Anonymous Resolution of DNS Queries

S. Castillo-Perez[1] and J. Garcia-Alfaro[1,2]

[1] Universitat Autònoma de Barcelona,
Edifici Q, Campus de Bellaterra, 08193, Bellaterra - Spain,
scastillo@deic.uab.es

[2] Universitat Oberta de Catalunya,
Rambla Poble Nou 156, 08018 Barcelona - Spain,
joaquin.garcia-alfaro@acm.org

Abstract. The use of the DNS as the underlying technology of new resolution name services can lead to privacy violations. The exchange of data between servers and clients flows without protection. Such an information can be captured by service providers and eventually sold with malicious purposes (i.e., spamming, phishing, etc.). A motivating example is the use of DNS on VoIP services for the translation of traditional telephone numbers into Internet URLs. We analyze in this paper the use of statistical noise for the construction of proper DNS queries. Our objective aims at reducing the risk that sensible data within DNS queries could be inferred by local and remote DNS servers. We evaluate the implementation of a proof-of-concept of our approach. We study the benefits and limitations of our proposal. A first limitation is the possibility of attacks against the integrity and authenticity of our queries by means of, for instance, man-in-the-middle or replay attacks. However, this limitation can be successfully solved combining our proposal together with the use of the DNSSEC (DNS Security extensions). We evaluate the impact of including this complementary countermeasure.

Key words: IT Security, Privacy, Anonymity, Domain Name System, Privacy Information Retrieval.

1 Introduction

The main motivation of the present work comes from privacy and security concerns regarding the use of the protocol DNS (Domain Name System) as the underlying mechanism of new Internet protocols, such as the ENUM (*tElephone NUmber Mapping*) service. ENUM is indeed a set of service protocols used on VoIP (Voice over IP) applications. One of the main characteristics of ENUM is the mapping of traditional phone numbers associated to the ITU-T (International Telecommunications Union) E.164 recommendation, to URIs (Universal Resource Identifiers) from VoIP providers, as well as to other Internet-based services, such as e-mail, Web pages, etc. We overview in this section some of the features of this service, as well as some security and privacy concerns regarding the use of the DNS protocol in ENUM.

1.1 The ENUM Service

The ENUM service is a suite of protocols used in VoIP applications whose main goal is the unification of the traditional telephone E.164 system with the IP network of the Internet. Designed and developed by the *Internet Engineering Task Force* (IETF) in late nineties, ENUM allows the mapping of IP services by using an indirect lookup method based on DNS technologies. In this manner, an by simply using existing DNS implementations, ENUM allows retrieving lists of IP based services, such as SIP (Session Initiation Protocol) identifiers for VoIP applications, e-mail addresses, Web pages, etc., associated to the principal of an E.164 telephone number. ENUM uses a particular type of DNS records, called Naming Authority Pointer (NAPTR) [9]. Instead of resolving host or service names into IP addresses, the ENUM service translates E.164 telephone numbers into Uniform Resource Locators (URLs) embedded within NAPTR records. At long term, ENUM is expected to become a decentralized alternative to the E.164 system. For a more detailed introduction to the suite of protocols associated with ENUM, we refer the reader to [6].

As a matter of fact, ENUM is just a simple convention for the translation of E.164 telephone numbers, such as +1-012345678, into URI (*Uniform Resource Identifier*) strings. These strings are associated to the DNS system by using the following convention: (1) special symbols like '+' and '-' are deleted (e.g., +1-012345678 becomes 1012345678); (2) the resulting string of digits is inverted from left to right (e.g., 8765432101); (3) a symbol '.' is inserted between each two digits (e.g., 8.7.6.5.4.3.2.1.0.4.1); (4) the domain name .e164.arpa (registered by the IETF for ENUM resolution) is finally concatenated to the previous string (e.g., 8.7.6.5.4.3.2.1.0.1.e164.arpa). The resulting string of characters and digits is then ready to be used as a normal query towards the DNS system. At the server side, the URI associated to every possible telephone number registered by ENUM is stored together with information about its principal (e.g., owners or users of those telephone numbers). Such an information is stored on DNS records of type NAPTR. The internal structure of these records offers to ENUM enough storage space and flexibility for managing complex information (e.g., use of regular expressions).

