International Journal of Security and Its Applications
Vol. 10, No. 6 (2016) pp.331-340
http://dx.doi.org/10.14257/ijsia.2016.10.6.31

Cryptographic Encapsulation in the New ENC-DNSSEC Protocol

Kaouthar Chetioui, Ghizlane Orhanou and Said El Hajji

Laboratory of Mathematics, Computing and Applications; Faculty of Sciences; Mohammed V University in Rabat, Morocco.
kaoutharchetioui@gmail.com, orhanou@fsr.ac.ma, elhajji@fsr.ac.ma

Abstract

The Domain Name System (DNS) is a fundamental component of the critical infrastructure of the Internet. The need of the DNS service is vital, as it is used in every Internet transaction. It always becomes an object of different types of attacks like cache poisoning and Man in The Middle attacks. DNSSEC, the extended version of DNS, is based on asymmetric cryptography in order to provide security services. It improves security level by adding authenticity and integrity services to DNS protocol. Unfortunately, it doesn't provide confidentiality service and consequently, data transmitted across a network is readable to the public. In this paper, we propose a new protocol ENC-DNSSEC to enhance DNSSEC security. ENC-DNSSEC proposes a new structure of DNS message which encapsulates the standard version; this cryptographic encapsulation associates two types of encryption protocols and a signature to give a high level of security in data transaction. First, we give an overview of DNS and DNSSEC protocols; particularly, we present security limitations for both of them. After that, we explain our proposal and we analyze the impact of our method to enhance DNSSEC security.

Keywords*: DNSSEC; Encapsulation; Signature; Encryption; Confidentiality*

1. Introduction

The domain name system (DNS) takes an integral part of the internet. It provides an important information to Internet operations, such as translation host names to IP addresses and vice versa. Unfortunately, this service knows a lot of types of attacks that inject or modify DNS messages. These modifications can be propagated across the network.

Considering the importance of DNS service, it is necessary to look after new techniques and solutions in order to secure communication within the DNS system.

Securing DNS means ensuring data authentication, integrity protection and confidentiality. DNSSEC, the extended version of DNS uses public-key cryptography to provide data integrity and authentication of source data. Also, TSIG (Transaction Signatures) based on symmetric techniques and public-key signature authenticates and ensures data integrity in transaction between local servers. But confidentiality service is not provided by the existing systems, so, data stored in DNS database and also data in transit are still visible to the public.

In this paper, we describe a new protocol ENC-DNSSEC which is based on cryptographic encapsulation in DNSSEC protocol to protect confidentiality of data in transit. We present first the DNSSEC system. Then, we will explain the limitation of security in this service and the necessity of a new method to enhance the level of security in DNS exchanges. Finally, we give an analysis of the efficiency of our method by comparing it with other existing methods especially with the one using IPsec to secure DNS exchange.

ISSN: 1738-9976 IJSIA

2. DNS and DNSSEC Message Structures

2.1. DNS Message Presentation

As already known, DNS is a distributed database, whose principal activity is translating hostnames to IP addresses and vice versa. This operation is completed when the server answers queries coming from the client [8].

When a DNS client enters a domain name 'www.example.ma' in a web URL, the domain name considered as a query is transformed by the system on DNS request readable by the resolver. This transformation depends on the local OS and data structure on the system. The format of a standard query is presented in "Figure 1".

Header	OPCODE=SQUERY
Question	QNAME=example.ma., QCLASS=IN, QTYPE=A
Answer	<empty>
Authority	<empty>
Additional	<empty>

Figure 1. Format of Standard Query in DNS Message [9]

As shown in "Figure 1", the standard query has other fields in addition to the domain named entered by the client. These fields contain all detail information about the query. The communication in DNS follows the client / server system, so, when the server receives the query, it looks for the response in its database to have the associated IP address to the desired domain name, especially, it makes search in zone files which contain all information about domains names.

There are two types of queries: iterative and recursive queries "Figure 2".

