International Journal of Security and Its Applications
Vol.9, No.12 (2015), pp.313-322
http://dx.doi.org/10.14257/ijsia.2015.9.12.29

Encryption of Query in DNS Message

Kaouthar Chetioui, Ghizlane Orhanou and Said El Hajji

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

Abstract

DNS is an integral part of the internet infrastructure. It's one of the principal elements in all IP communications. Since its invention in 1983, the protocol has evolved to overcome its various limitations. This article proposes a new solution to secure DNS protocol which consists on encrypting query in DNS message between 'master' and 'slave' servers. We will see first an overview of the name resolution via DNS protocol. We will give the basic information about the resolution process in DNS. Then, we will expose some DNS vulnerabilities such as the creation or modification of messages and cache poisoning. After that, we will propose a new solution that will help to ensure the confidentiality in exchanges between DNS server and client and will also ensure the availability of the DNS architecture. Finally, we will conclude with an analysis of the benefits and the weaknesses of this solution.

Keywords: *DNS; Query; Encryption; Confidentiality*

1. Introduction

The domain name system (DNS) is a crucial element of the internet. Mapping a domain name (*e.g*, "example.ma") to an IP address is a behavior that everybody do when they connect to the internet and especially when they want to have access to websites. Unfortunately, this service doesn't include efficient techniques that can stop attackers from modifying or injecting false DNS queries.

Users visiting websites can be redirected to false addresses, they become vulnerable to different types of attacks such as cache poisoning or DNS spoofing. Consequently, attackers can manipulate DNS messages easily.

The main contribution of this paper is to secure the DNS message by encrypting queries sent by clients and servers to DNS servers. First, we will give some related works that present different researches about security of DNS query. After that, we will explain the process of DNS resolution by giving different types of queries that we have to encrypt. We will also present the benefits and limits of two methods: TSIG and DNSSEC in the security of the DNS protocol and then, we will present the necessity of encryption DNS queries to ensure confidentiality in DNS transaction.

After having described our method, we will show clearly the results by interrogating a DNS server and capturing the encrypted traffic sent across the network. Finally, we will conclude by discussing our results and giving some perspectives.

2. Related Works

The RFC 1034 introduces the domain name system (DNS) [6]. It gives details about DNS messages, specifies the role of resolvers solicited by sending queries to name servers. The RFC also describes the resolution process by which resolvers retrieve IP address of a domain name.

ISSN: 1738-9976 IJSIA

The RFC 3833 published in August 2004 [9], analyzes threats of the domain name server (DNS). It talks about vulnerabilities of the DNS protocol; it also discusses problems of security which are not resolved by DNSSEC protocol, for example the need to sign the 'non-existence' of a domain name message.

C. Perez and G. Alfaro [2] analyze the use of statistical noise for the construction of proper DNS queries. They propose a new solution that aims to reduce the risk that sensible data within DNS queries could be inferred by local and remote DNS servers.

Finally, D. Herrman *et al*. propose query obfuscation schemes [3] to provide query privacy. They make their experience in a real-world web surfing scenario and they discuss the benefits of their method in the DNS query security.

All the proposals presented overhead study the security of the DNS queries but the proposed solutions don't ensure data confidentiality in DNS transactions. For this reason, we thought of a new method that can enhance the security of DNS queries by introducing encryption.

3. Presentation of DNS Operation

The DNS is a hierarchically distributed database, whose main function is to translate domain names to their corresponding IP addresses. Initially, the Internet was formed by few machines. So name resolution was made easily from a file "hosts.txt" which included all the areas' names and addresses. But now, given the large number of machines connected to the Internet, the resolution has become much more complex: for each area, there is a name server called "primary name server "and a "secondary name server" that replaces the primary name server in case of malfunction or unavailability.

DNS has zone files that define it to be authoritative for some domains and slave for others and be configured to provide different behaviors for other domains or users.

When DNS client wants to resolve an IP address domain name, for example when we want to consult to a web page with URL www.example.ma, the procedure described in the following decomposable diagram process is executed ("Figure 1"):

Before being able to connect to the desired web server, the client must look for the IP address corresponding to the web URL.

