DHCP AUTHENTICATION USING CERTIFICATES

Jacques Demerjian and Ahmed Serhrouchni
Ecole Nationale Supérieure des Télécommunications – LTCI-UMR 5141 CNRS, France
{demerjia, ahmed}@enst.fr

Abstract: In this paper, we describe several methods of DHCP authentication. We propose an extension to DHCP protocol in order to allow a strict control on equipments by using a strong authentication. This extension, called E-DHCP *(Extended-Dynamic Host Configuration Protocol)* is based on two principles. The first one is the definition of a new DHCP option that provides simultaneously the authentication of entities (client/server) and DHCP messages. The technique used by this option is based mainly on the use of asymmetric keys encryption RSA, X.509 identity certificates and attribute certificates. The second principle is the attribution of PMI (*Privilege Management Infrastructure*) attribute authority server functionalities to DHCP server. This server creates an attribute certificate to the client, which ensures the relation between the identity certificate of the client and the allocated IP address. This attribute certificate will be then used in the access control.

Key words: Access Control, Attribute Certificate, Authentication, DHCP, X.509 Identity Certificate.

1. INTRODUCTION

A protocol of dynamic attribution of Internet addresses is necessary for the functioning of a considerable number of networks and this, for two reasons. The first is the lack of Internet addresses, which does not allow static attribution of addresses. The second is that the mobility of the equipment is adapted to dynamic addressing. DHCP *(Dynamic Host Protocol Configuration)* [1], is thus in the centre of networks architectures, this protocol provides a framework for passing configuration information to hosts on a TCP/IP network. DHCP is based on the *Bootstrap Protocol*

Y. Deswarte et al. (eds.), *Security and Protection in Information Processing Systems*

'BOOTP' [3], adding the capability of automatic allocation of reusable network addresses and additional configuration options.

DHCP is built directly on UDP [12] and IP [7], which are as yet inherently insecure. 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 [6]. Unauthorized DHCP servers may be easily set up. Such servers can then send false and potentially disruptive information to clients such as incorrect or duplicate IP addresses, incorrect routing information, incorrect domain name server addresses, and so on. Clearly, once this seed information is in place, an attacker can further compromise affected systems. Malicious DHCP clients could masquerade as legitimate clients and retrieve information intended for those legitimate clients. Where dynamic allocation of resources is used, a malicious client could claim all resources for itself, thereby denying resources to legitimate clients. Therefore, DHCP currently provides no authentication or security mechanisms.

For all of these problems, a solution would be to authenticate both the client and the server. Many different contributions [8][13] regarding how DHCP should be authenticated already exist. Some contributions include cryptography, some do not.

In this paper, we propose an extension [2] to DHCP protocol in order to allow a strict control on the equipments by strong authentication. This extension ensures on one hand, the authentication of the entities and DHCP messages and, on the other hand, the access control in DHCP system.

The remainder of the paper is structured as follows: Section 2 introduces the DHCP basic operations; section 3 presents the DHCP importance; section 4 explores the DHCP shortcomings and vulnerabilities. Section 5 presents some existing contributions that define how authentication should be handled in DHCP, and exposes their limits. Section 6 illustrates our proposed extension, and finally section 7 concludes the paper and identifies possible future work.

2. BASIC DHCP OPERATIONS

The DHCP provides a way to automate and manage the network configuration of desktop computers and other network devices that use the TCP/IP protocol. DHCP, like *BOOTP* uses client-server model and is set on UDP, utilizing ports 67 and 68. It uses the same packet format as *BOOTP* for its compatibility [5]. DHCP relays messages between the client and the server. In DHCP a client initiates all interactions, and a server replies.

By using DHCP, dynamically configuring the host on the network is done by a simple handshake. The process to be followed to get configuration data from DHCP server can be divided into two phases.

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* [21].

3. IMPORTANCE OF DHCP

The introduction of DHCP alleviated the problems associated with manually assigning TCP/IP client addresses. Network administrators have quickly appreciated the importance, flexibility and ease-of-use offered in DHCP. DHCP has several major advantages over manual configurations. No manual reconfiguration is required at all. DHCP reduces the amount of work required to administer large IP networks by eliminating the need to individually assign, configure, and manage a permanent IP address for every machine. Each computer gets its configuration from a *"pool"* of available numbers automatically for a specific time period. When a computer has finished with the address, it is released for another computer to use. Configuration information can be administered from a single point. Major network resource changes (e.g. a router changing address), requires only the DHCP server be updated with the new information, rather than every system.