Let us show in the following a complete example in which ENUM is used for the translation of the telephone number +1-012345678 associated to a user U_1. Let us assume that a user U_2 wants to get in contact with user U_1. First of all, user U_2 translates the previous telephone number into the string 8.7.6.5.4.3.2.1.0.1.e164.arpa. U_2 then uses the obtained URI to construct a DNS query of type NAPTR by using the command line tool *dig*:

```
dig @$NS -t NAPTR 8.7.6.5.4.3.2.1.0.1.e164.arpa
```

As a result, U_2 obtains the following information:

Order	Pref.	Flags	Service	Regexp.	Replacement
100	10	u	sip+E2U	!^.*$!sip:u1@sip.com!'	.
101	10	u	mailto+E2U	!^.*$!mailto:u1@mail.com!	.
102	10	u	http+E2U	!^.*$!http://www.u1.com!	.
103	10	u	tel+E2U	!^.*$!tel:+1-012355567!	.

Let us analyze the response returned by *dig*. As we introduced above, NAPTR records support the use of regular expression pattern matching [9]. In case a series of regular expressions from distinct NAPTR records need to be applied consecutively to an input, the field *Order* is used. The value given in the first line, set to 100, indicates that from the four results of the query, the service *SIP* has the highest priority. In case of having more that one record with the same *order* values, the following field, i.e., *Pref.*, decides which information must be used first. The field *Flag* given for each line, and set to the value *u*, indicates that the field *Regexp.* associated with every record contains the URI associated to the requested E.164 telephone number. A field *Replacement* containing the operator '.' indicates to the ENUM client of user U_2 that the final URL is indeed the string placed between the markers *'!^.*$!'* and *'!'* of the expression contained within the field *Regexp*. The field *Service* indicates the kind of IP service that can be found in the resulting URL. For example, the field *Service* associated with the first line indicates that the resulting service is based on the SIP protocol [13]. The other three options returned as a result of the query are (1) an e-mail address associated with user U_1, (2) his personal Web page, and (3) the use of an additional E.164 telephone number.

Let us notice from our example that the ENUM service does not resolve the IP addresses associated to the URLs embedded within the NAPTR records. A DNS query of type 'A' must follow after an ENUM resolution with the objective of resolving the appropriate IP address that will eventually be used to contact the final service. In our example, and given the values of the field *Order* discussed above, user U_2 contacts again the DNS server in order to obtain the IP address associated to the SIP at `sip.u1.com` to request the connection to user U_1 (i.e., `u1@sip.u1.com`).

1.2 Threats to the ENUM Service

The use of the DNS protocol as the underlying mechanism of the ENUM service leads to security and privacy implications. The exploitation of well known vulnerabilities of DNS-based procedures is a clear way of attacking the ENUM service. A recent analysis of critical threats to ENUM may be found in [19, 20]. Rossebø et al. present in these works their risk assessment analysis of the ENUM service based on a methodology proposed by the European Telecommunications Standards Institute (ETSI) [5]. Both threats and vulnerabilities reported in these works are indeed an heritage of the vulnerabilities existing in DNS mechanisms. We can find in [2] a complete analysis of threats to DNS technologies. The most important threats to DNS technologies can be grouped as follows: (1) authenticity and integrity threats to the trustworthy communication between resolvers and servers; (2) availability threats by means of already existing denial of service attacks; (3) escalation of privilege due to software vulnerabilities in server implementations. Moreover, the DNS protocol uses clear text operations, which means that either a passive attack, such as eavesdropping, or an active attack, such as man-in-the-middle, can be carried out by unauthorized users to capture queries and responses. Although this can be considered as acceptable for the resolution of host names on Web services, an associated loss of privacy when using DNS for the resolution of ENUM queries is reported in [19, 20] as a critical threat.