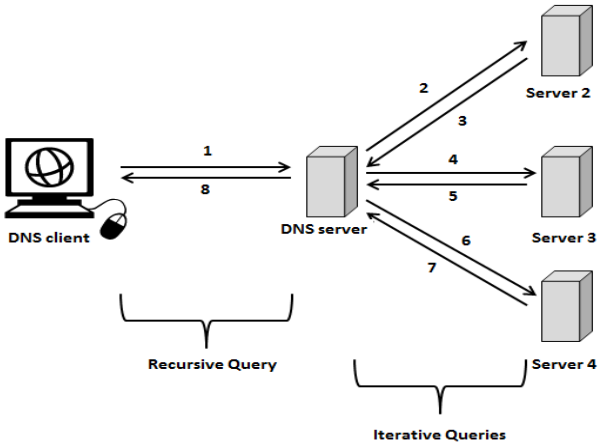

Figure 2. Schema of Iterative and Recursive Queries

In recursive query, the DNS server has to do the entire job in order to search the answer and give it back to the DNS client. In the iterative query, the DNS server doesn't give the complete answer to the client, but it gives the referral of other servers that can have the answer. The standard format of the answer is shown in the figure bellow:

Header	OPCODE=SQUERY, RESPONSE, AA
Question	QNAME=example.ma., QCLASS=IN, QTYPE=A
Answer	example.ma. 86400 IN A 26.0.0.73
Authority	<empty>
Additional	<empty>

Figure 3. Format of Standard Response in DNS Message [9]

Referring to "Figure 1" and "Figure 3", the query and the answer in a DNS request have the same structure except the answer which indicates that this message is an answer and it gives the IP address of the domain name queried by the resolver. Finally, the resolver returns the response to the client indicating the IP address of the domain name 'example.ma.', the entire process of this operation is called resolving.

Unfortunately, as the DNS becomes very important in our usual communication over the internet, it becomes also the principal objective of different types of attacks. A hacker can intercept the network, modify data or inject false information to the client.

2.2. DNSSEC Message Presentation

DNSSEC the extended version of the protocol DNS, [RFC4033-4035], is proposed and standardized in 1997. DNSSEC uses cryptography in order to protect DNS from different types of attacks like cache poisoning and Man-In-The Middle [2].

DNSSEC uses two types of keys ZSK (Zone Signing Key) and KSK (Key signing Key) to sign DNS zone. In addition to DNS resource records, DNSSEC uses new records: RR KEY, SIG, NSEC and DS [4].

a) DNSKEY RR recording: The DNSKEY resource record stores all public key pairs that are necessary for decryption information received as DNS replies.

b) RRSIG record: The RRSIG record results of the signing of RRsets made by the private key of the pair and it provides the digital signature to the provided data. So, it accompanies every RR and it's considered as the basic block of DNSSEC that is necessary to verify the authenticity of the returned data.

c) NSEC record: is used by the DNSSEC protocol when the requested name doesn't exist so, it's called proof of nonexistence and occasionally denial of existence.

d) DS record: It allows a parent zone to validate the KEY record of its child zone.

DNSSEC uses private keys to sign zone files and it generates a new record called SIG, and every resource record in DNSSEC is associated to SIG that contains the signature. So, when a hacker tries to modify any data in zone files, the signature will be false and verification will indicate that there is a change in records. This verification ensures the integrity of data in DNS protocol.

As shown in this section, DNSSEC uses its four new resource records to secure data in DNS protocol, they ensure authenticity and integrity of data but there is no measure to secure the channel, so it is still vulnerable to security attacks. For this reason, we thought to a new idea that can be added to DNSSEC in order to ensure, authenticity, integrity and confidentiality services to DNS protocol.

3. DNS and DNSSEC Security Related Works

G. Ateniese and S. Mangard [3] propose a new approach to DNS security (DNSSEC) based on symmetric-key cryptography. Their proposal is called SK-cryptography (secret key cryptography), it offers a high level of security in request exchanges. In addition to that, it reduces storage requirements and enables efficient mutual authentication.