4. DHCP SECURITY

In this section, we illustrate major DHCP shortcomings and vulnerabilities.

4.1 DHCP shortcomings

DHCP suffers from some significant shortcomings, which include [22]:

1. The lack of robust administrative tools: DHCP lacks such administrative capabilities as the ability to associate an address with a user name.
2. The lack of intelligence: The DHCP server is simply a server, distributing IP addresses as requested and collecting them when they expire according to a set of simple rules. It doesn't ask questions about the user or track information other than the IP address and lease parameters. It cannot perform more sophisticated tasks to effectively manage the IP address asset.
3. Limited security: DHCP servers provide only limited facilities, such as log files, to audit IP address distribution or maintain a record of IP address usage. After manually combing the file, the administrator comes up with a MAC address, which still doesn't identify the machine to which the address has been assigned. Yet, such identification is critical when trying to track network activity at a given time. As a result, DHCP provides no effective way to identify conflicts or track down rogue addresses without manually examining reams of log files.

It is clear that DHCP was never intended as a full IP address management solution. Rather, DHCP was designed to perform one critical task very well: the automatic assignment of temporary IP addresses and maintenance of an IP address pool.

4.2 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, any intruder can effectively impersonate the identity of any client that divulges its identification information [6]. Likewise, an intruder can impersonate a DHCP server, and send erroneous information to any local DHCP client. When connecting to the network with the DHCP service, even an illegitimate user without the right to use the network can obtain an IP address from the DHCP server.

DHCP itself does not have an access control, illegitimate users inside of a network segment can easily abuse inside or outside the network. To solve

this problem, introduction of a MAC 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, the 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 [19]. 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 entity (client/server DHCP) authentication and content authentication of DHCP messages.

5. EXISTING CONTRIBUTIONS

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

1. **DHCP Authentication via Kerberos V** [13]: This authentication method authenticates the client only, and involves communication with the Kerberos server, in addition to the DHCP standard communication.
2. **Token Authentication** [8]: 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** [8]: 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 *(Message Authentication Code)* in the DHCP message. The authenticity of DHCP messages is confirmed using both an index to that key and a hash of the packet using that key. 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. For this reason, we shall introduce its related issues.

5.1 Delayed Authentication issues

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 protocol relies on shared cryptographic keys being already known by the client and the server.

The first issue is one of the major drawbacks to the use of shared keys [18]. 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 [9], would provide a better security.

The delayed authentication option is exposed to additional drawbacks:

1. Delayed Authentication is vulnerable to a denial of service attack through flooding with *DHCPDiscover* messages, which are not authenticated by this protocol. Such an attack may overwhelm the computer on which the DHCP server is running and may exhaust the addresses available for assignment by the DHCP server.
2. Delayed authentication does not support interdomain authentication.
3. Delayed authentication 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.

6. E-DHCP

Because of the inherent vulnerabilities of the current authentication mechanisms, it proves to be necessary to find solutions answering effectively this legitimate security preoccupation.

We propose an extension to DHCP protocol called E-DHCP *(Extended-Dynamic Host Configuration Protocol)* in order to stricter control of the equipment user through a stronger authentication process. This extension ensures on the one hand, the authentication of the entities and DHCP messages and, on the other hand, the access control to a DHCP system.

The following sections provide an overview of E-DHCP, and a scenario demonstrating the use of E-DHCP 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. Finally in this section, we present E-DHCP advantages.

6.1 E-DHCP Overview

The E-DHCP solution is based mainly on the certificate concept; so it is useful to briefly define some certificate related concepts before going further.

6.1.1 Essential background and concepts

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 [23].

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 RFC 3281, 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 attribute certificate to an X.509 PKC. 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. This field is a *SEQUENCE* allowing three different syntaxes: *baseCertID*, *EntityName* and *objectDigestInfo*. Only one option should be present. For any environment where the AC is passed in an authenticated message or session and where the authentication is based on the use of X.509 PKC, the *holder* field should use the *baseCertificateID*. With the *baseCertificateID* option, 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 the 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 the case where the same name was reassigned to different authorities through time. This field is optional.
- ***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.

