DHCPAuth – A DHCP Message Authentication Module

Dumitru Daniel Dinu, Mihai Togan
Department of Computer Science and Military Information Systems
Military Technical Academy, Bucharest, Romania
dinu.dumitrudaniel@yahoo.com, mtogan@mta.ro

***Abstract*—DHCP is one of the most used network protocols, despite the security issues it has. Our work is motivated by the numerous attacks that can be launched against DHCP and the impact that they can have. Firstly, we formulate the constraints and design principles for a DHCP message authentication module that is flexible and easy to integrate with current DHCP implementations, while providing the necessary level of security. Then we present DHCPAuth, a module for authenticating DHCP messages. The module uses the RFC 3118 authentication option format and is able to authenticate DHCP client and server messages using two trust models: PKI and PGP. The proposed module is evaluated using different public key pairs in the considered trust models to determine the overhead introduced and the impact on DHCP operation. Results show the additional time required to process the DHCP messages, either when signing or verifying the signatures, as well as the authentication option length and the DHCP packet length. We also provide an analysis of worse case time for verifying the authentication option when more certificates or public keys are available on certificate store or public key ring. These information can help network administrators in selecting the trust model, the key types and sizes to use.**

I. INTRODUCTION

Nowadays the society is more interconnected than ever [1]. Each person has several devices to connect with others. These connections are usually realized through private networks or Internet. Almost all of these connections are done using TCP/IP protocol suite. Few of these devices have their own public IP address. The rest of them are using private IP addresses, which are either statically or dynamically allocated. Dynamic address allocation is more widespread because of the advantages that it provides. But these advantages are not for free. Dynamic address allocation using DHCP exposes the organization to a number of vulnerabilities that are described later in this paper. Our work is motivated by these vulnerabilities and the need for securing the network configuration information allocation via DHCP. We also had in mind the fact that although the migration from IPv4 to IPv6 has already started, there are still a lot of devices that are using IPv4 and the transition process will take some time. During this time a lot of devices and users are still not safe.

Let us assume that a coffeehouse is providing Internet access to customers. The coffeehouse team does not want to change a secret password each day and share it to customers. They want a method that prevents customers from neighboring coffeehouses to connect to their network as well as someone from the public park in front of the coffeehouse. In the same time the coffeehouse prefers to authenticate the customers who try to connect to the network using the recommendations from previous customers. In this case using the web of trust model built around PGP could be a very good solution. The coffeehouse customers prefer also to authenticate the server that is providing the configuration option to be sure that they are connecting to a trusted network and not to a rogue access point advertised as a trusted network. The PKI trust model can be used to authenticate DHCP server messages.

Our contribution is twofold. Firstly, we analyze the requirements and constraints of a good DHCP authentication module that is flexible, easy to adopt and secure in the same time. Then we design a module that authenticates all DHCP messages sent either by a server or a client. The proposed module is, to the best of our knowledge, the first one that uses two trust models that provide different levels of flexibility. Secondly, we implemented the DHCPAuth module and evaluated the implementation in different usage conditions. The results can be used by network administrators to select the appropriate level of security they need, considering in the same time the DHCP message overhead and the additional time required to process the authentication information.

The rest of the paper is organized as follows. Section II provides an overview of basic concepts. In Section III we describe the constraints and design principles for a flexible, easy to adopt and secure authentication module, then we introduce the proposed module. Section IV evaluates the DHCPAuth module in different usage conditions. Section V provides a short description of related work and Section VI concludes the paper.

II. BACKGROUND

A. DHCP

DHCP was introduced in [2] as an improvement of Bootstrap Protocol (BOOTP) and then updated by [3]. It allows the network administrator to automatically distribute network configuration information to devices in the network, reducing the human work and increasing the flexibility and mobility.

When it was designed, more than fifteen years ago, security was not a primary concern. Meanwhile, the security plays an important role and DHCP protocol lacks some security mechanisms. Therefore, the protocol is vulnerable to a number of attacks. We classify these attacks in two classes, depending on whose identity is stolen by the attacker: rogue server and malicious client. A rogue DHCP server can provide incorrect network

978-1-4799-9911-8/15/$31.00 ©2015 IEEE

configuration information to clients only for fun or to be able to build more complex and devastating attacks like sniffing, man in the middle (MITM) or phishing. A malicious DHCP client can get unauthorized access into network and use the network services without being allowed to. He can even create other attacks like denial of service (DoS), making impossible for legitimate users to obtain access into the network.

