC address of the other one. The address resolution time (RT) is measured for different encryption key size. The experiment is repeated for 1000 times, where the ARP cache of clients is flushed after each experiment. The result of this experiment is shown in table 3. As shown in the table, increasing the key size increases the address RT because of

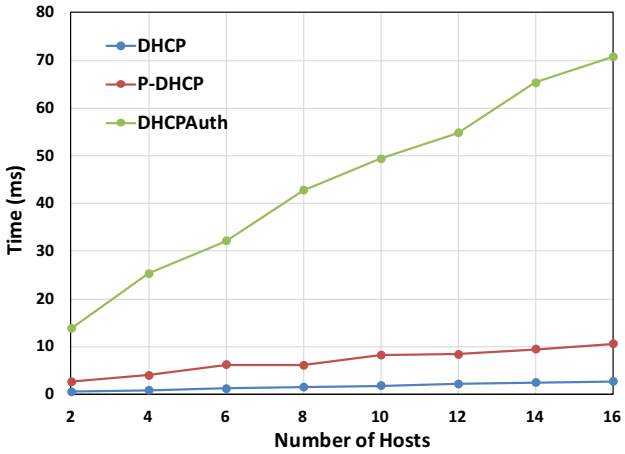


Figure 12. CT versus the number of clients.

Table 3. RT (μs) with key of 128 bits, 192 bits, and 256 bits.

	128	192	256
Min	1038	1352	1579
Max	1690	1977	2101
Mean	1245	1688	1869
Stan. Dev.	65	90	93

Table 4. RT (μs) for ARP, P-ARP, and TARP.

	ARP	TARP	P-ARP
Min	206	3834	1579
Max	383	4715	2101
Mean	312	4256	1869
Stan. Dev.	26	155	93
Overhead	N/A	3944	1267

increasing the time required for processing P-ARP messages. In addition, for large key size, the cost of address resolution is very small.

Next, the overhead of P-ARP is compared with that of TARP. The last experiment was repeated to measure the address RT for ARP, P-ARP, and TARP. Table 4 summarises the minimum, maximum, mean, and standard deviation of RT. Also, table 4 shows the overhead of P-ARP and TARP, where it was computed from the mean value. As shown, the overhead of TARP is about three times greater than that of P-ARP. For resolving a MAC address using TARP and P-ARP, the number of exchanged messages is 4 and 5, respectively. In addition, the size of messages used by TARP is nearly equal to that used by P-ARP. Therefore, from the performance point of view, the main difference between P-ARP and TARP is the adopted cryptographic technique. TARP adapts using the asymmetric cryptography for encoding and decoding the ticket attached to ARP messages, which is much slower than the

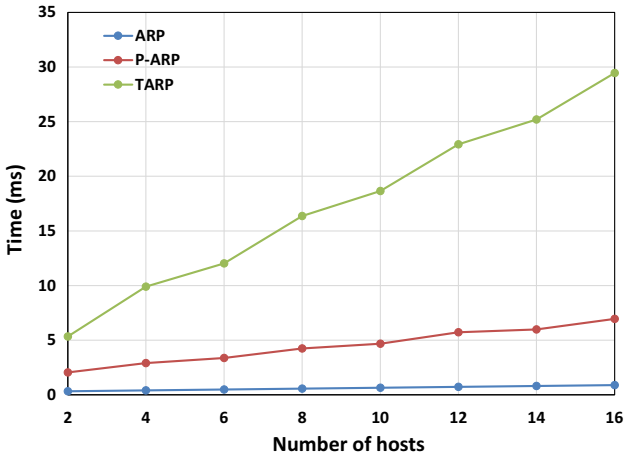


Figure 13. RT versus the number of clients for ARP, P-ARP, and TARP.

symmetric cryptography adopted by P-ARP, as explained earlier. Therefore, the overhead of TARP is much larger than that of P-ARP.

