New Protocol E-DNSSEC to Enhance DNSSEC Security

Kaouthar Chetioui, Ghizlane Orhanou, and Said El Hajji
(Corresponding author: Kaouthar Chetioui)

Laboratory of Mathematics, Computing and Applications, Faculty of Science, Mohammed V University in Rabat
BP. 1014 RP, Rabat, Morocco
(Email: kaoutharchetioui@gmail.com)
(Received Sep. 28, 2016; revised and accepted Jan. 15, 2017)

Abstract

The Domain Name System (DNS) is an essential component of the internet infrastructure. Due to its importance, securing DNS becomes a necessity for current and future networks. DNSSEC, the extended version of DNS has been developed in order to provide security services. Unfortunately, DNSSEC doesn't offer query privacy; we can see all queries sent to resolver in clear. In this paper, we evaluate the security of DNS and DNSSEC protocols, and we would see clearly that DNSSEC is insufficient to secure DNS protocol; it doesn't ensure confidentiality to data transiting over the network. That's why, we propose a new method named 'E-DNSSEC' which aims to add, in addition to DNSSEC security features, queries confidentiality, by encrypting them between DNSSEC servers. After that, an implementation of E-DNSSEC protocol will be given. Finally, we conclude by an analysis to prove the positive impact of this method to enhance DNSSEC security.

Keywords: Confidentiality; DNSSEC; E-DNSSEC

1 Introduction

DNS is a distributed database globally accessible using a request/response architecture. The DNS protocol resolves domain names readable by humans to (IP) Internet Protocol addresses. So, the DNS resolution is the first step in any network communication. It is therefore essential that the DNS infrastructure be robust and secured [5]. That's why, we need to enhance DNS protocol security to be able to ensure at least authentication, integrity and confidentiality.

TSIG (Transaction Signatures) defined in RFC 2845 [10] is a solution used in order to ensure the integrity of channels; it allows two machines talking DNS to check the identity of the caller. Unfortunately, this mechanism does not authenticate source data, only it secures transmission data between two parties who share the same secret key. The original source data can come from a compromised zone master or can be corrupted during transit from an authentic zone master to some "caching forwarder" [10]. These signature mechanisms are reserved only to protect zone transfers and dynamic update messages. So, TSIG is mostly used between master and slave DNS servers to secure zone transfers and today almost all transfers between authoritative severs are protected by TSIG.

DNSSEC (DNS Security Extension) defined in [RFC4033-4035], is proposed and standardized in 1997 [3], it solves some security issues related to DNS protocol. DNSSEC secures data sent by DNS servers; it ensures two security objectives namely authentication and integrity of source of data. These extensions use cryptography to sign DNS records and put the signature in DNS. Thus, a suspicious DNS client can retrieve the signature and using the key of the server, it can check if data is correct. DNSSEC allows delegation of signatures and the register of a TLD (Top-Level Domain) can announce that this subdomain is signed. By using DNSSEC, we can also build a chain of trust from the root server.

Despite of services provided by DNSSEC protocol, it has some gaps which make its deployment slowed:

- The compatibility with the existent equipment and software;
- The deployment of DNSSEC in a wide range of DNS servers and DNS resolvers (clients);
- The protection of data transiting in the network that ensures confidentiality service;

When communications requires private channels, SSH or IPsec are used to interact with DNS. These technologies are considered as there is no DNS solutions proposed for this case. But, all of them suffer from different security problems [4].

In this paper, we propose a new method E-DNSSEC which uses cryptography to encrypt DNSSEC query transiting across the network. This method aims to add

a new strong security service to DNSSEC protocol and consequently enhance security in DNS service and Internet communications. We describe first the structure of DNSSEC message especially DNSSEC query which is subject of different type of attacks. We'll give some types of these attacks which justify the necessity of thinking about new method to secure data in transit. Then, we explain the new E-DNSSEC protocol by giving different steps of the query encryption. The result of the E-DNSSEC implementation will be presented. After that, we give security analysis of the results obtained in order to prove the efficiency of our method to enhance DNSSEC security. Finally, we conclude by comparing this solution with other existing methods and we give some perspectives.