In [4] is proposed a standard for authenticating DHCP messages using an authentication option and two authentication methods based on a secret shared by client and server. The document lists as limitation the impossibility of using a digital signature mechanism such as RSA due to computational complexity, although it would provide better security.

The limitation mentioned in [4] could be overcome after in [5] was defined a method for encoding long options inside a DHCP message. It describes how to encode more than 255 bytes of data in the same option, repeating the same option code several times and using *sname* and *file* fields of the DHCP message when they are not used for their initial purpose.

B. PKI

Public Key Infrastructure (PKI) is a framework that allows generation, verification and revocation of public key certificates using a Registration Authority (RA) and a Certificate Authority (CA). The digital certificate is used to prove the ownership of a private key and in the same time the identity of an entity.

PKI uses a centralized trust model, where each entity trusts the CA, which is able to issue digital certificates for users. The information about the revoked certificates is provided in the form of a Certificate Revocation List (CRL) by the CA. To verify a digital signature, a user has to check the validity of the digital certificate associated with that signature and then to check if the digital signature was generated using the private key that corresponds to the public key from digital certificate.

Despite the rigid trust model used, PKI has some weaknesses as pointed in [6, 7]. As seen in the case of DigiNotar [8] and Comodo [9] certificate authorities, the model is not completely secure.

C. PGP

Pretty Good Privacy (PGP) is a software used for encryption, decryption, signing and verifying signatures. The first PGP specifications were published in [10]. PGP and similar software now follow the OpenPGP standard specifications defined in [11] and updated by [12].

The OpenPGP implementations include a vetting scheme, which can be used to determine which certificate will be trusted. A user accepts another user's public key certificate in the key ring as valid if at least one of the conditions is met:

- The key belongs to the owner of the key ring.
- The key ring contains at least *C* certificates from completely trusted introducers with valid public keys.
- The key ring contains at least *M* certificates from marginally trusted introducers with valid public keys.

The scheme is flexible and leaves the trust decision in the hands of users which can adjust the *C* (completes needed) and *M* (marginals needed) parameters [13].

The decentralized trust model used in PGP is called web of trust. The principle of this trust model is that each user builds its own trust in other user's public key using the trust in that public key of other trusted users called introducers. Although it is very flexible, PGP has several disadvantages derived from the trust model and OpenPGP specifications [14, 15].

III. MODULE DESCRIPTION

A. Constraints

Before formulating the design principles of a good DHCP authentication module, we presents the main limitations of the protocol. These limitations affect the design of an authentication module which is closely related to the protocol.

The DHCP specifications don't allow to split messages over several IP packets. This leads to a maximum DHCP packet size of 1500 bytes. Eliminating 8 bytes required for UDP datagram header and 20 bytes required for IP packet header without options, we get 1472 bytes. Subtracting the DHCP message header of 236 bytes, we get 1236 bytes for DHCP options. Considering that the *sname* and *file* fields can be used for DHCP options through the DHCP option overload, it results a total of 1428 bytes for DHCP options in the best case. This amount is so small and it cannot allow the transmission of a digital certificate or a longer public key in a single DHCP message.

B. Design Principles

In order to be ease the integration of the authentication module in current DHCP implementations and also to offer compatibility with DHCP implementations without the module, we formulated the following design principles.

The authentication option used should follow the format defined in [4]. This format contains enough fields to provide the required level of security and with small changes allows easy integration of other authentication methods.

The DHCP client state machine should not be modified by introducing new states. Modifying DHCP client state machine implies major changes in DHCP operation. They can lead to interoperability issues between DHCP entities with the authentication module enabled and DHCP entities without the authentication module enabled.

The module must not introduce new DHCP messages or options. If the proposed module will introduce new DHCP messages, it will modify the DHCP specifications. But we want to avoid major changes in the protocol operation. Introducing new DHCP options requires upgrading the current DHCP implementations and decreases the acceptance level.

The authentication module should not introduce unnecessary communication with other servers or services. This rule is given to ease the adoption of the proposed solution by not forcing dependencies with other entities. Even though we cannot restrict the communication between DHCP entities and other services, we do not constrain the operation of authentication module by the communication with other servers or services.