To investigate how the number of hosts in the network affects the overhead of ARP, P-ARP and TARP, the last experiment was repeated for a different number of hosts. Each host in the network tries to resolve the MAC address of another host at the same time. By varying the number of hosts from 2 to 16, RT was measured for ARP, P-ARP, and TARP. Figure 13 shows the results of this experiment. As shown in figure 13, compared to P-ARP, the overhead of TARP increases rapidly with increasing the number of hosts. TARP uses the LTA server to distribute tickets of clients. The higher the number of clients, the higher the number of tickets queued on the LTA server to be processed and distributed by the LTA server. Because the overhead of TARP for a single host is much larger than that of the P-ARP, as explained earlier, the message queuing delay induced by TARP exponentially increases with increasing the number hosts. As a result, increasing the number of hosts significantly increases the overhead of TARP compared to P-ARP.

8. Conclusion and future work

ARP and DHCP are crucial protocols for the operation of LANs. However, these protocols were designed without considering any security requirements. Therefore, using ARP or DHCP results in many serious security vulnerabilities that can be easily exploited to attack the network, such as the DHCP starvation, rogue DHCP server, ARP spoofing, MITM, and DoS attacks. Although many techniques were proposed to secure ARP, they are vulnerable to host impersonation, MITM, and DoS attacks. This is because these techniques did not consider security issues of DHCP. Most techniques proposed in the literature to protect DHCP messages and entities cannot mitigate the DHCP starvation attack.

This work introduced a new solution for securing ARP and DHCP. It consists of two schemes called P-DHCP and P-ARP. The two schemes authenticate entities, and ARP and DHCP messages using symmetric key cryptography. In addition, they protect the integrity of exchanged messages. To analyse the performance of P-ARP and P-DHCP, they have been implemented in Linux and real measurements have been taken for different metrics. Compared to related schemes introduced in the literature, the proposed schemes are more efficient in terms of security and performance. The proposed schemes can mitigate the DHCP starvation, MITM, and DoS attacks, whereas other related techniques are vulnerable to these attacks. Compared with other related techniques, the overhead of P-DHCP and P-ARP is better, especially with a large number of hosts. The proposed schemes do not prevent DoS attack based on flooding. In the future, we will add a module to the proposed solution for mitigating this attack.

References

[1] Altunbasak H C 2006 *Layer 2 security inter-layering in networks*. Thesis dissertation, Georgia Institute of Technology

[2] Droms R 1997 Dynamic host configuration protocol, RFC 2131

[3] Plummer D C 1982 An Ethernet address resolution protocol or converting network protocol addresses to 48 bit Ethernet address for transmission on Ethernet hardware, RFC 826

[4] Singh J, Kaur G and Malhotra J A 2015 Comprehensive survey of current trends and challenges to mitigate ARP attacks. In: *International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, Visakhapatnam

[5] Yu Yao and Yao Y 2010 A switch-based ARP attack containment strategy. In: *Second International Conference on Communication Systems, Networks and Applications* (ICCSNA)

[6] Dessouky M M, Elkilany W and Alfishawy N 2010 A hardware approach for detecting the ARP attack. In: *7th International Conference on Informatics and Systems* (INFOS)

[7] L. N. R. Group, arpwatch, the Ethernet monitor program; for keeping track of ethernet/ip address pairings, Last accessed September 17, 2016

[8] ARP-Guard, http://www.arp-guard.com, Accessed October 2016

[9] Puangpropitag S and Masusai N 2009 An efficient and feasible solution to ARP Spoof problem. In: *6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, vol. 02, pp. 910–913

[10] Bhirud D S G and Katkar V 2011 Light weight approach for IP-ARP spoofing. In: *The Second Asian Himalayas International Conference on Internet (AH-ICI),* pp. 1–5

[11] Hou X, Jiang Z and Tian X 2010 The detection and prevention for ARP spoofing based on Snort. In: *The International Conference on Computer Application and System Modeling* (*ICCASM*), pp. 137–139

[12] Ortega A P, Marcos X E, Chiang L D and Abad C L 2009 Preventing ARP cache poisoning attacks: A proof of concept using OpenWrt. In: *Latin American Network Operations and Management Symposium (LANOMS),* pp. 1–9

[13] Qian A Z 2000 The automatic prevention and control research of ARP deception and implementation. In: *World Congress on Computer Science and Information Engineering*, pp. 555–558

[14] Boughrara A and Mammar S 2012 Implementation of a SNORT's Output Plug-In in reaction to ARP Spoofing's attack. In: *6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pp. 643–647

