CERTIFICATE-BASED ACCESS CONTROL AND AUTHENTICATION FOR DHCP

Jacques Demerjian, Ahmed Serhrouchni
GET-Télécom Paris – LTCI-UMR 5141 CNRS,46 Rue Barrault,Paris, France

Mohammed Achemlal
France Telecom R&D – DMI/SIR – 42, Rue des Coutures, BP 6243, 14066 Caen Cedex 4, France

Keywords: Access Control, Attribute Certificate, Authentication, DHCP, PKI, PMI, X.509 Identity Certificate.

Abstract: In the current Dynamic Host Configuration Protocol, security is not considered. DHCP itself does support neither an access control for a proper user nor the mechanism with which clients and servers authenticate each other. In this paper, we introduce a novel authentication and access control mechanism for DHCP systems. This solution defines a new DHCP option that provides the authentication of both, entities (client/server) and DHCP messages. We built up our mechanism on the use of public key cryptography, X.509 identity certificates and attribute certificates. In addition, the PMI (Privilege Management Infrastructure) functionalities are attributed to a new server that groups DHCP server and AA (Attributes Authority) server. The resulting server creates an attribute certificate to the client that will be used then in the access control.

1 INTRODUCTION

Dynamic Host Configuration Protocol '*DHCP'* (Droms, 1997a) was developed to support automatic host configuration. DHCP is built directly on UDP (Postel, 1980) and IP (DelRey, 1981), which are as yet inherently insecure. However, security in DHCP framework is not sufficient, because security considerations around DHCP were intentionally omitted in the IETF standardization process. DHCP does not support the mechanism with which clients and servers authenticate each other. If the client configures network resources such as IP address, any client can use the network. In addition, current DHCP servers allocate network resources to any client that requests them. To solve these problems, several authentication methods for DHCP messages have been proposed. However, they have several drawbacks.

In this paper, we propose an extension (Droms, 1997b) to DHCP protocol in order to allow a strict control on the equipments through a strong authentication. This extension ensures on one hand, the authentication of the entities (clients and servers) and DHCP messages and, on the other hand, the access control to a DHCP system.

The remainder of this paper is structured as follows: Section 2 introduces DHCP design and operation; section 3 presents DHCP vulnerabilities, section 4 explores some existing contributions that define how authentication should be handled in DHCP, and exposes their limits. Section 5 introduces some essential background and concepts used in our solution. Section 6 illustrates our proposed authentication mechanism called CACAD (*Certificate-based Access Control & Authentication for DHCP*). Section 7 concludes this paper and gives directions for future work.

Demerjian J., Serhrouchni A. and Achemlal M. (2004).
CERTIFICATE-BASED ACCESS CONTROL AND AUTHENTICATION FOR DHCP.
In *Proceedings of the First International Conference on E-Business and Telecommunication Networks*, pages 99-106
DOI: 10.5220/0001396100990106
Copyright © SciTePress

2 DHCP DESIGN AND OPERATION

Dynamic Host Configuration Protocol is designed around a traditional client/server operation model. DHCP provides a mechanism to automate and manage network configuration of desktop computers and other network devices that use TCP/IP protocol. DHCP is based on Bootstrap Protocol '*BOOTP'* (Croft, 1985). DHCP retains the basic message format of *BOOTP* and *BOOTP* relay agents operation, and shares UDP ports initially assigned to *BOOTP* (67 and 68). This backward compatibility with *BOOTP* allows DHCP to use the BOOTP relay agents installed base and avoid the requirement of a DHCP server on every network segment. A key advantage of DHCP over *BOOTP* is that addresses can be assigned dynamically. Additionally, DHCP allows recovery and reallocation of network addresses through a leasing mechanism.

By using DHCP, dynamically configuring the host on the network is done by a simple handshake. DHCP clients and server interact through a series of client-initiated request-response transaction (fig.1).

DHCP Message type	Sent by	Description
Discover	Client	Locate available DHCP servers and request configuration parameters
Offer	Server	Reply to a DHCPDiscover message and offer to provide configuration
Request	Client	Request specific network address and configuration parameters
Ack	Server	Reply of a server that contains parameters and an IP address.
Nak	Server	Use to tell a client that its lease is over or that the configuration it has chosen is wrong
Decline	Client	Decline offered parameters; for exemple, client has detected address already in use
Release	Client	Client release its current configuration and allocated address to server
Inform	Client	Client has address and request other parameters

Figure 1: Types of DHCP message

The process to be followed to get configuration data from DHCP server can be divided into two steps.