The server module must be able to verify the DHCP client requests authenticity and to authenticate the sent responses. In the same way, the client module must be able to authenticate the client requests and also to verify the DHCP server responses. To maintain compatibility with DHCP implementations without the authentication module, a server without the module should be able to process a request coming from a client with the module enabled. Similarly, a client without the module enabled, must be able to process the responses from a DHCP server with the module enabled.

A DHCP entity with the authentication module enabled should not accept and process messages that are coming from entities without the authentication module enabled. Also the messages that do not contain a valid authentication option (valid signature or valid replay detection value) should be discarded by the module.

In addition to authenticating the DHCP messages, the module must prevent the replay attacks. This requirement is met by using the *Replay Detection* field from the authentication option.

C. DHCPAuth Overview

DHCPAuth is an authentication module for DHCP that follows the design principles previously formulated. It is integrated in the DHCP implementation and acts like a firewall for the protocol:

- All incoming DHCP messages are firstly verified by the authentication module. If messages are valid, they are passed to DHCP engine. If an incoming DHCP message is not valid, it will never reach the DHCP engine.
- All outgoing DHCP messages are passed through authentication module before being sent over the network. The module adds the authentication information to each DHCP message and signs the message content using the sender's private key.

The proposed module is able to authenticate DHCP messages using the two trust models described: PKI and PGP. The right trust model for authenticating DHCP clients and servers is at the choice of the network administrator. Although the module provides the flexibility of choosing the preferred authentication method, we suggest using PKI trust model for authenticating DHCP server responses and the web of trust for authenticating client requests. The rigidity of the PKI trust model allows clients to have an increased level of confidence that the DHCP server to whom they are communicating is who it pretends to be. On the other hand, using PKI authentication for clients would be more complicated than using PGP authentication and will imply additional costs, reducing the usability. For these reasons, authenticating DHCP clients using the web of trust is a better solution. It reduces the costs and offers, in the same time, a good tradeoff between the security and usability.

D. Authentication Option Format

The authentication option used by DHCPAuth follows the format defined in [4] with minor changes. To be able to accommodate digital signatures the *Authentication Information* field is split into two fields: *Signature Length* and *Message Signature*. The *Signature Length* field allows a verifier to recover the signature from the *Message Signature* field.

For the *Protocol* field we introduce two new values. We use value 2 for PKI signatures and 3 for PGP signatures. For the *Algorithm* field we use value 2 for SHA-1 message digest and value 3 for SHA-2 message digest. For *RDM* field we use the same value as in the authentication option specification, which means that the *Replay Detection* field will contain a monotone strictly increasing value. To obtain the *Replay Detection* field value, the DHCPAuth module uses the current time of the machine it is running on. The value is expressed in seconds and microseconds of the remaining second fraction from the Unix epoch.

E. Sent Message Processing

Each DHCP message sent by client or server passes through DHCPAuth module before being sent through the network. The module adds the authentication option to the message to be sent, then it fills the authentication option fields. The DHCPAuth module uses the current time of the machine it is running on to fill the *Replay Detection* field. In the end, it signs the DHCP message using the given private key. The signature is computed over the entire DHCP message, including the authentication option. The *hops*, *giaddr*, *Signature Length* and *Message Signature* field values are set to zero. These fields are set to zero because *hops* and *giaddr* fields can be modified by a relay agent, while the *Signature Length* and *Message Signature* are not known before computing the digital signature.

The DHCP users have the freedom to configure which sent DHCP messages they want to authenticate and which not. For the sent authenticated messages they can choose between PKI and PGP signatures. This is done using the DHCPAuth configuration file. If the module is enabled, each message to be sent passes through it, even those that don't have to be authenticated. DHCPAuth module passes the messages that should not be authenticated to the network without any change.

F. Received Message Processing

In the case of DHCP clients and servers with the DHCPAuth module enabled, each received message is firstly processed by the module and if everything is correct, the message is then processed by DHCP engine. If the module configuration indicates that the message must not be authenticated, the received message is passed to the DHCP engine. If the received DHCP message must be authenticated, the DHCPAuth module verifies if the received message contains an authentication option. If the received message does not contain the authentication option, then the message is dropped. If the message contains an authentication option, then the DHCPAuth module verifies the *Replay Detection* field value and then the message signature. If one of these verifications fails, the DHCP message is discarded.