[15] Md. Ataullah and N Chauhan 2012 ES-ARP: an efficient and secure address resolution protocol. In: *Conference on Electrical, Electronics and Computer Science (SCEECS)*, Bhopal, pp. 1–5

[16] Cisco Systems, Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide, 12.2(25) EW. http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/port_sec.html. Accessed October 2016

[17] Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Configuring DHCP Snooping. http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/snoodhcp.pdf. Accessed September 2016

[18] Catalyst 6500 Release 12.2SX Software Configuration Guide, Dynamic ARP Inspection, http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/dynarp.html. Accessed September 2016

[19] Gouda M and Huang C 2003 A secure address resolution protocol. *Comput. Netw.* 41: 860–921

[20] Bruschi D, Ornaghi A and Rosti E 2003 S-ARP: a secure address resolution protocol. In: *Proceedings of 19th Annual Computer Security Applications Conference*, pp. 66–74

[21] Jerschow Y I, Lochert C, Scheuermann B and Mauve M 2008 CLL: a cryptographic link layer for local area networks, security and cryptography for networks. In: *Lecture Notes in Computer Science*, vol. 5229, pp. 21–38

[22] Lootah W, Enck W and McDaniel P 2007 TARP: ticket-based address resolution protocol. *Comput. Netw.* 51: 4322–4337

[23] Droms R and Arbaugh W 2001 Authentication for DHCP messages, RFC 3118

[24] Stevens M M J 2007 *On collisions for MD5*. Master Thesis, Eindhoven University of Technology

[25] Xu Y, Manning S and Wong M 2011 An authentication method based on certificate for DHCP. *DHC Internet Draft*

[26] Glazer G, Hussey C and Shea R 2003 Certificate-based authentication for DHCP. http://www.cs.ucla.edu/∼chussey/proj/dhcp_cert/cbda.pdf. Accessed 20 Oct 2016

[27] Duangphasuk S, Kungpisdan S and Hankla S 2011 Design and implementation of improved security protocols for DHCP using digital certificates. 2011 In: *ICON*, Singapore

[28] 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 Publication

[29] K Hornstein, T Lemon, B Adoba and J Trostle 2001 DHCP Authentication Via Kerberos V. In: *IETF DHC Working Group*

[30] Ricciardi F 2007 Kerberos Protocol Tutorial. *National Institute of Nuclear Physics Computing and Network Services, LECCE*, Italy

[31] Dinu D D and Togan M 2014 DHCP server authentication using digital certificates. In: *The 10th International Conference on COMMUNICATIONS* (COMM2014), Bucharest, May

[32] Dinu D D and Togan M 2015 DHCPAuth—a DHCP message authentication module. In: *2015 IEEE 10th Jubilee International Symposium on Applied Computational Intelligence and Informatics*, Timisoara, pp. 405–410

[33] Kent S and Seo K 2005 Security architecture for the internet protocol. RFC 4301

[34] Dierks T and Rescorla E 2006 The Transport Layer Security (TLS) Protocol Version RFC 4346

[35] Song D 2016 dsniff: a collection of tools for network auditing and penetration testing. http://www.monkey.org/dugsong/dsniff. Accessed November 2016

[36] Ellison C and Schneier B 2000 Top 10 PKI risks. *Comput. Secur. J.* 16(1): 1–7

[37] Jonczy J, Wuthrich M and Haenni R 2006 A probabilistic trust model for GnuPG. In: *23rd Chaos Communication Congress*, Berlin

[38] Simpson W 1996 PPP Challenge Handshake Authentication Protocol (CHAP), RFC 1994

[39] Aboba B, Blunk L, Vollbrecht J, Carlson J and Levkowetz H 2004 Extensible Authentication Protocol (EAP), RFC 3748

[40] Agrawal M and Mishra P 2012 A comparative survey on symmetric key encryption techniques. *Int. J. Comput. Sci. Eng.* 4: 877–882

[41] Mahajan P and Sachdeva A 2015 A study of encryption algorithms AES, DES, and RSA for security. *Glob. J. Comput. Sci. Technol.* 13(15): 12–21