6.1.2 E-DHCP Principles

E-DHCP is based on two principles:

The first is the definition of a new DHCP option [17] "*E-DHCP Authentication*" that provides simultaneously the authentication of entities and DHCP messages. The technique used by this option is based mainly on the use of *asymmetric* keys encryption RSA, X.509 identity certificates [10] and attribute certificates [11]. The definition of new DHCP options is possible because the options field envisages the implementation of new options [2].

The second is the attribution of the AA *(Attribute Authority)* server functionalities of a PMI *(Privilege Management Infrastructure)* [11] to the DHCP server. This server creates the client attribute certificate 'AC'. This certificate ensures the relation between the client identity certificate and the allocated IP address. This AC will be then used in the access control.

6.1.3 E-DHCP Requirements

1. The client must hold a valid X509 identity certificate delivered by a trusted CA (*Certification Authority*).
2. The server must hold a valid X509 identity certificate delivered by a trusted CA (*Certification Authority*).

6.1.4 E-DHCP Architecture

Figure 1 depicts the different components of E-DHCP architecture which are described as follows:

1. ***E-DHCP Client***: An 'E-DHCP Client' or 'client' is an equipment using DHCP to obtain configuration parameters.
2. ***E-DHCP Server***: An 'E-DHCP Server' or 'server' is a server that:

a) Returns configuration parameters and IP address to E-DHCP clients.
b) Creates a client AC, which contains the IP address allocated.

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.

Details about the interaction between these elements will be presented in 6.2 and 6.3.

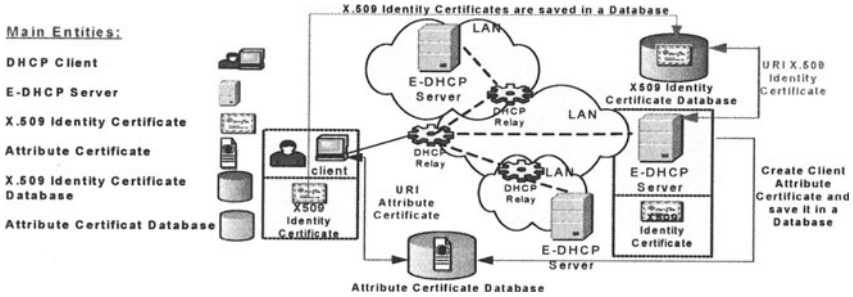

Figure 1. E-DHCP Architecture

6.1.5 E-DHCP Authentication option structure

This sub-section presents the "*E-DHCP Authentication*" option structure (see Fig. 2), which constitutes the first E-DHCP principle.

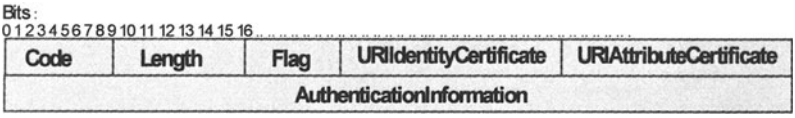

Figure 2. E-DHCP Authentication option structure

- ***Code***: Indicates the option code (To Be Determined).
- ***Length***: Indicates the entire option length.
- ***Flag*** (0 or 1): Indicates if the client used the server public key or alternatively if the server used the 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 allows the message receiver to know if their public key was used by the message sender.
- ***URIIdentityCertificate***: Defines the X.509 identity certificate URI *(Uniform Resource Identifiers)* [4] of the message sender (client or server).
- ***URIAttributeCertificate***: Defines the client attribute certificate URI. This certificate is created by the E-DHCP 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 requiring the client to be in possession of the server public key, would avoid a denial of service attack through flooding with non-authenticated *DHCPDiscover* messages. The same technique is used by the server to sign the message and encrypt its signature if the server is in possession of the client public key.

6.2 E-DHCP Scenario

The E-DHCP works the same way as DHCP Delayed Authentication. That is, the client and server send authentication information in an option within each DHCP packet and the DHCP protocol itself remains unchanged.

The following steps present a scenario of successful DHCP exchanges between clients and servers:

1. The client broadcasts a *DHCPDiscover* message (see Fig.3) on its local physical subnet. This message may include options, and among them, those which suggest values for the network address and lease duration or, that which provides both entity and message authentication "*E-DHCP Authentication*"option, etc.