In the first step, the client broadcasts a *DHCPDiscover* message to collect proposals from servers. The client may specify preference of a lease and/or an IP address.

A DHCP server receiving the *DHCPDiscover* message may or not return *DHCPOffer* message (Many servers may receive the *DHCPDiscover* message). If a server decides to respond, it offers a selection of configuration parameters and puts an available address into *yiaddr* field and broadcasts the *DHCPOffer* to the client. At this point, there is no agreement of an assignment between the server and the client.

In the second step, the client gets one or more *DHCPOffer* and chooses one server from them. The client puts the IP address of the chosen server into the '*Server identifier'* option of a *DHCPRequest* and broadcasts it over the network. Each server checks the '*Server identifier'* option. If it does not match its own address, the server considers it as an implicit decline. The selected server sends the *DHCPAck* (if its address is available) or the *DHCPNak* (for example, the address is already assigned to another client).

The client which gets the *DHCPAck* starts using the IP address. If it gets *DHCPNak*, it restarts to broadcast a *DHCPDiscover* message. If the client finds a problem with the assigned address of *DHCPAck*, it sends *DHCPDecline* to the server, and broadcasts a new *DHCPDiscover*. The client can release the address before its lease expires by *DHCPRelease* (Tominaga, 1995).

3 DHCP VULNERABILITIES

There was no attempt in the design of DHCP to protect against malicious Internet hosts, and consequently the protocol is vulnerable to a variety of attacks. Since the DHCP server doesn't do any authentication of client *DHCPDiscover* requests, an intruder can impersonate the identity of any client that divulges its identification information (Perkins, 1995). Likewise, an intruder can impersonate a DHCP server, and send erroneous information to any local DHCP client.

DHCP itself doesn't have an access control for a proper user. So, malicious users inside the network segment can easily abuse IP addresses and the network. To solve this problem, introduction of a MAC *(Message Authentication Code)* address authentication scheme has been proposed, whereby, the MAC address of the equipment must be registered on the DHCP server before accessing the network. When an IP address is requested, the server authenticates the equipment by the MAC address.

Using authentication by MAC address constrains the user to use the IP address affected by the DHCP server on the terminal with the same MAC address. In this mechanism, DHCP server authenticates the terminal through its MAC address rather than the client. However, since only registered terminals can use an IP address, as it stands, the MAC authentication is inconvenient. Moreover, illegitimate users who fabricate a MAC address can

easily deceive the DHCP server and obtain an IP address (Komori, 2002). Therefore, DHCP in its current form is quite insecure. Hence, for all of these problems, we need stricter new authentication mechanisms, which can provide both authentications, of entity (DHCP client/server) and DHCP content messages.

4 EXISTING AUTHENTICATION MECHANISMS

Several different contributions regarding how DHCP should be authenticated already exist. Among them:

1. DHCP Authentication via Kerberos V (Hornstein, 2000): This authentication method authenticates the client only, and involves communication with the Kerberos server, in addition to the DHCP standard communication.

2. Token Authentication (Droms, 2001): This involves sending a token such as a plaintext password from the client to the server to identify the client. This protocol provides only weak entity authentication and no message authentication. This mechanism is vulnerable to interception and provides only the most rudimentary protection against inadvertently instantiated DHCP servers.

3. Delayed Authentication (Droms, 2001): This requires a shared secret key for each client on each DHCP server with which that client may wish to use the DHCP protocol. Each secret key has a unique identifier that can be used by a receiver to determine which secret was used to generate the MAC in the DHCP message. The server and the client authenticate each other by the MAC included with the DHCP message. Delayed Authentication is the most secure and interesting contribution for DHCP Authentication, which has been more formally designed and accepted than many of the others. The main issues of this option are key distribution and key flexibility. None of these affect the security of the protocol, but both have potential to affect its applicability in practice.