To be able to verify the *Replay Detection* field value, the DHCPAuth module maintains a table with a binding between the message transaction identifier field *xid* and the last received *Replay Detection* field value. If the current received value is strictly higher than the stored one, the table value is updated and the processing continues. If the received value is not higher than the stored one, the received DHCP message is discarded because it does not contain a valid replay detection value. In the case that no entry in the DHCPAuth module table contains the received message transaction identifier, a new entry is added and the message is considered valid.

To be able to verify the messages received the DHCP entity must have or be able to quickly obtain the sender's public key or certificate. For our implementation we considered that the verifying entity has in its local store the certificates or public keys of authorized entities. The acquisition of certificates and public keys can be done by different means. We suggest downloading the certificate of the DHCP server from a share server before communicating to DHCP server and updating the DHCP server key ring with public keys of DHCP clients from a public key server at least once a day.

G. *Security Considerations*

The proposed module, prevents all major security vulnerabilities that a DHCP client or server can encounter.

Authenticating all DHCP messages, the DHCPAuth module prevents illegitimate clients to get access to a network without being allowed to. A malicious client can send as much DHCP request he wants to a DHCP server with DHCPAuth module enabled with no effect. The unauthenticated messages will be discarded by the authentication module and will never be processed by the DHCP server. There is the possibility that a legitimate client to create a large amount of DHCP request. In this case the DHCP server may exhaust all configuration information available. We don't consider this a major threat to DHCP security since the client is trusted by the server. Although, if we want to protect against insider attacks, we can improve the DHCPAuth module with a history of requests signed with the same private key, such that the module blocks the user making more requests than a given threshold. Another simple way to prevent this attack is to change the trust in client key after a preset number of requests in a given amount of time.

Using the DHCPAuth module, the DHCP clients are at safe against rogue DHCP servers. Just verifying the authenticity and validity of received responses, DHCP clients can prevent more dangerous attacks. A rogue DHCP server will never be able to trick a user which checks the server responses using DHCPAuth module, unless it manages to steal the private key of a valid DHCP server. Considering the fact that DHCPAuth prevents users from obtaining configuration information from untrusted DHCP servers, attacks like phishing and MITM become ineffective.

Trying to replay a previously captured DHCP message will not be effective because of the replay detection checking that is done by the DHCPAuth module. Replacing the *Replay Detection* field value with a new one is almost impossible, because the change is automatically reflected in the DHCP message signature.

Hence, the DHCP security increases and the only possible way to bypass the DHCPAuth security is by compromising the secret keys used for authenticating DHCP messages. Although this is not impossible, it can be very difficult if DHCP users (clients and servers) follow some simple rules regarding the manipulation and storing of secret keys.

Even though the proposed DHCPAuth module achieves its goal to protect DHCP clients and servers against the presented attacks, it also increases the chances of a DoS attack. This is mainly because the signature verification process is time consuming. The best method to prevent DoS attacks is to limit the number of requests processed in the amount of time and to speed the signature verification operation.

H. *Implementation*

The DHCPAuth module was implemented in C language using the ISC DHCP software [16], which is present in all major Linux distributions. For PKI functions we used the open source cryptographic library OpenSSL [17]. The PGP operations are provided by GPGME library which runs over GnuPG [18] implementation of OpenPGP standard. The modifications to original DHCP source code are minimal. The DHCPAuth module is provided in the form of a patch [19] which can be applied to ISC DHCP implementation.

IV. Evaluation

In this section we evaluate the DHCPAuth module implementation to understand the impact that it has on protocol operation. We measured the authentication option length, the DHCP packet length, the authentication option processing time and DHCP message processing time for different key pairs. On the client we measured the time to sign the DHCPREQUEST message and the time to verify the DHCPACK message. On the server we measured the time required to verify the received DHCPREQUEST message and sign the sent DHCPACK message. The authentication option length and the DHCP packet length were measured for the same DHCP messages on client and server.

We also tried to find if there are differences between the two trust models, as well as the worst processing time for verifying the messages depending on the number of PKI certificates or PGP public keys.