2 DNSSEC Structure and Security Issues

In this section, we focus on the structure of DNSSEC query and DNSSEC resolution process and finally, we describe the limitations of DNSSEC protocol.

As DNSSEC is the extended version of DNS, it has the same tree structure as DNS [9], but it adds some improvements; it includes new records, services and techniques to secure DNS protocol. DNSSEC uses cryptography to secure zone files. So, each zone has at least a pair of key. The public key of the child zone (like "example.ma") is signed by the private key of the parent zone (in this example ".ma") with the exception of the root which is signed by itself. This process forms the trust chain (Chain of Trust).

DNSSEC uses cryptographic keys ZSK (Zone Signing Key) and KSK (Key Signing Key) and adds new resource records (RR KEY, SIG, NSEC and DS) to DNS messages in addition to the original DNS service records [1, 6].

1) DNSKEY RR record: The DNSKEY resource record stores all public key pairs that are necessary for signing the zone.

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

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

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

In addition to that, DNSSEC slightly modify the header of the classic DNS packet with the use of AD and CD bits:

- AD (Authenticated Data): As specified by RFC 2535 [7] indicates in a DNS response that all information included in the "Answer" and "Authority" have been authenticated by the server according to its security policy. However, in practice this does not seem very useful since a properly configured DNS server should not respond to a request with data that have not been authenticated.

- CD (Checking Disable): This bit specifies whether the resolver accepts unverified responses, when set to 1. Otherwise (value 0), the principle of verification is active.

When a DNSSEC resolver (client) sends a query to DNSSEC (server), the 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. We keep the standard format of query as presented in Figure 1.

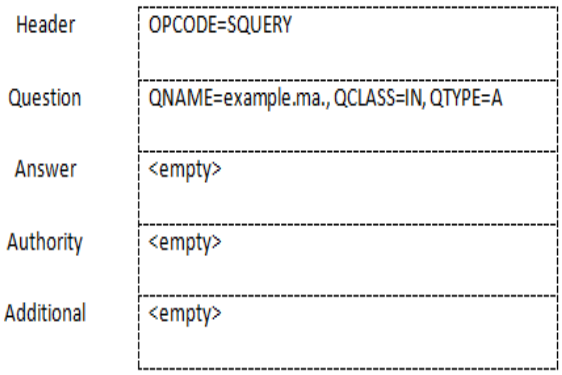

Figure 1: Format of standard query in DNS message [10]

As mentioned in Figure 1, the standard query contains other fields in addition to the domain named entered by the client (ex: "example.ma"). The communication in DNS follows the client/server model. So, when the server receives the query, it looks for the response in its database to have the IP address associated for the desired domain name, especially, it makes search in zone files which contain all information about domains names.

The process of DNSSEC resolution is described in Figure 2.

As described in Figure 2, we observe that the query (1) sent by recursive server to the authoritative server on a subdomain is neither encrypted nor signed.

Referring to RFC4033 [3], many security services are not provided by DNSSEC:

- DNSSEC doesn't provide confidentiality;

- DNSSEC doesn't provide access list control;

- DNSSEC doesn't protect from denial of service attacks.