1. The client checks, first, if the desired IP address is available in its own local resolver cache. The local resolver cache contains all mappings of hostnames / IP addresses that have been previously resolved. The DNS name cache is stored in random access memory, in the client machine, which accelerates the process of resolving host names when the user accesses frequently to the same server.
2. If the desired IP address is not present in the local resolver cache, the client consults the hosts file "/*etc*/hosts".
3. If the mapping is not found in the hosts file, the client will send a DNS query to the first DNS server whose IP address has been defined in its TCP / IP settings.
4. If the first DNS server is unreachable, it sends a request to the second server and so on ...

At the end of this process, if no IP address was found, the client cannot obtain the corresponding IP address and cannot contact the web site "www.example.ma". In all cases the results of the DNS query will be recorded in the local resolver cache on the client side.

In the present paper, we are especially interested in the DNS Queries security.

4. DNS Security Issues

As described above, the resolution of DNS names needs to transmit queries across a network. The client sends the query to the cache and the host files in the local system. After that, the query is sent to the DNS server that can be in the same network or located in another network, to get an answer.

Unfortunately, this service doesn't include any mechanism to secure transactions and indeed the exchanged data. And therefore, the DNS remains highly vulnerable to attacks such as the creation or modification of DNS messages, cache pollution, deny of service, *etc*.

"Figure 2." below shows where attacks are possible. Each possible vulnerability is marked with a question mark [1].

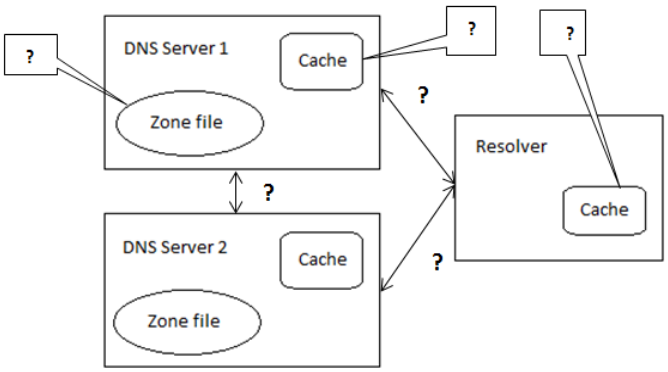

Figure 2. Weaknesses of DNS

We can see that there are weaknesses located between servers during DNS message exchanges, as well as potential intrusions in the zone file or in the cache level. During a DNS resolution by a cache server, the attacks may be multiple.

To overcome DNS security weaknesses and to satisfy some of the DNS security needs, a number of solutions were suggested.

Indeed, there is three methods (TSIG, SIG(0) and TKEY) for cryptographically securing zone transfers, but in practical purposes, the only method available to secure zone transfers is TSIG [4].

TSIG (Transaction signatures) uses a MAC with a shared secret both to authenticate and ensure the data integrity of every transaction involved in zone transfers between slave server and its master. The shared-secret data is never placed in the DNS zone files, instead, the shared secret is used by the two servers when exchanging data, such as a zone transfer.

On the other side, DNSSEC (Domain Name System Security Extensions) is a protocol based especially on asymmetric cryptography; it provides two essential security services to DNS: DNSSEC ensures data integrity and authentication of the source data [4]. "Figure 6" below illustrates the use of public keys in DNSSEC protocol. A number of observers were suggesting that DNSSEC will be the default state for all zones.

However, all these methods don't ensure the data confidentiality and the DNS protocol is still vulnerable to some types of attacks.

We can classify security threats in four categories:

- Local threats: It designates threats which are provided from the interior of the system. These type of attacks are usually easy to prevent by updating system-administration policies.

- Server to server threats: It designates threats coming from the network as unauthorized updates or IP spoofing. For these types of attacks, we can use TSIG and TKEY for authenticating requesting sources and destinations, but it only minimizes the risk of attack, and it is still inadequate to ensure data confidentiality.
- Server to client threats: It designates threats coming from the network as cache poisoning by IP spoofing, date interception or subverted master or slave. However, if the website contains sensitive data, DNSSEC solution may be essential.
- Remote cache-client: It designates threats in the cache client as data interception or poisoned cache and it also can be resolved by the DNSSEC protocol.

As we can see, no solution has been proposed to ensure data confidentiality between servers.

We aim, in the present paper, to introduce confidentiality during the resolution process.

5. Security in DNS Query Message

In this section, we describe our proposal that can secure the query in a DNS message during the resolution process. The sender resolver encrypts the query before transmitting the DNS request message across the network. “Figure 3” below explains the position of the query in DNS message.