A. *Processing Time*

We used different PKI and PGP key pairs to determine the additional time required to sign or verify the authentication option. To be able to determine how much of the total processing time of a DHCP message is required by the authentication option processing, we also included the time to process the DHCPREQUEST and DHCPACK messages.

Fig. 1 shows the time required to create the authentication option for DHCPREQUEST message and the time spent to verify the DHCPACK message authentication option for different RSA key sizes. We can see that the time for creating the authentication option for the client request is almost equal with the time required to process the DHCPREQUEST message. This is a big overhead, but if we consider that the actual time for processing the message is less than 40 milliseconds for a 3072 bits key, this is not really bad. On the other hand verifying the authentication option from DHCPACK message takes considerably less than the total time required to process the DHCP message. In this case the overhead introduced by the authentication option processing is very small.

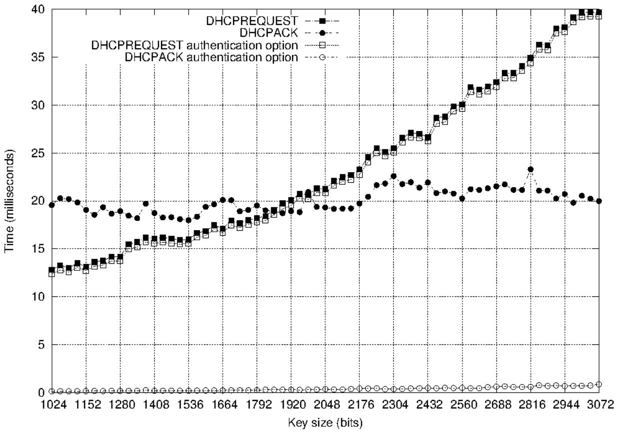

Figure 1. RSA signature processing time on DHCP client

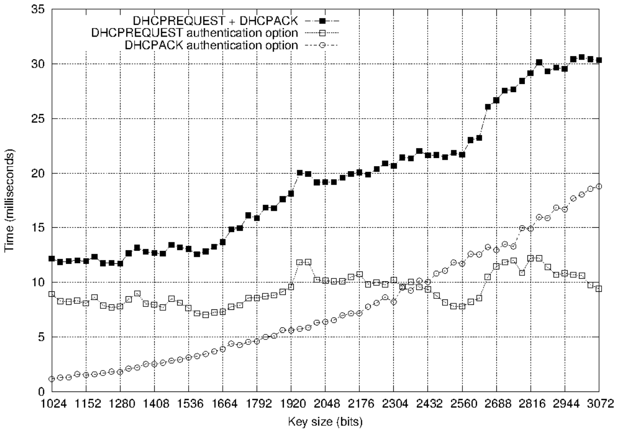

Figure 2. RSA signature processing time on DHCP server

Table I. DSA signature processing time

Key size (bits)	Client processing time				Server processing time		
	REQUEST		ACK		REQUEST[1] + ACK[2]		
	Total (ms)	Auth (ms)	Total (ms)	Auth (ms)	Total (ms)	Auth[1] (ms)	Auth[2] (ms)
1024	13.81	13.34	19.33	1.01	12.17	9.89	0.61
2048	26.07	25.57	24.94	5.50	20.51	15.98	2.76
3072	39.71	39.27	26.08	10.88	29.58	21.58	6.18

The time spent for processing the received DHCPREQUEST message and creating the sent DHCPACK message is shown in Fig. 2. For RSA key sizes less than 2400 bits, processing the client request authentication option requires more time than creating the authentication option for DHCPACK message. For key sizes greater than 2400 bits, the time required to process the DHCPREQUEST authentication option is less than the time to create the DHCPACK authentication option.

The time required to create the DHCPREQUEST message on client is higher than the time required by server to verify the request and send the DHCPACK authenticated message. For the same key size, signing using PGP takes more than using PKI. Verifying PGP authentication option requires more time than verifying PKI authentication option.

B. *Message Overhead*

The authentication option length increases as the key size increases, reaching 428 bytes for DHCPREQUEST message and 397 bytes for DHCPACK message. The PKI signature size is always smaller, than the PGP signature for the same key size. The difference is given by the PGP signature information which is embedded in the actual signature.