We consider that the loss of privacy in ENUM queries is an important concern. Beyond the engineering advance that the ENUM service supposes, it is worth considering the consequences that the exposure of people's information may suppose. The achievement of such information by dishonest parties exploiting flaws and weaknesses in the service itself or its underlying protocols must be avoided. We can consider, for instance, worst case scenarios where dishonest servers associated to unscrupulous service providers start keeping statistics of ENUM queries and building people's profiles based on their communication patterns [23]. These scenarios may lead to further violations, such as spam, scams, untruthful marketing, etc. Consumers must be ensured that these activities are not possible [7]. However, current DNS query methods used by ENUM can be exploited if the whole process is not handled by appropriate countermeasures.

1.3 Privacy Weakness in the DNS Protocol

When the DNS protocol was designed, it was not intended to guarantee privacy to people's queries. This makes sense if we consider that DNS is conceived as a distributed hierarchical database which information must be accessed publicly. In scenarios where the DNS protocol is used for the mapping of host and domain names towards traditional Internet services, the inference of information by observing queries and responses can fairly be seen as acceptable — from the point of view of people's privacy. Nevertheless, the use of the DNS protocol on new lookup services, such as the ENUM suite of protocols, clearly introduces a new dimension. Vulnerabilities on the DNS, allowing the disclosure of data associated with people's information, such as their telephone numbers, is a critical threat [19, 20]. Let us summarize these privacy weaknesses from the following three different scopes: (1) DNS local resolvers, (2) communication channel, and (3) remote DNS servers.

On the first hand, Zhao et al. identify in [23] some privacy threats related with local malware targeting the client. Applications such as keyloggers, trojans, rootkits and so on can be considered as a way to obtain the relation between DNS queries and the client who launches them. Let us note that our work does not address the specific case of malware targeting the privacy of the DNS service at the client side. On the second hand, we can identify two main threats targeting the communication channel: (1) passive eavesdroping and (2) active attacks against the network traffic. In the first case, the eavesdroping of plaintext DNS traffic flowing across unprotected wired or wireless LAN networks can be used as a form of anonymity violation. In the second case, traffic injection can also be used to attack the privacy. These attacks can be used to redirect the traffic to a malicious computer, such as ARP spoofing, ICMP redirect, DHCP spoofing, port stealing, etc. Thus, an attacker can redirect every query to a malicious DNS server with the objective of impersonating the correct one and, as a result, to compromise the client privacy. On the third hand, the existence of dishonest or malicious servers can also reduce the level of privacy. Indeed, the DNS cache model allows intermediate servers to maintain a query-response association during a given period of time. The expiration time of every entry in the cache of a server is based on the IP TTL field of a DNS response — as it is defined in [10]. During this period of time, if a client queries a

cached entry, the response will be served without any additional resolution. Otherwise, after this time has elapsed, the entry is removed from the cache and, if a client requests it again, the server resolves it, caches it, and sends the response to the client.

Under certain conditions, the observation of the TTL field can be used by attackers to infer the relation between a client and a particular query, reducing the level of anonymity. If attackers suspect that a given client has launched a specific query, they can resolve the same query on the server used by the client. After the response has been retrieved by the attackers, they can determine the current cache expiration time provided by the server. If the returned value is the maximum expiration time defined by the authoritative server, the attackers can deduce that the query has not been launched by the client in, at least, a period that equals the maximum cache expiration time. However, if the value is less than the TTL value, the attackers can consider, with a certain level of probability, that this query was made by the client at most at *maximum expiration time* minus *current expiration time*. This strategy can be applied by potential attackers under certain circumstances. First of all, it can only be considered in networks composed by a few number of clients and/or a DNS server that receives few queries by these clients. Otherwise, the probability of a correct correlation between the specific query and a given client must be considered almost zero. Secondly, if the expiration time defined by the authoritative server has a low value, it can lead to a situation where attackers might launch the query after it expires in the DNS cache (previously created by the client).