R. Curtmola *et. al.*, [1] discuss the advantages and disadvantages of two protocols PK-DNSSEC and SK-DNSSEC to secure DNS protocol. Also they examine the impact of

these protocols on DNS performance such as the functionality and the complexity of this protocol. This impact was justified by an experimental study of dependencies within domain registrars and within domains in the DNS infrastructure and observes that multiple dependencies result in more vulnerable DNS.

The RFC 2535 [6] define PK-DNSSEC as public key DNS security extensions. PK-DNSSEC uses three Resource Records (RR):

- Key: encode public key associated with a zone
- SIG: encode digital signatures over a RR set
- NXT: indicate what doesn't exist in a zone

PK-DNSSEC is based on these three RR to ensure authenticity and data integrity. In this RFC, authors discuss the security of DNS protocol; they focus their studies on two important types of attacks: cache poisoning and MiTM attacks by giving different scenarios and situations that explain how attackers can modify information exchanged across the network via DNS protocol. In addition to that, they explain the impact of DNS security on e-commerce, economical losses to business, internet availability and connectivity. At the end, they explain the necessity of DNSSEC protocol to detect and prevent from these types of attacks.

Another method used in demand by DNS to secure the canal is IPsec. Using IPsec, when a mobile worker wants to connect to a web server, it enters the URL which contains the server domain name. Normally, once the IP address of the wanted server is entered, the local IPsec starts the process of securing the canal, when it observes that there is a new entry in the SPD 'Security Policy Database' for the destination IP address, thus it calls the IKE to establish a security association with the server. But this operation cannot be established all the time in a case of a corruption of the DNS cache information.

Indeed, a hacker who wants to attack a DNS system communication secured by IPsec can corrupt the data of a DNS server on the Internet, taking advantage of a false transfer zone, by a wrong answer or a dynamic update, *etc*. So, when the client requests the IP address of the DNS server, it sends the request to the DNS which replies by the false response injected by the attacker. The information is then back down to the customer. The IP address of the server is not listed in the SPD, therefore it does not establish IKE Security Association, and the default policy applies: DNS server will send the DNS message clearly or it'll reject the request. So, the present example explains that even with using IPsec, isn't sufficient to prevent DNS server from attacks.

In the next section, we will focus on the description of our proposal to enhance security in DNSSEC protocol.

4. The New Protocol ENC-DNSSEC

As we aim, through the use of the new protocol, to encapsulate DNS request between DNS client and DNS server, we will refer to our proposal as ENC-DNSSEC (Encapsulation of DNSSEC request).

The idea is to implement a secure communication between two DNS servers using a cryptographic encapsulation combining two types of encryption (symmetric and asymmetric) and a signature (hash function).

The principle objective is to ensure the confidentiality, to enhance the level of authentication of servers and consequently to ensure the integrity of information provided by servers. For this, the two servers are provided with a keyset (private, public) built and delivered by a third party acting as a CA (Certification Authority) and this set of key is used for the asymmetric encryption.

The notation used in this work is presented in "TABLE 1". We suppose that the domain name which we are looking for is "example.ma".

S1	local name server
S2	distant name server
K_{TS}	session key
(K_{privS1}, K_{pubS1})	private, public key for S1
(K_{privS2}, K_{pubS2})	private, public key for S2
J_t	token

Table 1: Notation

We suppose that we have two servers "S1" and "S2", "S1" is our local name server, the root is not authoritative, so when we are looking for the IP address of "example.ma" the query will refer the resolver to the second DNS server "S2".

To secure communication between servers, S1 and S2 are provided by a pair of keys (public and secret keys) built and delivered by the root server acting as a CA which is responsible for generating a delivering certificates to users "Figure 5".

Asymmetric cryptographic algorithms are based on the sharing of public key between servers and they are stored in the Kerberos system [7]. However, these keys require authentication service because any hacker can access to the directory and change the public key existing in the system by his public key. So, this service is usually achieved by a CA in order to give certificates to users.