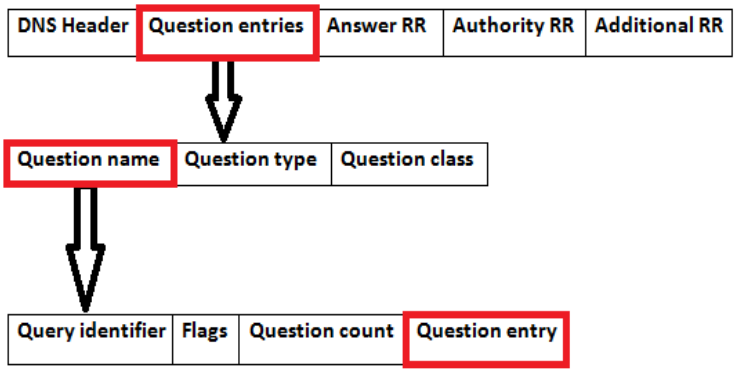

Figure 3. The Query Position in DNS Message

The general idea behind our proposal is quite simple but efficient. Acting as a resolver, we look after the IP address of the domain name “example.ma”. The query (for this example, the query is equal to “example.ma”) is encrypted and then transmitted over the network.

To implement and test our solution, we have used DNS client and server based on the BIND. BIND is a complex program having its own tasks, threads and memory management entity. We selected version 9 of BIND, its source code is freely available.

Initially, with the original version of Bind 9, without the confidentiality issue, we have performed the name resolution process to resolve “serveur.example.ma”. For that, we have executed the following command: ‘dig server.example.ma’ which send a DNS request message to one of the authoritative name servers for the “example.ma” zone.

Dig (domain information groper) command is a network administration command-line tool for querying servers. We use also "tcpdump" to capture network packets. This is very useful to examine and analyze security problems.

The "Figure 4" below shows the dig command output.

```
; <<>> 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.

;; Query time: 5 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Tue Nov 18 15:32:53 PST 2014
;; MSG SIZE  rcvd: 115
```

Figure 4. Result of Dig Command (Bind 9)

The output from a dig command is a formatted version of wire format; the message response to the query is formed from the dig command parameters. The preceding response reflects a typical positive response to a dig command and includes the following items [4]:

• The QUESTION SECTION: reflects the original query that is being answered; in this case, it's a query for the A RR of "server.example.ma".

• The ANSWER SECTION: provides the four A RRs for "server.example.ma" that fully answers the question in this case. If the ANSWER SECTION is present but contains no entries, then the query was not successful; the status field in the HEADER typically provides the reason of this situation unless the response was a referral, in which case the status field will be NOERR.

• The AUTHORITY SECTION: provides the NS RRs of the servers that are authoritative for the domain "example.ma".

• The ADDITIONAL SECTION: provides information that may be useful to the server; in this case, it is the A RRs of the name servers.