1.4 Privacy Countermeasures and Security Enhancements

Some initial measures for special DNS-based services, like the ENUM service, have been proposed by the IETF. Some examples are the limitation and the kind of information to be stored by the servers, the necessity of requesting the consent of people and/or institutions, etc. Nonetheless, beyond limiting and granting access to store people's information, no specific mechanisms have been yet proposed in order to preserve the invasion of privacy that a service like ENUM may suppose. The use of anonymity-based infrastructures and anonymizers (e.g., the use of the Tor infrastructure [14], based on *Onion Routing* cryptography [21]) is often seen as a possible solution in order to hide the origin (i.e., the sender) of the queries. However, these infrastructures might not be useful for anonymizing the queries themselves against, for example, insecure channels or dishonest servers [8]. The use of the security extensions for DNS (known as DNSSEC), and proposed by the IETF in the late nineties, cannot address those privacy violations discussed in this section. DNSSEC only addresses at the moment authentication and integrity problems in the DNS. Although it must certainly be seen as an important asset to enhance the security of DNS applications, it requires to be combined with additional measures to cope the kind of violations discussed in this section. Finally, the use of Privacy Information Retrieval (PIR) [18] based approaches can also be seen as a mechanism to handle the private distribution of information on the DNS service [23, 24]. Unfortunately, no specific evaluations or practical results are presented in these works. The processing and communication bandwidth requirements of a PIR approach seem to be impractical for low latency services like the DNS/ENUM [22]. We

consider however that these approaches head into the right direction in order to address the problematic discussed in this section.

Inspired by the approaches proposed by Zhao et al. in [23, 24], we sketch in this paper an alternative model for perturbing DNS queries with random noise. The goal of our model is to prevent privacy violations due to attacks against the communication channel level or the existence of dishonest servers. Our approach addresses and enhances some security deficiencies detected in [23, 24], such as the possibility of response manipulation or range intersections. We also present in this work the evaluation of a first proof-of-concept developed and tested upon GNU/Linux setups. Our implementation combines our approach together with the use of DNSSEC extension to preserve authentication and integrity of queries. Although our experimentations reveal high bandwidth consumption as the main drawback, we consider the results as a prove of the validity of our approach.

1.5 Paper Organization

The remainder of this paper has been organized as follows. Section 2 introduces some related works. Section 3 sketches our proposal. Section 4 overviews the use of the security extensions for DNS (i.e., DNSSEC). Section 5 presents the evaluation results of combining our proposal with the security enhancements offered by DNSSEC. Section 6 closes the paper with some conclusions.

2 Related Work

A first solution to address the privacy concerns discussed in Section 1 is the use of anonymous-based communication infrastructures. The use of strong anonymity infrastructures can suppose however a high increase of the latency of a service like the DNS and the ENUM services. We recall that a communication infrastructure for these services must ensure that the service itself is able to deliver both queries and responses accurately and in a timely fashion. Thus, strong anonymity does not seem to be compatible with this requirement. On the other hand, the use of low latency infrastructures, such as the anonymous infrastructure of the Tor (*The second generation Onion Router*) project [14], based in turn on the *Onion Routing* model [21], is more likely to meet the performance requirements of the DNS/ENUM service. Nevertheless, a solution based on both Tor and *Onion Routing* may only be useful for hiding the origin of the queries. Although by using such proposals senders are indeed able to hide their identities through a network of *proxies*, they do not offer anonymity to the queries themselves. For instance, threats due to the existence of dishonest servers, are not covered by these solutions [8].