The first issue is one of the major drawbacks to the use of shared keys (Glazer, 2003). Their distribution is complicated. The technical specification of Delayed Authentication itself attempts to remedy this and suggests using a master server key with multiple client keys to simplify the key distribution, but this can decrease system security.
The second issue (flexibility) to using shared keys becomes apparent when the client switches between networks. Different networks should require different keys, and this introduces a new issue with shared key management: the key chain. Management of multiple shared secret keys can quickly become cumbersome. A real digital signature mechanism such as RSA (Jonsson, 2003), would provide a better security.

The delayed authentication option is exposed to additional drawbacks:

a) It is vulnerable to a denial of service attack through flooding with *DHCPDiscover* messages, which are not authenticated by this protocol. Such attack may overwhelm the computer on which the DHCP server is running and may exhaust the addresses available for assignment by the DHCP server.

b) It does not support inter-domain authentication.

c) It may also be vulnerable to a denial of service attack through flooding with authenticated messages, which may overwhelm the computer on which the DHCP server is running as the authentication keys for the incoming messages are computed.

4. Certificate-Based DHCP Authentication *'CBDA'* (Glazer, 2003): This authentication method uses X.509 identity certificates to authenticate DHCP entities. CBDA involves sending X.509 identity certificate or certificates chains with a common signer as an option in *DHCPDiscover* and *DHCPOffer* packets, and then sending only signed hashes of the packets in *DHCPRequest* and *DHCPAck* packets.
The standard 576 byte maximum size for a DHCP message may be too short to contain X.509 identity certificate or certificates chains. Certificates are very large and DHCP packets were originally designed to be relatively small. Because of this, clients implementing CBDA should send a Maximum DHCP Message Size (Droms, 1997b) option if DHCP client's TCP/IP stack is capable of receiving larger IP datagrams. In this case, the client should set the value of this option to at least the MTU (*Maximum Transmission Unit*) of the interface that the client is configuring. The client may set the value of this option higher, up to the size of the largest UDP packet it is prepared to accept. Note that the value specified in the Maximum DHCP Message Size option is the total maximum packet size, including IP and UDP headers. DHCP clients requesting this option, and DHCP servers sending this option, must implement DHCP option concatenation (Lemon, 2002).

In (Droms, 1997a), no universal limit exists for DHCP packets, but in practice there necessarily needs to be a limit to prevent flooding of a host. Clients may specify the maximum length of DHCP packet they will accept, and many of these limits may need to be redesigned if long certificate chains are used.

5 ESSENTIAL BACKGROUND AND CONCEPTS

This section describes some essential background and concepts upon which our solution is based.
In order to access a resource, both authentication and authorization are needed. PKI (*Public Key Infrastructure*) can provide a strong authentication support for a system by using PKCs (*Public Key Certificate*), while PMI (*Privilege Management Infrastructure*) can provide authorization support for a system by using ACs (*Attribute Certificate*). The use of public-key certificates proves the identity of the certificate holders. X.509 certificate is widely accepted as the appropriate format for public key certificates (Demerjian, 2003).
Similar to PKC, an AC binds the attributes such as group membership, roles, or other authorization information associated with the AC holder to that entity through the signature of a so-called AA (*Attribute Authority*). As outlined in (Farrell, 2002), an AC may consist of the following fields:
Version: This field indicates the version (1 or 2) of the AC format in use.
Holder: This field is used to bind an attributes certificate to an X.509 PKC (fig.2). The *Holder* field identifies the client with which the attributes are being associated. Identification can be either by name or by reference to an X.509 PKC. The holder's PKC *serialNumber* and *issuer* must be identical to the AC holder field.
Issuer: This field identifies the AA that issued the AC.
Signature: This field indicates AC digital signature algorithm.
Serial Number: This field contains a unique AC serial number.
Validity Period: This field contains a time period during which the AC is assumed to be valid.
Attributes: This field contains information (*SEQUENCE OF Attribute*) concerning the AC Holder. Each Attribute may contain a set of values.
Issuer Unique Identifier: This field is used to make the name of the issuing AA unambiguous, in case where the same name was reassigned to different authorities through time. This field is optional.