Figure 3. E-DHCP Authentication option structure in a DHCPDiscover message

To authenticate themselves (the client), and to confirm their identity to the server, the client includes the "*E-DHCP Authentication*" option in the *DHCPDiscover* message. Before this step, the client must specify in the corresponding "*E-DHCP Authentication*" option fields:

- '***URIIdentityCertificate***': Defines the URI (Uniform Resource Identifiers) of his X.509 identity certificate.
- '***URIAttributeCertificate***': Puts the value 0 in this field.
- '***Flag***': Indicates if they used the E-DHCP Server public key to encrypt the content of the field "*AuthenticationInformation*" (if key used, Flag=1, otherwise, Flag=0).
- '***AuthenticationInformation***': Contains the signature value if Flag=0. The signature is applied to the whole *DHCPDiscover* message including the header and the options except 'hops' and 'giaddr'. This signature is created using the client private key. The client may then encrypt this

signature using the server public key and put the resulting value in the *AuthenticationInformation* field, which means Flag=1.

N.B: The use of Digital signatures provides authenticity and integrity of transmitted data, and the use of encryption guarantees confidence in sensitive data.

2. To validate the authentication of the client and the incoming message '*DHCPDiscover*', the E-DHCP Server:

a) Uses the URI of the client X.509 identity certificate, contained in the '*URIIdentityCertificate*' field of *DHCPDiscover* message to extract the client X.509 identity certificate, and then to extract the client public key from this identity certificate.

b) Verifies the value contained in the 'Flag' field:

- If the field value is equal to 0, the server uses the client public key (extracted from the client X.509 identity certificate) in the verification of the validity of the signature (contained in the '*AuthenticationInformation*' field).
- If the field value is equal to 1, the server uses its private key to decrypt the value contained in the '*AuthenticationInformation*' field. The result of this decryption is the 'signature'. The server uses the client public key (extracted from the client X.509 identity certificate) in the verification of this signature.

If the authentication of the message and the client is validated, the server prepares an offer to send to the client. Otherwise, the server discards the message.

c) May choose to accept unauthenticated *DHCPDiscover* messages, or only accept authenticated *DHCPDiscover* messages based on its policy.

d) Responds with a *DHCPOffer* message (see Fig. 4) that may include "*E-DHCP Authentication*" option.

To authenticate itself (server), and to confirm its identity to the client, the server includes the "*E-DHCP Authentication*" option in the *DHCPOffer* message. Before this step, the server must specify in the corresponding "*E-DHCP Authentication*" option fields:

- '***URIIdentityCertificate***': Defines the URI *(Uniform Resource Identifiers)* of the server X.509 identity certificate.
- '***Flag***': Indicates if the sever used the client public key to encrypt the content of the field "*AuthenticationInformation*" (if key used, Flag=1, otherwise, Flag=0).
- '***URIAttributeCertificate***': Puts the value 0 in this field.
- '***AuthenticationInformation***': Contains the signature value if Flag=0. The signature is applied to the whole *DHCPOffer* message including the header and the options except 'hops' and 'giaddr'. This signature is created using the server private key. The server may then encrypt this

signature using the client public key and put the resulting value in the *AuthenticationInformation* field, which means Flag=1.

Figure 4. E-DHCP Authentication option structure in a DHCPOffer message

3. To validate the authentication of the server and the incoming message 'DHCPOffer', the client:

a) Uses the URI of the server X.509 identity certificate, contained in the '*URIIdentityCertificate*' field of *DHCPOffer* message to extract the server X.509 identity certificate, and then to extract the server public key from this identity certificate.

b) Verifies the value contained in the 'Flag' field:

- If the field value is equal to 0, the client uses the server public key (extracted from the server X.509 identity certificate) in the verification of the validity of the signature (contained in the '*AuthenticationInformation*' field).
- If the field value is equal to 1, the client uses its private key to decrypt the value contained in the '*AuthenticationInformation*' field. The result of this decryption is the 'signature'. The client uses the server public key (extracted from the server X.509 identity certificate) in the verification of this signature.

If the message or the server fail to pass the authentication validation or, if the offer suggested by the E-DHCP Server is unacceptable to the client; the latter discards the message, and sends a new *DHCPDiscover* message.

Otherwise, if the message and the server pass successfully the authentication validation and, if the offer suggested to the E-DHCP Server is acceptable by the client, the latter sends a *DHCPRequest* message (see Fig.5) to the server (including "*E-DHCP Authentication*" option): (1) requesting offered parameters from the selected server and implicitly declining offers from all others, (2) confirming the correctness of the previously allocated address after, e.g., system reboot, or (3) extending the lease on a particular network address.