The approach presented by Zhao et al. in [23, 24] aims at preserving the anonymity of DNS/ENUM queries from the point of view of the channel and/or the service providers. The main objective of these proposals is the achievement of anonymity by using a PIR

(Privacy Information Retrieval) model [18]. The authors propose devising the communication protocol involved between DNS clients and servers by considering queries as secrets. Instead of querying the server by a specific host name h, for example, Zhao et al. propose in [23] the construction and accomplishment of random sets of host names $[h_1, h_2, \ldots, h_n]$. The resulting protocol aims at avoiding that by listening into the channel or controlling the destination service, an attacker learns nothing about the specific host name h from the random list of names. The main benefit of this proposal is the simplicity of the approach. The main drawback is the increase in communication bandwidth that it may suppose. Zhao et al. extend in [24] this first proposal towards a two-servers PIR model. The objective of the new protocol is to guarantee that DNS clients can resolve a given query, at the same time that they hide it to each one of the servers. Nevertheless, compared with the previous proposal, this approach reduces the bandwidth consumption. The approach requires, however, significant modifications on traditional DNS implementations. We analyze more in detail these two proposals in Section 3.

The proposals presented in [23, 24], as well as Tor, do not offer preservation of authenticity and integrity of DNS responses. Therefore, without other countermeasures, these solutions cannot avoid man-in-the-middle or replay attacks aiming at forging DNS responses. A proper solution for avoiding this problem is to combine the use of anonymity with the integrity and authenticity offered by the security extensions of DNS — often referred in the literature as DNSSEC (cf. Section 4 for more information about DNSSEC). In this manner, we can guarantee the legitimacy of the response while maintaining an acceptable performance. We show in Section 5 that the impact on the latency of the service when using DNSSEC is minimal. We consider that authenticity and integrity threats are hence reduced by combining a proper anonymity model together with DNSSEC. None of these proposals guarantees the confidentiality of the queries. Although the use of alternative techniques such as IPSec [16] could be seen as a complementary solution to protect the exchanges on data between servers and clients of DNS, we consider that they are not appropriate for solving our motivation problem. First of all, the bandwidth and processing time overheads of using IPSec are much higher, and can render the solution impractical [17]. Secondly, IPsec does not offer protection during the caching processes between resolvers and/or intermediate servers. Furthermore, it is quite probable that servers of a global DNS service may not be IPsec capable. We consider that this approach is not an appropriate solution to our problem. Since our motivation is focused on privacy issues rather than confidentiality concerns, we consider that the combination of anonymity preservation together with integrity and authentication aspects offered by DNSSEC are worth enough to conduct our study.

3 Use of Random Ranges to Anonymize DNS Queries

Before going further in this section, let us first recall here the schemes presented by Zhao et al. in [23, 24]. The first approach [23] works as follows: a user U, instead of launching just a single query to the DNS server NS, constructs a set of queries $Q\{H_i\}_{i=1}^{n}$. If we assume DNS queries of type A, the previous range of queries will

include up to n different domain names to be resolved. The query $Q\{H_i\}$ will be the only one that includes the domain name desired by U. All the other queries in $Q\{H_1\} \dots Q\{H_{i-1}\}$ and $Q\{H_{i+1}\} \dots Q\{H_n\}$ are chosen at random from a database DB. The authors claim that this very simple model considerably increases the privacy of user U queries. Indeed, the only information disclosed by user U to third parties (e.g., DNS server NS and possible attackers with either active or passive access to the channel between U and NS) is that the real query $Q\{H_i\}$ is within the interval $[1, n]$. Zhao et al. presume that the probability to successfully predict query $Q\{H_i\}$ requested by user U can be expressed as follows: $P_i = \frac{1}{n}$. We refer the reader to [23] for a more accurate description of the whole proposal.