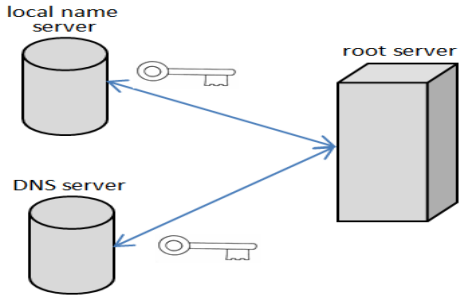

Figure 4. Certificate Authority

When a client wants to consult a website, for example 'www.example.ma', the DNS agent is contacting its local name server (local resolver), to look for the corresponding IP address. If the desired IP address is not available in its cache, the resolver has to send the query to the first DNS server indicated by the DNS root. The process of constructing the query before sending it is described in "Figure 8".

Before sending the DNS request to the DNS server S2, the local name server has to do the following steps "Figure 8":

1) Elaborating the session key K_{TS}. K_{TS} is used only once as a private key for symmetric encryption;
2) Encrypting the request under K_{TS} key;
3) Encrypting K_{TS} key by K_{pubS2} public key generated by DNS root to obtain the token J_t;
4) Writing the new DNS message (EDNS message) "Figure 6";

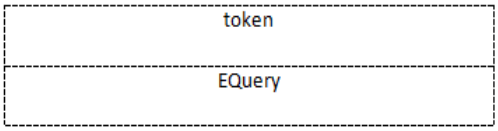

Figure 5. Format of EDNS Message

5) Hashing the EDNS message;
6) Encrypting the hash (EDNS message) with the private key K_{privS2} to have the signature 'sign';
7) Encapsulation of the request and sending it to the DNS server S2 "Figure 7" and "Figure 8".