"Figure 5" below presents the DNS traffic captured by tcpdump tool when transiting over the network between the slave and the master DNS. We note that the DNS query is sent in clear in the network, and this makes it vulnerable to different types of attacks like 'Man In The Middle' attack.

```
[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 5. DNS Query Captured by Tcpdump (Clear)

So, our goal, as explained in the section above, is to enhance actual DNS protocol by introducing the confidentiality issue to secure data exchange between two

servers. Precisely, we encrypt 'QNAME' which is a part of question section [6] just after its reception by the local resolver. Thus, the encrypted QNAME can be transited securely across the network before arriving to the distant resolver to look for an answer.

To resolve a query, the local DNS server starts the recursion process as illustrated in "Figure 6" below. After the resolver has constructed the query message (Q) and before sending it out to an external DNS server, it encrypts it (Qe). The resolver in the remote DNS server receiving (Qe) decrypts it before resolving the domain name to IP address.

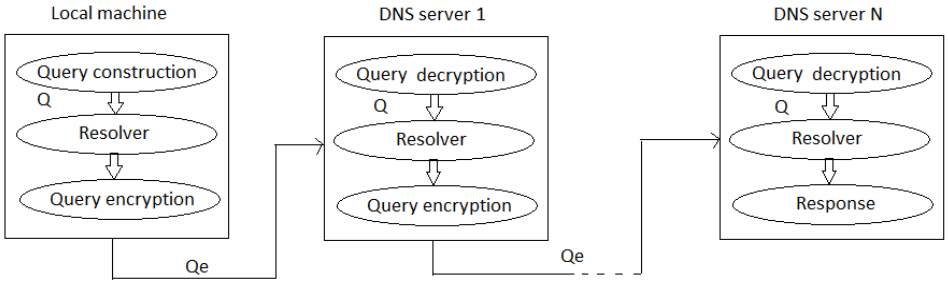

Figure 6. Process of Securing DNS Query in an Iteration Resolution

In this case, after introducing the confidentiality issue, when DNS client wants to resolve an IP address domain name, we take the same example as before: resolving the URL www.example.ma, the resolution follows the activity diagram illustrated in "Figure 7" below.

The description of the main steps of the diagram "Figure 7" is as follows:

1. As described in "Figure 1", the client checks, first, if the desired IP address is available in its own local resolver cache.
2. If the desired IP address is not present in the local resolver cache, the client consults the hosts file "/*etc*/hosts".
3. If the mapping is not found in the hosts file, the DNS client resolver will first encrypt the DNS query, and send the whole DNS message to the first DNS server whose IP address has been defined in its TCP / IP settings.
4. The DNS server decrypts the DNS query before starting the resolution process.
5. If the first DNS server is unreachable, it sends a DNS request message (containing the encrypted query) to the second server and so on …

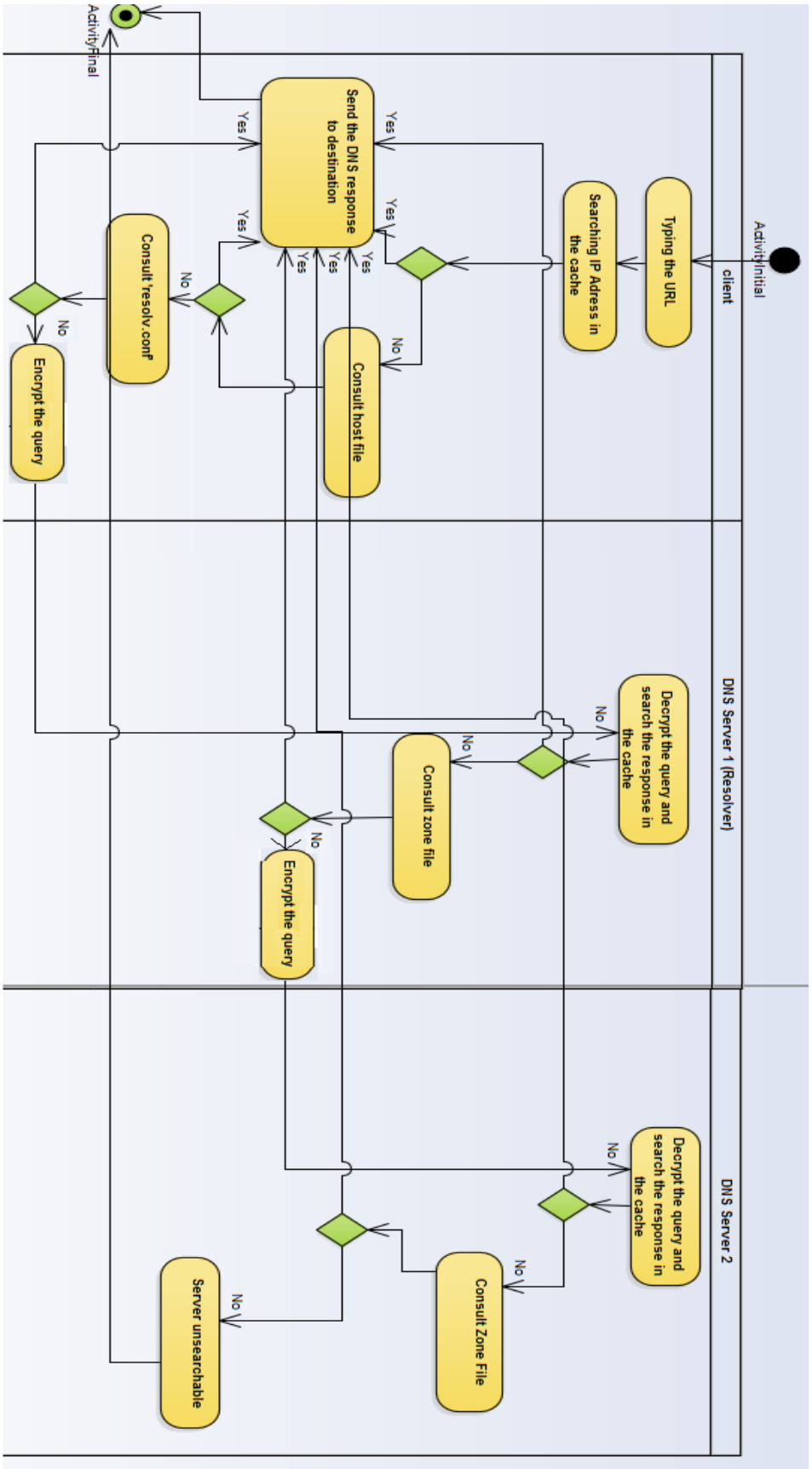

Figure 7. Enhanced Domain Name Resolution Process (Encrypted Query)

As shown in “Figure 7”, the encryption of DNS query starts when DNS client has to consult a DNS server, so all transactions across the network are encrypted. And, even if packets are sniffed by a malicious person, it won’t be easy to know what the client query is.

To show that the DNS query sent by the ‘slave’ server to ‘master’ server’ was encrypted, we use “Tcpdump” tool whose output is presented in “Figure 8”.