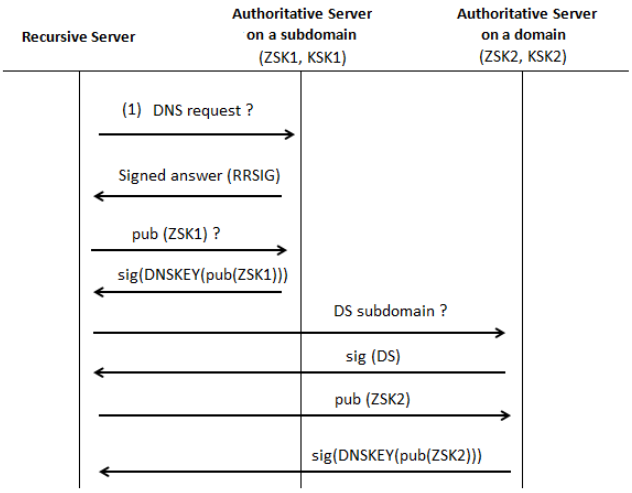

Figure 2: Diagram of DNSSEC resolution process

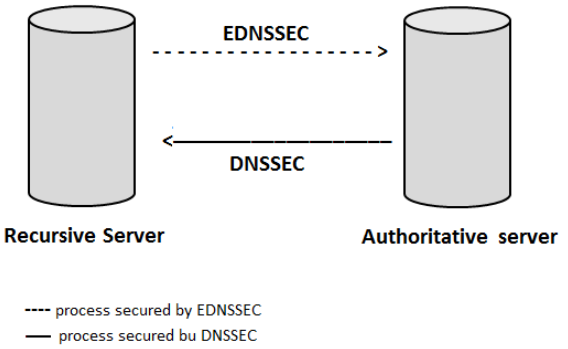

Figure 3: Processes secured by DNSSEC and E-DNSSEC protocols

In addition, DNS security extensions use public key cryptography to sign and authenticate DNS resource records, RFC 4035 [3]. An active attacker who can obtain the CD bit in a DNS query message or the AD bit in a DNS response message can use these bits to decrease the protection that DNSSEC attempts to provide to resolvers in recursive mode.

For this reason, the use of these control bits by a DNSSEC resolver in recursive mode requires a secure channel.

In the next section, we present the new method E-DNSSEC which encrypts data (DNSSEC query) transmitted across the network. This solution aims to secure the channel between DNS servers.

3 The New E-DNSSEC Protocol

In this section, we will describe our proposal which consists on E-DNSSEC protocol. We begin by a description of DNSSEC resolution process and after that, we present the new E-DNSSEC process in order to enhance DNSSEC security level. As we aim to encrypt DNSSEC query between DNSSEC client and DNSSEC server, we will refer to our proposal as E-DNSSEC, (Encrypted DNSSEC query).

The idea of E-DNSSEC protocol is to take the query from recursive server and encrypt it before sending it to the authoritative server; and in the reception, the authoritative server decrypts it before starting resolution and finally, it sends the response secured using DNSSEC protocol. So, the principal objective of this method is to combine DNSSEC properties with E-DNSSEC protocol to secure DNS message from the beginning of resolution to the end consequently ensuring authenticity, confidentiality and integrity of data transiting in the network.

In our demonstration as mentioned in Figure 3, we use two entities: a resolver (handling outgoing requests and incoming responses) and a DNS server (handling incoming requests and outgoing responses). These entities communicate on same DNS server because we suppose that the request received by the server needs recursion to be resolved. In recursive resolution, a stub resolver is created to be a DNS client; the stub resolver after consulting /etc/hosts, sends a recursive DNS query to a DNS server. In this case, the DNS server asks the resolver for the resolution of a request and sends the response to the stub resolver or another that queried it.

In the following, we will describe what happens during a resolution process inside a DNS server during the lifetime of a DNS query. Figure 4 below explains the steps of DNSSEC resolution.