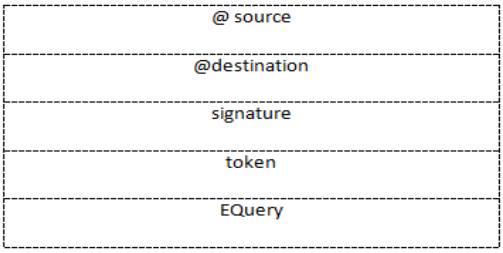

Figure 6. Format of the New ENC-DNSSEC Message

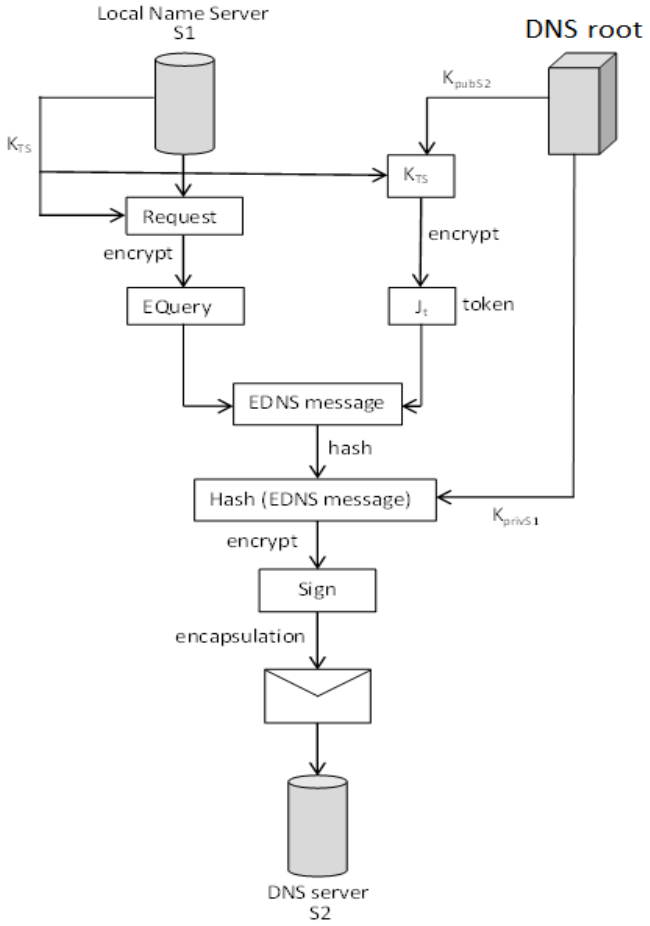

Figure 7. Process of Encapsulating Request in DNS Message

Once the authoritative server S2 is queried by the resolver of the local name server, it will decapsulate the query, look for the answer, encapsulating the response with the same method and finally, send it to the client. The process of decapsulating the query is described in "Figure 9" below.

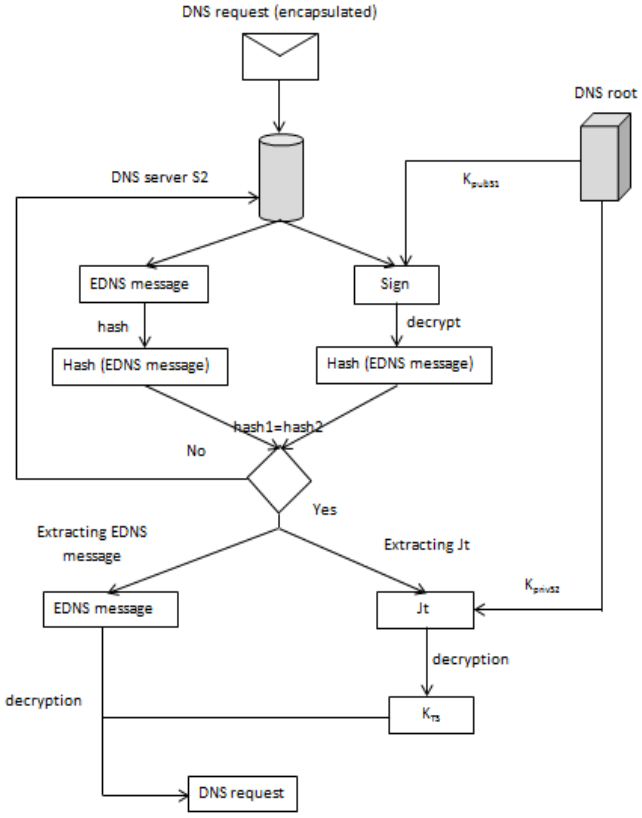

Figure 8. Process of Decapsulating Request in DNS Message

As illustrated in "Figure 9", when the server receives an envelope (DNS request encapsulated), it:

1) extracts the signature 'sign' and 'EDNS message'
2) hashes 'EDNS message' to obtain hash1
3) decrypts the signature with the public key K_{pubS1} to obtain hash2
4) compares hash1 with hash2
5) if hash1=hash2, the process continue, else, the process is interrupted
6) extracts the token J_t and the encrypted request from the hash message
7) decrypts the token with the private key K_{privS2} to obtain the session key K_{TS}
8) decrypts the encrypted message 'EDNS message' with the session key K_{TS} to obtain the clear DNS request.

5. Analysis of ENC-DNSSEC Protocol

To implement ENC-DNSSEC, we use two types of encryption, a symmetric encryption (using a single key) and asymmetric encryption (using a private-public key). According to the current version of this protocol, the successful symmetric cipher is DES (Data Encryption Standard) and concerns the step 2 of encapsulation and step 8 of decapsulation

while the selected algorithm for asymmetric encryption is RSA (Rivest Sahmir Adleman) and it concerns the step 3 of encapsulation and step 7 of decapsulation. The hash algorithm necessary and proposed to build the signature is SHA (Secure Hash Algorithm) and concerns Step 5 for encapsulation and step 2 for decapsulation. ENC-DNSSEC has to be implemented with DNSSEC protocol "Figure 9".