We consider that the probability model presented in [23] is unfortunately very optimistic. In fact, we believe that the degree of privacy offered by the model can clearly be degraded if we consider active attacks, in which an adversary is capable of interacting with the channel. For example, we consider that the approach does not address possible cases in which the resolution of query $Q\{H_i\}$ fails. If we assume an active attacker manipulating network traffic (e.g., by means of RST attacks, or sending suitable ICMP traffic) to drop query $Q\{H_i\}$ — or its associated response. If so, user U will be forced to restart the process and generate a new range of queries — i.e., requesting once again $Q\{H_i\}$. Depending on how this new range is managed, the degree of privacy estimated by the probabilistic model in [23] will clearly decrease. Let $Q_j\{H_i\}_{i=1}^{n}$ be the $n\text{-}th$ consecutive range exchanged for the resolution of query $Q\{H_i\}$, the probability of success for an attacker trying to guess $Q\{H_i\}$ must then be defined as follows:

$$P_{ij} = \frac{1}{|Q_1\{H_i\}_{i=1}^{n} \cap Q_2\{H_i\}_{i=1}^{n} \cap \dots \cap Q_j\{H_i\}_{i=1}^{n}|}$$

Zhao et al. present in [24] a second approach intended to reduce the bandwidth consumption imposed by the previous model. The new approach gets inspiration from Privacy Information Retrieval (PIR) approaches [3]. It relies indeed on the construction of two ranges $Q_1\{H_i\}_{i=1}^{n}$ and $Q_2\{H_i\}_{i=1}^{n+1}$, where $H_{n+1} \in Q_2$ is the true query defined by user U. Once defined Q_1 and Q_2, such ranges are sent to two independent server NS_1 and NS_2. Assuming the resolution of DNS queries of type A, each server resolves every query associated with its range, obtaining all the associated IP adresses (defined in [24] as X_i) associated to the query H_i. NS_1 computes $R_1 = \sum_{i=1}^{n} \otimes X_i$ and NS_2 computes $R_2 = \sum_{i=1}^{n+1} \otimes X_i$. Both R_1 and R_2 are sent to user U, who obtains the resolution associated to H_{n+1} using the expression $X_{n+1} = R_1 \otimes R_2$. As we can observe, the bandwidth consumption of this new approach is considerably smaller than the one in [23], since only two responses (instead of n) are exchanged.

The main benefit of this last proposal, beyond the reduction of bandwidth consumption, is its achievement on preserving the privacy of the queries from attacks at the server side. However, it presents an important drawback due to the necessity of modifying DNS protocol and associated tools. Let us note that the proposal modifies the mechanisms for both querying the servers and responding to the clients. Moreover, it still presents security deficiencies that can be violated by means of active attacks against the communication channel between resolvers and servers. Indeed, attackers control-

ling the channel can still intercept both range Q_1 and Q_2. If so, they can easily obtain the true query established by user U by simply applying $Q_1 \setminus Q_2 = H_{n+1}$. Similarly, if attackers successfully intercept both R_1 and R_2 coming from servers NS_1 and NS_2, they can obtain the corresponding mapping address by performing the same computation expected to be used by user U, i.e., by computing $X_{n+1} = R_1 \otimes R_2$. Once obtain such a value, they can simply infer the original query defined by user U by requesting a reverse DNS mapping of X_{n+1}. Analogously, an active control of the channel can lead attackers to forge resolutions. Indeed, without any additional measures, a legitimate user does not have non-existence proofs to corroborate query failures. This especially relevant on UDP-based lookup services, like the DNS, where delivery of messages is not guaranteed. Attacker can satisfactorily apply these kind of attacks by intercepting, at least, one of the server responses. An attacker can for example intercept R_1, compute $R_2^* = R_1 \otimes R_3$ (where R_3 is a malicious resolution), and finally send as a resulting response coming from server NS_2. Then, the resolver associated to user U will resolve the mapping address as follows: $R_1 \otimes R_2^* = R_1 \otimes R_1 \otimes R_3 = R_3$.