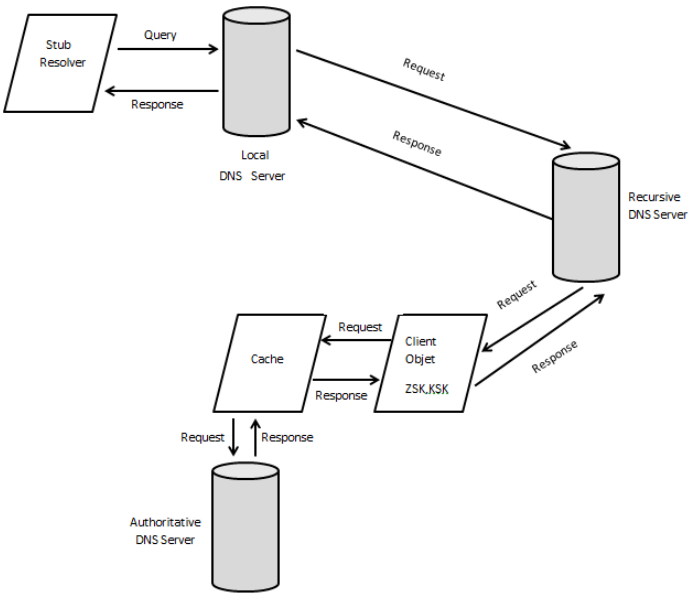

Figure 4: DNSSEC resolution process

Referring to Figure 4, the process of DNSSEC resolution starts when the stub resolver sends a query to its local DNS server. The local DNS server dispatches the re-

quest to the client manager, which is responsible for principal DNS communication, it's attached to each network interface and it manipulates all DNS messages arriving to the DNS server. After receiving the request by client manager, a client object is created in order to resolve the request. First, it verifies the public key signature, if the DNSSEC verification is successful; it looks for the response in the cache and in the authoritative DNS server. Once the response is found, it returned it to the DNS server, which sends it back to the Stub Resolver [8].

In our study, we focus on encrypting the query before sending it to the DNS server. So, E-DNSSEC keeps the structure of DNSSEC protocol and add the process of encrypting the query in a DNS message. Figure 5 shows when the process of encryption starts and finishes.

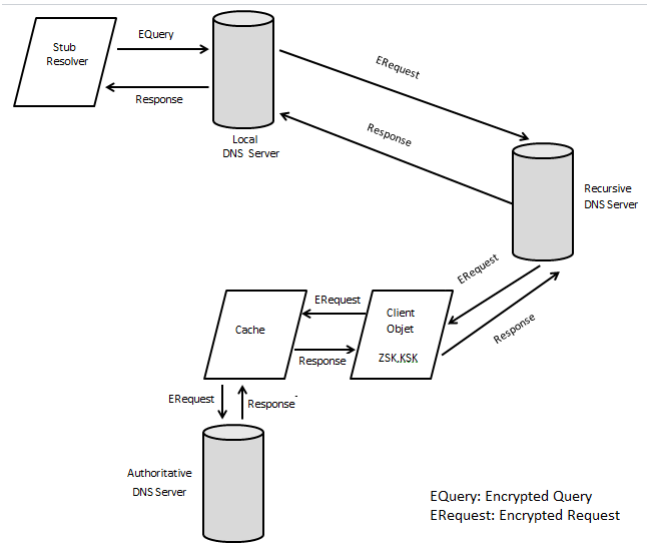

Figure 5: E-DNSSEC resolution process

To resolve a query by E-DNSSEC protocol, the local DNS server encrypts the query and inserts it in the DNS request. When the recursive DNS server receives the DNS request, it recuperates the E-query and decrypts it. The client object verifies public key signatures as usually done by DNSSEC protocol and looks for the response in its cache, if the response is found in the cache it returns it to the DNS server, else, the process of resolution continues by encrypting the query and sending the request to the authoritative DNS server. Once the response is found, the process of encrypting query is stopped. After that, the response is sent to the DNS server and the Stub Resolver. During this step, the response is secured using signed resource records of DNSSEC protocol.

To implement E-DNSSEC protocol, we use the original version BIND9. BIND is a complex program with its own tasks, threads, scheduler and memory management. We select version 9 of BIND, its source code is freely available. So, in this implementation, we look after the IP address of the domain name 'example.ma' and we use RSA algorithm for encryption. The different steps of our implementation are described below:

DNS ——> DNSSEC ——> E-DNSSEC

1) Firstly, we implement a simple DNS server with the domain name 'example.ma' and we interrogate it with 'dig' command:

```
; <<>> DiG 9.9.5 <<>> server.example.ma
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19267
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;server.example.ma.              IN      A

;; ANSWER SECTION:
server.example.ma.      86400   IN      A       192.168.1.1

;; AUTHORITY SECTION:
example.ma.             86400   IN      NS      192.168.1.1.example.ma.
example.ma.             86400   IN      NS      192.168.1.10.example.ma.
```

Figure 6: Result of Dig command using DNS

Figure 6 represents the result of this interrogation of a simple DNS server by 'dig' command. The response reflects a typical positive response to a dig command and includes the following items: Question Section, Answer Section, Authority Section and Additional Section. In this paper, we are interested only to the Question Section. Figure 7 shows that this section transit in clear across the network and could be intercepted by a malicious person [6].