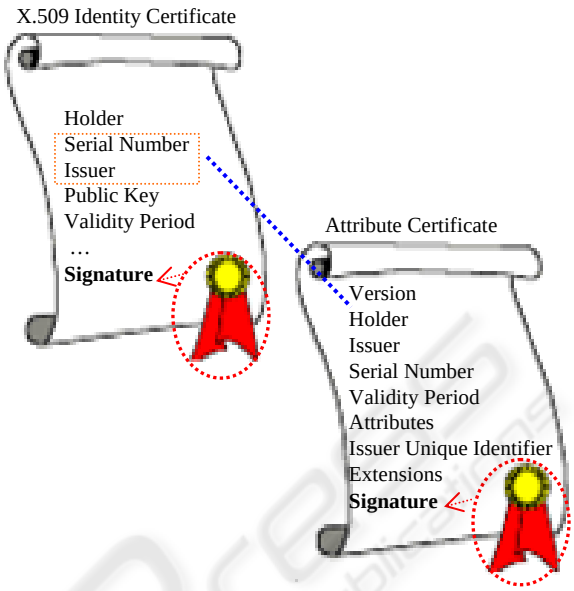

Figure 2: The link between Attribute Certificate and Identity Certificate

Extensions: This field allows the addition of new fields to the AC. The extensions defined for ACs provide methods for associating additional attributes with holders. This profile also allows communities to define private extensions to carry information unique to those communities (Farrell, 2002). We chose the content of Extensions filed to carry information such as:

a) Attributed IP address.
b) MAC address (optional)
c) Configuration parameters attributed by the CACAD server (optional).
d) Authorized services.

Next, we shall present our solution based on the use of attribute certificates introduced above.

6 A NEW DHCP AUTHENTICATION OPTION

Because of the inherent vulnerabilities of current authentication mechanisms, it proves to be necessary to find solutions answering effectively this legitimate security preoccupation.
We propose a new DHCP option based on certificate concept that guaranties DHCP client and server authentication, insuring an improved access control to a DHCP system.

The following sections start with an overview of CACAD *(Certificate-based Access Control & Authentication for DHCP)* solution, then a presentation of its architecture and proposed authentication option's structure. We present a scenario demonstrating the use of CACAD for obtaining an IP address and the set of configuration parameters, then a scenario to show the way our method can be used to access resources and services within a network. We go on by presenting the advantages of our solution and finally, we illustrate CACAD implementation.

6.1 CACAD overview

CACAD defines a new DHCP option (Droms, 1999) that provides simultaneously the authentication of entities (DHCP client/server) and DHCP messages.
The technique used by this option is based on the use of public key cryptography, X.509 identity certificates (ITU-T, 1997) and attribute certificates (ITU-T, 2000). The definition of new DHCP options is possible because the options field envisages the implementation of new options (Droms, 1997).
AA *(Attribute Authority)* server functionalities of a PMI *'Privilege Management Infrastructure'* (ITU-T, 2000) are then attributed to the CACAD server (fig.3).

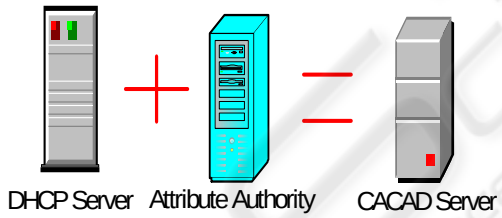

Figure 3: CACAD Server

The AA server creates the client attribute certificate that will be used in access control. The attribute certificate ensures the link between client's identity certificate and the allocated IP address. Therefore, the use of AC confirms client's ownership of the allocated IP address.

DHCP client and server must hold a valid X.509 identity certificate delivered by a trusted Certification Authority *(CA)*, both client and server must be able to validate the certificate of each other.

6.2 CACAD Architecture

Figure 4 depicts the four main components of our solution: 1) The ***Client***, is an equipment using DHCP to obtain configuration parameters. 2) The ***Server***, is an equipment that returns configuration parameters and IP address to clients, and creates a client AC, which contains the affected IP address. 3) ***X.509 Identity Certificate Database***, is a Database where entities' (client or server) X.509 identity certificates are saved. 4) ***Attribute Certificate Database***, is a Database where clients' ACs are saved.