As an alternative to the proposals presented in [23, 24], we propose to distribute the load of the set of ranges launched by user U among several servers $NS_1 \dots NS_m$. Unlike the previous schemes, our approach aims at constructing different ranges of queries for every server $NS_1 \dots NS_m$. The ranges will be distributed from $Q\{H_1^{NS_1}\} \dots Q\{H_{\frac{n}{m}}^{NS_1}\}$ to $Q\{H_1^{NS_m}\} \dots Q\{H_{\frac{n}{m}}^{NS_m}\}$. When the responses associated to these queries are obtained from the set of servers, user U verifies that the desired query has been successfully processed. If so, the rest of information is simply discarded. On the contrary, if the query is not processed, i.e., user U does not receive the corresponding response, a new set of ranges is generated and proposed to the set of servers. To avoid the inference attack discussed above, ranges are constructed on independent sessions to preserve information leakage of the legitimate query. Let us note that by using this strategy, we preserve privacy of queries from both server and communication channel. In order to guarantee integrity of queries, authenticity of queries, and non-existence proofs, our proposal relies moreover on the use of the DNS security extensions. We survey the use of DNSSEC in the following section. An evaluation of our approach is presented in Section 5.

4 The DNSSEC Specifications

The Domain Name System SECurity (DNSSEC) extension is a set of specifications of the IETF for guaranteeing authenticity and integrity of DNS Resource Records (RRs) such as NAPTR records. DNSSEC is based on the use of asymmetric cryptography and digital signatures. DNSSEC is often criticized for not being yet deployed after more than ten years of discussions and revisions. It is however the best available solution (when used properly) to mitigate active attacks against the DNS, such as man-in-the-middle and cache poisoning. DNSSEC only addresses threats on the authenticity and integrity of the service. Although early DNSSEC proposals presented clear problems of management associated with its key handling schema, the latest established version

of DNSSEC overcomes key management issues based on the Delegation Signer (DS) model proposed in RFCs 3658 and 3755. DNSSEC is being currently deployed on experimental zones, such as Sweden, Puerto Rico, Bulgaria, and Mexico (cf. *http://www.xelerance.com/dnssec/*). At the moment of writing this paper, more than ten thousand DNSSEC zones are enabled (cf. http://secspider.cs.ucla.edu/). Deployment at the root level of DNS is currently being debated, although the difficulties of deploying DNSSEC at this level seem to be of political nature rather than technical issues.

The main characteristics of the latest version of DNSSEC are described in RFCs 3658, 3755, 4033, 4034, and 4035. An analysis of threats addressed and handled by DNSSEC is also available in RFC 3833. DNSSEC provides to DNS resolvers origin authentication of Resource Records (RRs) (such as A, CNAME, MX, and NAPTR), as well as RR integrity and authenticated denial of existence (e.g., if a NAPTR record is queried in the global DNS service and it does not exist, a signed proof of non-existence is returned to the resolver). As we pointed out above, DNSSEC allows two different strategies to guarantee authenticity and integrity. On the one hand, administrators of a given domain zone can digitally sign their zones by employing their own private key and making available to resolvers the corresponding public key. On the other hand, administrators can rely on the use of a chain of trust between parent and child zones that enables resolvers to verify when the responses received from a given query are trustworthy. In order to implement these two strategies, DNSSEC relies on the use of four new DNS RR types: (1) Resource Record Signature (RRSIG) RRs that store the signature associated to every RR in a given zone, (2) DNS Public Key (DNSKEY) RR that contains the specific public key that will allow the resolver to validate the digital signatures of each RR, (3) Delegation Signer (DS) RRs that are added in parent zones to allow delegation functions on child zones, and (4) Next Secure (NSEC) RRs that contain information about the next record in the zone, and that allow the mechanism for verifying the nonexistence of RRs on a given zone. DNSSEC includes two bit flags unused on DNS message's headers to indicate (1) that the resolver accepts unauthenticated data from the server and (2) that those RRs included in the response were previously authenticated by the server.