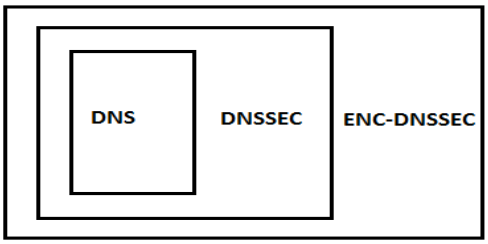

Figure 9. Position of ENC-DNSSEC with DNS and DNSSEC Protocols

As described above, authentication, confidentiality and data integrity are all absolutely necessary for DNS to deliver a safe service. Indeed, one must be able to ensure that the data received from a name server has not been altered and to verify the origin of data. In the initial structure of the DNS, no action has been taken to ensure the safety of the service and none of the security services was provided.

"Figure 10" presents a general overview of the possible sources of threat that can form the basis of any DNS security audit where every data path is considered as a potential source of threat.

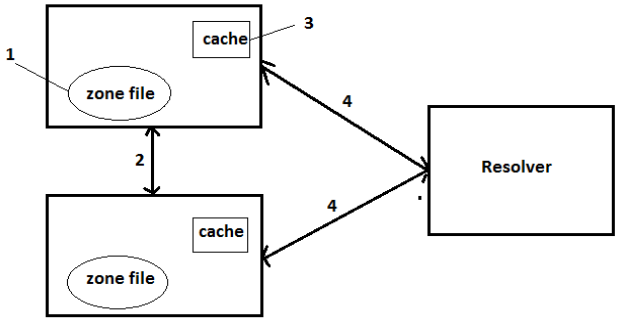

Figure 10. Security Overview

We can classify security threats in four categories:

- *Zone file (1)*: Attacks of zone files are generally provided from the local system. So, to protect zone files from these attacks, the system administrator has to define security policies and maintain them updated.

- *Dynamic update (2)*: DNS servers communicate with slave servers to execute zone transfers. ENC-DNSSEC is our proposal to encrypt data transiting between servers in order to protect DNS system from different type of attacks.

- *Cache (3)*: As known, the cache contains sensitive data, so due to IP spoofing, cache poisoning, data interception, and other hacks that can attack the cache which contains sensitive data, DNSSEC is considered as an essential solution to prevent it from these attacks.

- *Resolver (4)*: DNSSEC standards define all techniques and methods to prevent Resolver from attacks [4]. So, DNSSEC is considered actually sufficient to secure DNS resolver.

The "Table 2" below defines the potential outcomes of compromise at each point and the possible solutions.

Number	Area	Type of attack	Solutions					
			System administration	TSIG	DNSSEC	PK-DNSSEC	SK-DNSSEC	ENC-DNSSEC
1	Zone file (local)	File corruption	×			×	×	×
2	Dynamic update (server-server)	IP spoofing						×
3	Cache (server-client)	Cache poisoning			×	×	×	×
4	Resolver (remote cache client)	Data interception, IP spoofing		×	×	×	×	×

Table 2. Types of Attack at Each Point and the Possible Solutions

According to "TABLE 2", we can see that no ancient solution ensures data confidentiality between servers or between server and client, and the only solution that exist now is using IPsec to secure the canal, but this solution is still insufficient as we have explained in Section 3.

The principal objective of ENC-DNSSEC protocol is to enhance the DNS protocol security. ENC-DNSSEC aims to ensure authentication, confidentiality and integrity of DNS message without using a tunneling protocol like IP security (IPsec).

Authentication: In ENC-DNSSEC, authentication can be ensured by different ways:

a) Access control list: it's the first method deployed by BIND system in order to prevent DNS from IP spoofing attack. Access control lists give a list of IP addresses that are authorized to query the resolvers. This method is still inadequate to prevent DNS from attacks [5].

b) MAC function: As it's described in "Figure 6" and "Figure 9", when a server receives a query from another server, it calculates its MAC function and compare it with the hash, if they are the same, it considers that the request comes from an authorized name server and it treats it, else it rejects the request.

c) *DNSSEC:* by using DNSSEC, resolvers can verify the authentication of source data. Also, it uses signatures to authenticate parent with child zone (chain of trust) [4].