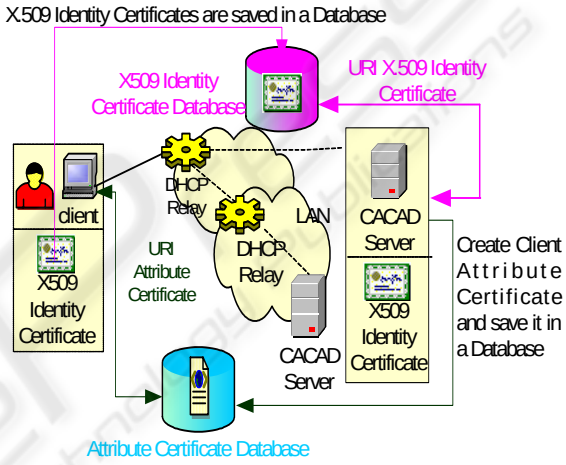

Figure 4: CACAD Architecture

Details about the interaction between these elements will be presented in 6.4 and 6.5.

6.2 Authentication option structure

This sub-section presents the proposed option structure (fig.5).

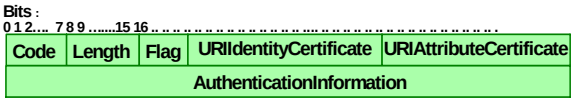

Figure 5: Authentication option structure

Code: Indicates the option code (211).
Length: Indicates the entire option length.
Flag (0 or 1): Indicates if the entity (client or server) used the (server or client) public key to encrypt the content of the field "*AuthenticationInformation*" (if key used, flag=1, otherwise, flag=0. Default value is 1). This field makes it possible to the message receiver to know if his/her public key was used by the message sender.

URIIdentityCertificate: Defines X.509 identity certificate URI *(Uniform Resource Identifiers)* (Berners-Lee, 1998) of the message sender (client or server).
URIAttributeCertificate: Defines client attribute certificate URI. This certificate is created by the CACAD Server.
AuthenticationInformation: Contains the signature value if Flag=0. The signature is applied to the whole DHCP message including the header and the options except 'hops' and 'giaddr'. This signature is created using the message sender's private key. The sender may then encrypt this signature using the receiver public key, and put the resulting value in the *AuthenticationInformation* field, which means Flag=1.
This double action signature/encryption requires the client or the server to be in possession of respectively the server's or client's public key.

6.4 CACAD Scenario

CACAD acts the same way as DHCP Delayed Authentication. That is, the client and server send authentication information in an option within each DHCP packet (Demerjian, 2004) and the DHCP protocol itself remains unchanged.
The client broadcasts a *DHCPDiscover* message on its local physical subnet. This message includes the proposed authentication option.
The client specifies its identity certificate URI in *DHCPDiscover* message, then in response, the server specifies its identity certificate URI in *DHCPOffer* message.
In all the transactions (fig.6), the sender (client/server) encapsulates the value of the encrypted signature of DHCP message on one hand. And on the other hand, the corresponding receiver (server/client) checks signature's authenticity.
Information included in X.509 identity certificates will be used by the client and the server in signature validation for the rest of the transaction.
When the server receives the *DHCPRequest* message, it will create the client's attribute certificate and save it in a database.

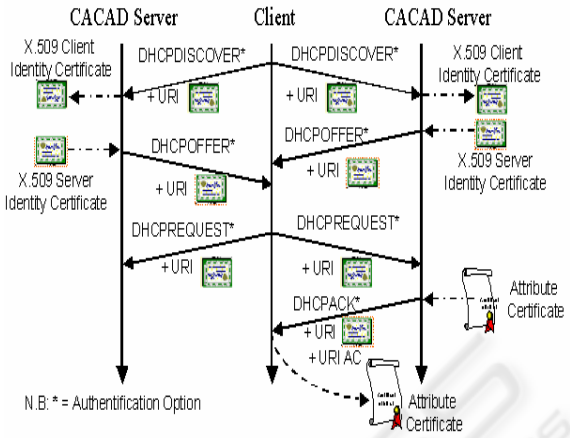

Figure 6: CACAD Scenario

The server specifies the attribute certificate URI in the *DHCPACK* message. This URI is used by the client to extract its attribute certificate from the database. The use of digital signatures provides authenticity and integrity of transmitted data, and the use of encryption guarantees confidence into sensitive data.