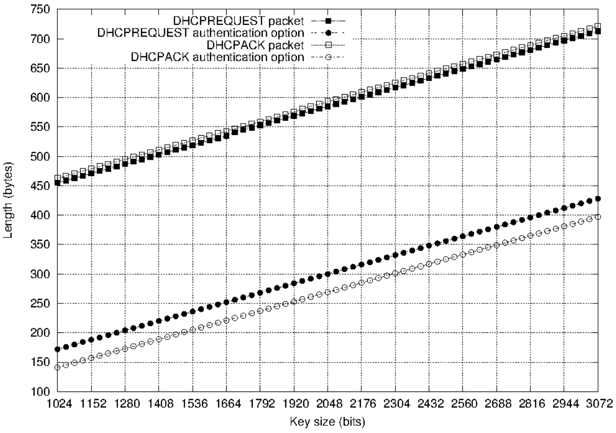

Figure 3. RSA signature authentication option overhead

C. *Worst Case Processing Time*

For the first received message in a communication session, the client has to identify the certificate of the server in the certificate store. Similarly, the server has to identify the client's public key in the public key ring to be able to verify the client signature. Table II gives some experimental values of the time required to process the authentication option of the first DHCP message depending on the number of certificates or public keys.

Table II. Worst case processing time

Number of PKI certificates / PGP public keys	Client processing time (ms)	Server processing time (ms)
1	1.02	11.40
2	1.59	11.33
5	1.98	10.40
10	5.43	10.75
20	5.61	10.86

V. RELATED WORK

There were a lot of attempts to increase the security of DHCP using different methods. While some of them failed to achieve their goals because they didn't consider the real constraints of DHCP [20, 21, 22, 23, 24, 25], other were too complex or required massive changes in protocol operation [26, 27, 28, 29, 30].

We note that in [31], the authors made a good review of literature, showing why previously proposed methods to secure DHCP failed to be widely accepted and used. The paper also presents a way to authenticate DHCP messages sent by DHCP server using public key certificates. Our approach improves the solution from this paper by adding the concept of certificate store to DHCPAuth module. The certificate store allows DHCP entities to authenticate messages from different senders. Our time measurements include the time for selecting the correct certificate to authenticate the incoming DHCP messages. Although the method proposed in [31] increases the client security, the server is still vulnerable to attacks that can be launched by DHCP clients. This is not the case for DHCPAuth.

DHCPAuth contains two authentication mechanisms. If several PKI authentication methods were previously proposed, to the best of our knowledge, there is no previous approach to secure DHCP messages using the web of trust built around PGP.

VI. Conclusion

In this paper we presented the DHCP security issues and motivated by these, we formulated a set of requirements for a flexible DHCP authentication module considering the protocol limitations. The design principles were formulated to ease the adoption and integration of the authentication module with existing DHCP implementations. Then we introduced DHCPAuth module. The module is able to authenticate DHCP messages using two different trust models: PKI and PGP, allowing different levels of flexibility without affecting the security or usability. The DHCPAuth module implementation was evaluated using different key types and sizes. The results indicate that the module requires several milliseconds to create or process the authentication option. The clients and servers do not suffer additional delays and the client state machine operation is not affected. We analyzed the overhead introduced by the authentication option for different key sizes in the studied trust models. The authentication option length increases as the key size increases.

While the DHCP protocol security has been studied a lot and different authentication solution were proposed, all of them failed to be widely adopted. These in mainly because they didn't consider the real constraints of DHCP or because they were too complex and reduced the usability too much. Our solution offers two levels of flexibility, which correspond to the two trust models used. Although solutions to authenticate DHCP messages based on digital certificates were previously proposed, to the best of our knowledge, there is no previous DHCP message authentication solution based on the trust model built around PGP.

References

[1] Internet Live Stats, "Internet users in the world". [Online]. Available: http://www.internetlivestats.com/

[2] R. Droms, "Dynamic Host Configuration Protocol", IETF, RFC 2131, March 1997.

[3] S. Alexander, R. Droms, "DHCP Options and BOOTP Vendor Extensions", IETF, RFC 2132, March 1997.

[4] R. Droms, W. Arbaugh, "Authentication for DHCP Messages", IETF, RFC 3118, June 2001.

[5] T. Lemon, S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", IETF, RFC 3396, November 2002.

[6] Carl Ellison, Bruce Schneier, "Top 10 PKI risks". *Computer Security Journal*, Volume XVI, Number 1, 2000.