2) We implement DNSSEC protocol. This implementation is done following several steps [6]: creating keys (ZSK, KSK), including the keys in the zone files, signature of the zone and reconfiguration of the 'named.conf' using DNSSEC. The results of 'dig' command are illustrated in Figure 8.

   As shown in Figure 8, after implementing DNSSEC protocol, the output of 'dig' command is changed especially in the Answer Section; the response is signed but the Question Section is still unchanged.

3) Based on DNSSEC architecture, we enhanced BIND source files by implementing the query encryption. We interrogate the server by 'dig' command as shown in Figure 9.

Figure 9 indicates that the Question Section is different compared to the previous result; the query 'server.example.ma' is encrypted.

4 Analysis Of E-DNSSEC Protocol

In this section, we aim to compare the new E-DNSSEC protocol with existing protocols. Furthermore, having good security results by using E-DNSSEC protocol is very

```
[root@localhost ~]# tcpdump -nn host 192.168.1.10 and host 192.168.1.1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
15:41:54.340664 IP 192.168.1.10.51965 > 192.168.1.1.53: 49733+ A? server.example.ma. (35)
15:41:54.340776 IP 192.168.1.1 > 192.168.1.10: ICMP host 192.168.1.1 unreachable - admin prohibited, length 71
```

Figure 7: DNS query captured by Tcpdump

```
[root@localhost ~]# dig +cd +multi example.ma dnskey
;; Truncated, retrying in TCP mode.

; <<>> DiG 9.6.1b1-RedHat-9.6.1-0.3.b1.fc11 <<>> +cd +multi example.ma dnskey
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41792
;; flags: qr aa rd ra cd; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;example.ma.            IN DNSKEY

;; ANSWER SECTION:
example.ma.             86400 IN DNSKEY 257 3 5 (
                                AwEAAcQK6XP43phN3h4x7eMbpkEidvPjWmeynsZAfECq
                                quRMdplwgkuMftFX3HsjBpo5u7IhGYc1MkyeTV8vTajq
                                rIcm4TerWq3PkjL+0iKUTPo3mca5vjxRyw5CNORLYbwA
                                ZhIP/RAOZ1eKW0tX/46z5FI8crFm1ycX1Dp2eJYQmuuA
                                ODOKorEX2biALW1oSmxshvq66yqbew3MVwMFFVbu4i/J
                                LzfgpXosgGKcTHaFIYNiR15JJm9g0x/QBQM58LJUyyOr
                                5OzmCDny5hwYjbfmbyOkW82/KSQZ9mR4QIDexf6UH5ug
                                c7OOJBnVjCH+yvKC92NuTProcGiWlCs=
                                ) ; key id = 25924
```

Figure 8: Result of Dig command using DNSSEC

```
[root@localhost ~]# dig server.example.ma +dnssec

; <<>> DiG 9.9.5 <<>> server.example.ma +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 16884
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;A33301A32D126CA38.             IN      A
```

Figure 9: Result of Dig command using E-DNSSEC

useful to prevent our systems from different types of attacks, but having good performance is also very important. So, we present also the impact of this method in the DNSSEC performance. The principal objective of E-DNSSEC protocol is to enhance the DNS protocol security. For that, we propose a new method that encrypts DNS query and we aim to ensure confidentiality without using a tunnelling protocol like IP security (IPsec). This new method prevents DNS system from cache poisoning and Man in The Middle attacks. Authentication: In E-DNSSEC, authentication can be ensured by different ways:

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 [7].

MAC function: 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.

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) [2].

Confidentiality: By using E-DNSSEC, queries are transmitted securely across the network; every query is encrypted before being sent to the destination. This encryption is used in order to ensure a high level of security to data transmitted between servers.