6.5 Service access scenario

CACAD was proposed in order to allow a strict control on equipments by using a strong authentication. The final objective is to allocate to the equipment an attribute certificate containing the Internet address dynamically allocated. This certificate ensures the link between the client identity certificate and the allocated IP address. This attribute certificate will be then used in access control. For their (equipments) authentication within network architectures, equipments can prove of their address by presenting their identity certificate and attribute certificate.
As soon as the client receives his/her IP address and attributes certificate, it becomes possible to reach the offered services beyond the access control server. A scenario of access control is illustrated on figure 7.
The steps to be followed are:
1. The client uses the IP address attributed by the

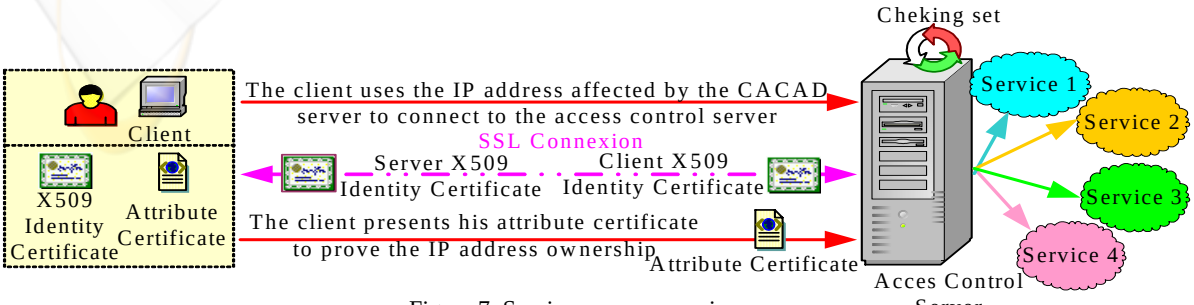

Figure 7: Service access scenario

CACAD Server to establish a connection with the access control server.
2. The client and the access control server use *'SSL client authentication'* and *'SSL server authentication'* (Freier, 1996) which allow:
a) A server to confirm a client identity.
b) A client to confirm a server identity.
3. The client presents his attributes certificate to the access control server.
4. The access control server verifies:
a) X.509 Identity certificate (Validity period, certification chain, etc.)
b) AC (Validity period, attributed IP address, authorized service, etc.)
c) Link validity between the X.509 identity certificate and the AC.
d) Equality between both, IP address value used by the client to connect to the server and IP address value contained in the AC.
5. If the verification in the preceding part is successful, the access control server allows the client to access the authorized service referenced in the AC Extensions field.

6.6 CACAD Advantages

In this section, we present some important advantages of our authentication solution for DHCP protocol:

1. CACAD provides simultaneously authentication of entities (client/server) and authentication of DHCP messages.
2. It uses RSA digital signature mechanism, which provides a better security than symmetric encryption. The use of this mechanism eliminates key distribution and key flexibility problems existing in the use of shared keys.
3. It allows a strict control over equipments by using a strong authentication (using X.509 Identity and Attributes Certificates).
4. *DHCPDiscover* messages are authenticated by this protocol, which makes the protocol invulnerable to denial of service attack through flooding with unauthenticated *DHCPDiscover* messages.
5. Is invulnerable to messages interception.
6. It supports inter-domain authentication.
7. The use of AC confirms the client IP address ownership.
8. CACAD is an open solution that can be generalized for use with other relevant problems such as DHCP-IPSec and DHCP-NAT (*Network Address Translation*).

And finally, our solution avoids changing current DHCP protocol.

6.7 Implementation

We re-used DHCP code base proposed by the Internet Software Consortium (ISC, 2004) under GPL license to implement the proposed solution in DHCP client and server.

ISC DHCP code base is an implementation of the DHCP protocol which comprises several components (a DHCP client, a DHCP server and support for DHCP relays).
We however largely modified the ISC DHCP sources, in order to add the suggested option. All exchanged messages from now on are signed by the client and the server. We also developed an attribute authority, to which the DHCP server is leaned.
We chose to set the Code option value to 211, which is still not used yet.
The structure of the proposed option was added in dhcpd.h (This file contains essential structures and functions declarations) in order to store information relating to this option on the server.