Confidentiality: by using ENC-DNSSEC, queries and answers are transmitted securely across the network; every request is encapsulated before being sent to the destination "Figure 8". This encapsulation which is based on symmetric and asymmetric cryptography in addition to MAC function is used in order to ensure a high level of security to data transmitted between servers.

Integrity: deploying ENC-DNSSEC generates a long process of encrypting and hashing requests before sending it to authoritative server, at the reception, the request is highly authenticated and encrypted. So, due to this complicated process of encapsulation, the destination is sure that the request is not modified or changed in transmission. Finally, the higher level of authentication and confidentiality in the encapsulation process ensure a high level in integrity of data.

6. Conclusion

In this paper, we give a proposal of a new protocol that can secure a request from the beginning of DNS resolution till the end. The encapsulation used in this proposal aim to protect data from all types of attacks. ENC-DNSSEC protocol includes all functionalities used by DNS and DNSSEC.

Furthermore, it ensures confidentiality and adds a new technique to authenticate servers. So, this technique could prevent DNS protocol from different attacks like IP spoofing and cache poisoning and even from attacks that could happen during the resolution process.

References

[1] R. Curtmola, A. Del Sorbo and G. Ateniese, "On the Performance and Analysis of DNS Security Extensions", in Proceedings of the 4th International Conference, CANS 2005, China, **(2005)**, pp. 288-303.
[2] K. Chetioui, G. Orhanou, S. El Hajji and A. Lakbabi, "Security of the DNS Protocol - Implementation and Weaknesses Analyses of DNSSEC", International Journal of Computer Science Issues, vol. 9, Issue 2, no 3, **(2012)**, pp. 340-345.
[3] G. Ateniese and S. Mangard, "A New Approach to DNS Security (DNSSEC)", in Proceedings of the 8th ACM Conference on Computer and Communication Security, ACM Press, **(2001)**, pp. 86-95.
[4] R. Aitchison, 'Pro DNS and BIND 10', Apress, **(2011)**.
[5] P. Albitz and C. Liu, 'DNS & BIND', O'Reilly, **(2006)**.
[6] D. Estlake, 'rfc 2535: Domain Names System Security Extensions', **(1999)** March.
[7] J. Kohl and C. Neuman, 'rfc 1510: The Kerberos Network Authentication Service (V5)', **(1993)** September.
[8] P. Mockapetris, 'rfc 1034: Domain Names, Concepts and Facilities', **(1987)** November.
[9] P. Mockapetris, ', rfc 1035: Domain Names, Implementation and Specification', **(1987)** November.

Authors

Kaouthar Chetioui, Ph.D student of computer science at the University Mohammed V – Rabat, Morocco. She received in 2011 a Master degree in Cryptography and Security of Information at the University Mohammed V – Rabat, Morocco and a license in 2009 in Network Technologies at University Sidi Mohammed Ben Abdellah – Fez, Morocco. Her main research domains include network and Internet protocols security.

Ghizlane Orhanou, an Associate Professor in the Computing Sciences Department, Faculty of Sciences, University Mohammed V – Rabat, Morocco. She received Ph.D degree in computer sciences from the University Mohammed V – Rabat, Morocco in 2011. She received in 2001 a Telecommunication Engineer diploma from Telecommunication Engineering Institute (INPT – Morocco). Her main research interests include networked and Information systems security.

Said El Hajji, Professor in the Mathematics Department since 1991 at Mathematical and Computer Sciences, Faculty of Sciences, University of Mohammed V-Rabat. Responsible of the Mathematics, Computing and Applications Laboratory. He received Ph. D degree from Laval University in Canada. His main research interests include modeling and numerical simulations, security in networked and Information systems.