Integrity: Deploying E-DNSSEC has a major advantage is that it keeps DNSSEC properties. As known, by using DNSSEC, DNS servers are required to sign the Resource Records in zone for which they are authoritative and answers the queries by returning the corresponding SIG RRs. DNSSEC uses new resource records to ensure integrity of data. So, due to this signed resources records, 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.

Our comparison is based essentially on the frequent DNS attacks like IP spoofing, file corruption and cache poisoning which are the famous attacks that make DNS system very weak. These attacks affect all DNS communication, whether it was between two DNS servers or a DNS server and a client.

Table 1: Types of attacks at each point and the possible solutions

Number	Area	Types of Attacks	Solutions			
			System Adm.	TSIG	DNSSEC	E-DNSSEC
1	Zone file (local)	File corruption	X			X
2	Dynamic update (server-server)	IP spoofing				X
3	Cache (server-client)	Cache poisoning			X	X
4	Resolver (Remote cache-client)	Data interception, IP spoofing		X	X	X

Table 1 below defines the position of different attacks in each area of a DNS system and the possible solutions.

According to Table 1, 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 2.

5 Conclusion

In this paper, we have proposed a new method E-DNSSEC. E-DNSSEC keeps all security properties of DNSSEC and encrypts the query in order to secure DNS across the network from different type of attacks. This solution adds confidentiality service to DNS system in addition to the authentication and integrity which are ensured by DNSSEC protocol. Furthermore, we have presented a functional implementation of E-DNSSEC protocol, we compare this method by other existing solutions and finally we discuss the impact of this solution on the security performance of DNS protocol.

References

[1] R. Aitchison, *Pro DNS and BIND 10*, Apress, 2011.

[2] P. Albitz, and C. Liu, *DNS & BIND*, O'Reilly, 2006.

[3] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, *DNS Security Introduction and Requirements*, RFC 4033, Mar. 2005.

[4] D. Atkins, R. Austein, *Threat Analysis of the Domain Name System (DNS)*, RFC 3833, Aug. 2004.

[5] P. Barnes, "Using DNS to protect networks from threats within," *Network Security*, vol. 2014, no. 3, pp. 9–11, Mar. 2014

[6] 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, no. 2, pp. 340–345, 2012.

[7] D. Estlake, *Domain Names System Security Extensions*, RFC 2535, Mar. 1999.

[8] D. Herrmann, M. Maa and H. Federrath, "Evaluating the security of a DNS query obfuscation scheme for private Web surfing," in *IFIP International Information Security Conference SEC 2014: ICT Systems Security and Privacy Protection*, pp. 205–219, Marrakech, Morocco, June 2014.

[9] A. Velagapalli and M. Ramkumar, "Trustworthy TCB for DNS servers," *International Journal of Network Security*, vol. 14, no. 4, pp. 187-205, July 2012.

[10] P. Vixie, O. Gudmundsson, D. Eastlake and B. Wellington, *Secret Key Transaction Authentication for DNS (TSIG)*, RFC 2845, May 2000.

Biography

Kaouthar Chetioui, she has received her Ph.D degree in 2017 in computer science from Mohammed V University in Rabat, Morocco. She received in 2011 a Master degree in Cryptography and Security of Information from the same university and in 2009 a licence in Network Technologies from Sidi Mohammed Ben Abdellah University, Fez, Morocco. Her main research domains include networks and Internet protocols security.

Ghizlane Orhanou, Professor in Faculty of Sciences and member of the Laboratory of Mathematics, Computing and Applications, Mohammed V University in Rabat, Morocco since 2013. She received Ph.D degree in Computer sciences from the Mohammed V University in Rabat in 2011 and the Habiltation to direct theses in 2016 from the same University. She received in 2001 a Telecommunication Engineer diploma from Telecommunication Engineering Institute (INPT - Morocco), and worked for about 3 years as GPRS and Intelligent Network Engineer, and for 9 years as System and Network Security Engineer. Her main research interests include network 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 networks and Information systems.