7 CONCLUSION AND FUTURE WORK

In this paper, we presented a new DHCP option based on the use of certificates. This option provides authentication of entities (client, server) and DHCP messages on one hand. And on the other hand, it allows an improved access control to the DHCP system by using attribute certificates.
In our proposal, DHCP server is leaned on an Attribute Authority server that creates a client Attribute Certificate, which ensures the link between the client identity certificate and the allocated IP address.
We have implemented CACAD by modifying the open source and free DHCP code base, developed by the Internet Software Consortium (ISC, 2004). We point out that the keys management protocol ISAKMP (Maughan, 1998) supports the attributes certificates. This is why we believe that CACAD solution perfectly articulates and interoperates with IPsec (Kent, 1998) protocol using the certificates. A future direction of our research is to validate the interoperability of our proposition with IPsec and NAT through additional developments and an establishment of real scale tests.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewer Mr. Salim Ferraz who has produced detailed reviews and much helped to produce this paper.

REFERENCES

Berners-Lee, T. Fielding, R. & Masinter, L. (1998). *Uniform Resource Identifiers (URI): Generic Syntax*, IETF, RFC 2396.

Croft, B. & Gilmore, J. (1985). *BOOTSTRAP PROTOCOL (BOOTP)*, IETF, RFC 951.

DelRey, M (1981). *INTERNET PROTOCOL*, IETF, RFC 791.

Demejian, J., Tastet, F., & Serhrouchni, A. (2003). Why certificates don't meet e-business needs?. In SSGRR'03W, International Conference on Advances in infrastructure for e-Electronic, e-Business, e-Education, e-Science, e-Medicine on the Internet. SSGRR Conference, 2003, pp 58.

Demerjian, J. & Serhrouchni, A. (2004). DHCP authentication using certificates. In SEC'04, 19th IFIP International Information Security Conference. SEC Conference, 2004.

Droms, R. & Arbaugh, W. (2001). Authentication for DHCP Messages, IETF, RFC 3118.

Droms, R. (1997a). *Dynamic Host Configuration Protocol*, IETF, RFC 2131.

Droms, R. & Alexander, S. (1997b). *DHCP Options and BOOTP Vendor Extensions*, IETF, RFC 2132.

Droms, R. (1999). *Procedure for Defining New DHCP Options*, IETF, RFC 2489.

Farrell, S. & Housley (2002), R., *An Internet Attribute Certificate Profile for Authorization*, IETF, RFC 3281.

Freier, A. and al., 1996. *The SSL Protocol, Version 3.0*, Netscape Communications Corp. Standards Information Base, The Open Group.

Glazer, G., Hussey, C & Shea, R. (2003). Certificate-Based Authentication for DHCP [Electronic version]. Retrieved March 20, 2003, from UCLA university, Computer Science Department Web site: http://www.cs.ucla.edu/~chussey/proj/dhcp_cert/cbda.pdf

Hornstein and al., 2000. *DHCP Authentication via Kerberos V*, IETF, Internet Draft.

ISC: Internet Software Consortium. Dynamic Host Configuration Protocol Distribution. Retrieved February 06, 2004, from http://www.isc.org/index.pl?/sw/dhcp/

ITU-T Recommendation X.509, 1997. Information technology-Open Systems Interconnection-The Directory: Authentication framework.

ITU-T Recommendation X.509, 2000. Information technology-Open Systems Interconnection-The Directory: Public-key and attribute certificate frameworks.

Jonsson, J. & Kaliski, B. (2003). Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, IETF, RFC 3447.

Kent, S. & Atkinson, R. (1998). *Security Architecture for the Internet Protocol*, IETF, RFC 2401.

Komori, T., & Saito, T. (2002). The secure DHCP System with User Authentication. In LCN'02, 27th Annual IEEE Conference on Local Computer Networks. LCN Conference, 2002, pp 0123.

Lemon, T. & S. Cheshire (2002), *Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)*, IETF, RFC 3396.

Maughan, D. Schertler, M., Schneider, M. & Turner, J. (1998). *Internet Security Association and Key Management Protocol (ISAKMP)*, IETF, RFC 2408.

Perkins, C., & Luo, K. (1995). Using DHCP with computers that move. In Wireless Networks, 1995, Volume 1, No. III, pp 341-354.

Postel, J. (1980). *User Datagram Protocol*, IETF, RFC 768.

Tominaga, A., Nakamura. O., Teraoka. F., & Murai. J. (1995). Problems and solutions of DHCP. In INET'95, The 5th Annual Conference of the Internet Society. INET Conference, 1995.