The client follows the same steps to specify the "*E-DHCP Authentication*" option field's value followed-up in the *DHCPDiscover* message detailed in 1, with only one possible difference which is to put, in the '*URIAttributeCertificate*' field, the URI attribute certificate (In this case the client sends the *DHCPRequest* message to extend the lease on a particular network address).

Figure 5. E-DHCP Authentication option structure in a DHCPRequest, DHCPRelease or DHCPDecline message

4. In the same way (as described in 2.a and 2.b) the server validates the authentication of the client and the *DHCPRequest* message.

If the authentication of the message and the client is validated or, if the server can't satisfy the client request, the server discards the message and sends a *DHCPNack* message to the client.

Otherwise, if the message and client pass successfully the authentication validation, the server verifies the value contained in the *'URIAttributeCertificate'* field:

a) If the value is equal to 0. The server creates an AC (version 2) for the client and saves it in the 'Attributes Certificate Database'. The issuer of this AC (E-DHCP Server) specifies the allocated IP address associated with the DHCP client in the AC *'Attributes'* field. In addition, the server can (optionally) specify in the *'Extensions'* field: (1) The MAC address of the equipment or, (2) The configuration parameters affected by the E-DHCP Server. The *'Holder'* field identifies the DHCP client that possesses the IP address. The validity period of this AC will correspond to the lease validity period. As from the moment when the validity period of the lease expires, the client can no longer exceed the access control server.

b) If the value is equal to an URI (Which means that the client has already an AC) then the server uses this URI to extract the AC from the Database. The server checks the certificate validity. If it is not expired, he can renew it. Otherwise, the server creates a new client attribute certificate and saves it in the 'Attributes Certificate Database'.

E-DHCP Server sends a *DHCPAck* message (see Fig.6) to the client containing "*E-DHCP Authentication*" option.

Figure 6. E-DHCP Authentication option structure in a DHCPAck message

E-DHCP Server follows the same steps to specify the "*E-DHCP Authentication*" option field's value as in *DHCPOffer* message in 2, with only one difference that the server puts, the URI client AC (renewed or created) in the *'URIAttributeCertificate'* field.

5. In the same way (as described in 3.a and 3.b), the client validates the authentication of the server and the message *DHCPAck*.

If the authentication of the message and the server is validated, the client uses the configuration parameters in the lease, the IP address that is affected to him/her, then extracts (following the given URI), and uses its attributes certificate containing the Internet address allotted dynamically.

6.3 Service access scenario

E-DHCP was proposed in order to allow a strict control on equipment by using a strong authentication. The final objective is to allocate to the equipment an AC containing the Internet address dynamically allocated. This certificate ensures the relation between the client identity certificate and the allocated IP address. This AC will be then used in the access control. For their (equipment) authentication within network architectures, the equipment can prove its address by presenting its identification certificate and the AC.

As soon as the client receives their IP address and attributes certificate, it becomes possible to reach the services offered beyond the access control server. A scenario of access control is illustrated in Figure 7.

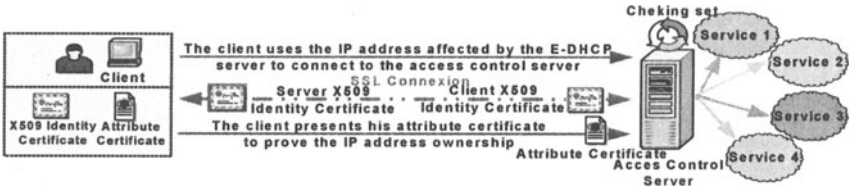

Figure 7. Scenario of access control

The steps to be followed are:

1. The client uses the IP address allocated by the E-DHCP 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' [14] 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) Identity certificate (Validity period, certification chain, etc.)
b) AC (Validity period, allocated IP, authorized service, etc.)
c) Validity of link between the X509 identity certificate and the AC.
d) The link between the identity of the client and the IP address.

5. If the verification in the preceding part is successful, the access control server allows the client to be connected to the authorized service.

6.4 E-DHCP advantages

These are the E-DHCP advantages, listed in order of importance:

1. E-DHCP avoids changing the current DHCP protocol.
2. It provides simultaneously the authentication of entities (client/server) and the authentication of DHCP messages. The technique used is based mainly on the use of asymmetric keys encryption, X.509 identity certificates and attribute certificates.
3. It uses the 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.
4. It allows a strict control over the equipment by using a strong authentication (using X.509 Identity and Attributes Certificates 'AC').
5. *DHCPDiscover* messages are authenticated by this protocol, which makes the protocol invulnerable to denial of service attack through flooding with unauthenticated *DHCPDiscover* messages.
6. Is invulnerable to message interception.
7. It supports inter-domain authentication.
8. The use of AC confirms the client IP address ownership.