```
[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
15:54:22.008264 IP 192.168.1.10.68 > 192.168.1.1.67: BOOTP/DHCP, Request from 00:0c:29:0f:99:8a, length 300
15:54:22.008362 IP 192.168.1.1 > 192.168.1.10: ICMP host 192.168.1.1 unreachable - admin prohibited, length 336
15:54:24.060806 IP 192.168.1.10.38112 > 192.168.1.1.53: 20267+ [1au] A? rdqudq.dwzlokd.lz. (46)
```

Figure 8. Encrypted DNS Query Captured by Tcpdump

We can also see this encryption in the dig command output in “Figure 9” below.

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

; <<>> DiG 9.9.5 <<>> server.example.ma
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 31232
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;rdqudq.dwzlokd.lz.             IN      A

;; Query time: 104 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Wed Dec 17 01:33:21 PST 2014
;; MSG SIZE  rcvd: 46
```

Figure 9. Result of Dig Command (Enhanced Version)

From “Figure 8” and "Figure 9”, we can see that 'server.example.ma' representing the “request data” has been encrypted. Thus, we can ensure the DNS query confidentiality when transiting over the network.

6. Conclusion

In this paper, we described our proposition to enhance DNS security by encrypting query before its transmission across a secured or an unsecured network.

The effectiveness of our work has been demonstrated using experimental results, by encrypting a real DNS query in Bind 9 and sending it over a network between a DNS client and a DNS server. This will contribute to improve security for communication between DNS clients and servers as well as between DNS servers, it will also facilitate the prevention from malicious activities on the internet more effectively.

References

[1] 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, pp. 340-345 **(2012).**

[2] C-Perez, S.García-Alfaro, J.: Anonymous Resolution of DNS Queries. In: On the Move to Meaningful Internet Systems. pp. 987–1000. Springer, LNCS 5332 **(2008).**

[3] D. Herrmann, Max Maaß, and H. Federrath: Evaluating the Security of a DNS Query Obfuscation Scheme for Private Web Surfing, ICT Systems Security and Privacy Protection, IFIP Advances in Information and Communication Technology , vol 428, pp. 205-219 **(2014).**
[4] R. Aitchison, "Pro DNS and BIND 10", Apress, **(2011).**
[5] P. Albitz & Cricket Liu, "DNS & BIND", O'Reilly, **(2006).**
[6] P. Mockapetris, rfc 1034: "Domain Names, Concepts and Facilities", November 1987.
[7] P. Mockapetris, rfc 1035: "Domain Names, Implementation and Specification", November **(1987)**.
[8] M. Crawford, rfc 2673: "Binary labels in the Domain Name System", August **(1999)**.
[9] D. Atkins and R. Austein, RFC 3833: "Threat Analysis of the Domain Name System", August **(2004)**.

Authors

Kaouthar Chetioui, she is a 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, she is 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, he is a 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.