[7] Peter Gutmann, "PKI: it's not dead, just resting", *IEEE Computer*, Volume 35, Issue 8.

[8] Wikipedia, "DigiNotar". [Online]. Available: http://en.wikipedia.org/wiki/DigiNotar

[9] Wikipedia, "Comodo". [Online]. Available: http://en.wikipedia.org/wiki/Comodo_Group

[10] D. Atkins, W. Stallings, P. Zimmermann, "PGP Message Exchange Formats", IETF, RFC 1991, August 1996.

[11] J. Callas, L. Donnerhacke, H.Finney, R. Thayer, "OpenPGP Message Format", IETF, RFC 2440, November 1998.

[12] J. Callas, L. Donnerhacke, H. Finney, D. Shaw, R. Thayer, "OpenPGP Message Format", IETF, RFC 4880, November 2007.

[13] Audun Jøsang, "An Algebra for Assessing Trust in Certification Chains", Proceedings of the Network and Distributed Systems Security Symposium (NDSS'99).

[14] Secure Share, "15 reasons not to start using PGP". [Online]. Available: http://secushare.org/PGP

[15] Jacek Jonczy, Markus Wuthrich, and Rolf Haenni, "A Probabilistic Trust Model for GnuPG", 23rd Chaos Communication Congress, Berlin, 2006.

[16] Internet System Consortium, "ISC DHCP: Enterprise Grade Solutions for Configuration Needs". [Online]. Available: https://www.isc.org/downloads/dhcp/

[17] OpenSSL, "Welcome to OpenSSL Project". [Online]. Available: https://www.openssl.org/

[18] GnuPG, "The GNU Privacy Guard". [Online]. Available: https://www.gnupg.org/

[19] DHCPAuth, "A DHCP Message Authentication Module". [Online]. Available: https://github.com/daniel-dinu/DHCPAuth

[20] Glenn Glazer, Cora Hussey, Roy Shea, "Certificate-Based Authentication for DHCP", March 2003.

[21] Jacques Demerjian, Ahmed Serhrouchni, "DHCP Authentication Using Certificates", Security and Protection in Information Processing Systems, IFIP Advances in Information and Communication Technology, Vol. 147, Springer, 2004.

[22] Surakarn Duangphasuk, Supakorn Kungpisdan, Sumeena Hankla, "Design and Implementation of Improved Security Protocols for DHCP Using Digital Certificates", ICON, Singapore, December 2011.

[23] Tuomas Aura, Michael Roe, Steven J. Murdoch, "Securing Network Location Awareness with Authenticated DHCP", 3rd Internation Conference on Security and Privacy in Communication Networks (SecureComm), Nice, France, September 2007.

[24] HongIl Ju, JohgWook Han, "DHCP Message Authentication with an Effective Key Management", World Academy of Science, Engineering and Technology, August 2005.

[25] Narendar Shankar, William A. Arbaugh, Kan Zhang, "A Transparent Key Management Scheme for Wireless LANs Using DHCP", HP Laboratories, Palo Alto, September 2001.

[26] Y. Xu, S. Manning, M. Wong, "An Authentication Method based on Certificate for DHCP", DHC Internet Draft, September 2011.

[27] Craig A. Shue, Andrew J. Kalafut, Minaxi Gupta, "A Unified Approach to Intra-Domain Security", IEEE International Symposium on Secure Computing (SecureCom), August 2009

[28] Kathryn De Graaf, John Liddy, Paul Raison, John C. Scano, Sanjay Wadhwa, "Dynamic Host Configuration Protocol (DHCP) Authentication using Challenge Handshake Authentication Protocol (CHAP) Challenge", United States Patent Application Publication, 2011.

[29] Ken Hornstein, Ted Lemon, Bernard Adoba, Jonathan Trostle, "DHCP Authentication Via Kerberos V", IETF DHC Working Group, October 2001.

[30] Tadashi Komori, Takamichi Saito, "The Secure DHCP System with User Authentication", Proceedings of the 27th Annual IEEE Conference on Local Computer Networks, November 2002.

[31] D. D. Dinu, M. Togan, "DHCP Server Authentication using Digital Certificates", The 10th International Conference on COMMUNICATIONS (COMM2014), Bucharest, May 2014.