7. CONCLUSION AND FUTURE WORK

This paper has presented an extension to DHCP protocol. This extension, called E-DHCP *(Extended-Dynamic Host Configuration Protocol)*, uses asymmetric keys encryption RSA mechanism, X.509 identity certificates and, attribute certificates 'AC' to provide simultaneously the authentication of entities (client/server) and the authentication of DHCP messages. In E-DHCP, DHCP server asks on an Attribute Authority server to create a client AC, which ensures the relation between the client identity certificate and the allocated IP address. This AC is used in the access control.

We have implemented E-DHCP by modifying the open source and free DHCP code base, developed by the Internet Software Consortium [20], then by the development of an attribute authority, to which the DHCP server is attached. We point out that the keys management protocol ISAKMP [15] supports the attributes certificates. This is why we believe that E-DHCP perfectly articulates and interoperates with IPSec [16] protocol using the certificates. A future direction of our research is to validate the interoperability of our proposition with IPsec through real scale developments and tests.

8. ACKNOWLEDGEMENTS

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

REFERENCES

1. R. Droms "Dynamic Host Configuration Protocol", IETF, RFC 2131, Mar. 1997.
2. R. Droms and S. Alexander, "DHCP Options and BOOTP Vendor Extensions", IETF, RFC 2132, Mar. 1997.
3. B. Croft and J. Gilmore, "BOOTSTRAP PROTOCOL (BOOTP) ", IETF, RFC 951, Sep. 1985.
4. T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", IETF, RFC 2396, Aug. 1998.
5. R. Droms, "Interoperation Between DHCP and BOOTP", IETF, RFC 1534, Oct. 1993.
6. C. Perkins and K. Luo, "Using DHCP with computers that move", Wireless Networks, Mar. 1995.
7. M. del Rey, "INTERNET PROTOCOL", IETF, RFC 791, Sep. 1981.
8. R. Droms and W. Arbaugh, "Authentication for DHCP Messages", IETF, RFC 3118, Jun 2001.
9. J. Jonsson and B. Kaliski,"Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", IETF, RFC 3447, Feb. 2003.
10. Information technology-Open Systems Interconnection-The Directory: Authentication framework, ITU-T Recommendation X.509, 1997.
11. Information technology-Open Systems Interconnection-The Directory: "Public-key and attribute certificate frameworks", ITU-T Recommendation X.509, 2000.
12. J. Postel, "User Datagram Protocol", IETF, RFC 768, Aout 1980.
13. Hornstein and al., "DHCP Authentication via Kerberos V", Internet Draft, Nov. 2000.
14. A. Freier, P. Karlton and P. Kocher, "The SSL Protocol, Version 3.0", Netscape Communications Corp., November 1996. Standards Information Base, The Open Group.
15. D. Maughan, M. Schertler, M. Schneider and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP) ", IETF, RFC 2408, Nov. 1998.
16. S. Kent and R. Atkinson,"Security Architecture for the Internet Protocol", IETF, RFC 2401, Nov. 1998.
17. R. Droms, Procedure for Defining New DHCP Options, IETF, RFC 2489, Jan. 1999.
18. G. Glazer and al., "Certificate-Based Authentication for DHCP", Mar. 2003.
19. T. Komori and T. Saito, "The secure DHCP System with User Authentucation", LCN'02, 27th Annual IEEE Conference on Local Computer Networks, Nov. 2002.
20. Internet Software Consortium. Dynamic Host Configuration Protocol Distribution.
21. A. Tominaga, O. Nakamura, F. Teraoka and J. Marai, "Problems and solutions of DHCP", INET'95, The 5th Annual Conference of the Internet Society, Apr. 1995.
22. Threshold Networks, "RAZZO IP server appliance – Integrated solution for management of IP, DNS and DHCP", White paper, Apr. 2001.
23. J. Demejian, A. Serhrouchni and F. Tastet, "Why certificates don't meet e-business needs?", SSGRR'03W, International Conference on Advances in infrastructure for e-Electronic, e-Business, e-Education, e-Science, e-Medicine on the Internet, Jan. 2003.