Regarding the set of keys for signing RRs, one or two key pairs must be generated. If administrators decide to sign zones without a chain of trust, the complete set of RRs of each zone are signed by using a single pair of Zone Signing Keys (ZSKs). On the other hand, if the administrators decide to use a chain of trust between parent and child zones, two key pairs must be generated: a pair of Key Signing Keys (KSKs) is generated to sign the top level DNSKEY RRs of each zone; and a pair of ZSKs keys are used to sign all the RRs of each zone. Several algorithms can be used for the generation of key pairs, such as RSA, DSA (Digital Signature Algorithm), and ECC (Elliptic Curve Cryptosystem). These keys are only used for signatures, and not for encryption of the information. Signatures are hashed by using MD5 or SHA1, being the combination RSA/SHA1 the mandatory signature process that must be implemented at servers and resolvers. The type and length of these keys must be chosen carefully since it significantly affects the size of the response packets as well as the computational load on the server and the response latency. Results in [1] pointed out to an overhead of 3% up to 12% for

KSK/ZSK keys based on RSA and length of 1200/1024 bits; and 2% up to 6% for ECC based keys of length 144/136 bits.

The validity period associated with KSK/ZSK keys must also be defined carefully in order to avoid problems with key rollovers, since data signed with previous keys may still be alive in intermediary caches. Synchronization parameters are therefore very important in DNSSEC. Another issue, often referred in the literature as *zone enumeration* or *zone walking*, relies on the use of the NSEC RR. As we pointed out above, NSEC allows chaining the complete set of RRs of a zone to guarantee nonexistence of records and so, it also allows retrieving all the information associated to a given zone. Although the DNSSEC working group originally stated that this is not a real problem (since, by definition, DNS data is or should be public) they proposed an alternative method that uses a new RR called NSEC3 which prevents trivial zone enumeration to introduce a signed hash of the following record instead of including directly its name. Secure storage of trust anchors has also been actively discussed in the literature. Unlike PKI solutions, the chain of trust of DNSSEC offers higher benefits compared to the security of X.509 certificates since the number of keys to protect in DNSSEC is much lower.

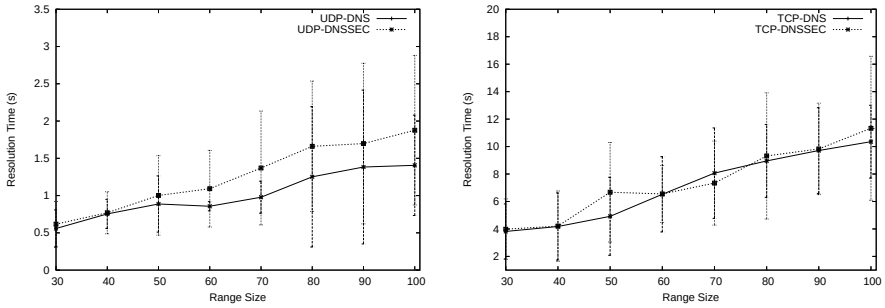
5 Evaluation of Our Proposal

This section shows the outcome of our evaluation steered towards measuring the latency penalty due to the use of our approach on a real network scenario for the resolution of DNS and DNSSEC queries. The hardware setup of our experimental scenario is the following. A host R, running on an Intel Core 2 Duo 2 GHz and 1 GB of memory, performs queries of type NAPTR to a global resolution service G. The implementation and deployment of our proposal in R is based on the *Python* language. More specifically, we base our implementation on the module *dnspython* [11] for the construction and resolution of DNS queries; and the module *m2crypto* [12] to access the *OpenSSL* library [4] for the verification of digital signatures defined by DNSSEC.

The global resolution service G is in turn implemented by means of three different hosts: S_1, that runs on an AMD Duron 1 GHz with 256 MB of memory; S_2, that runs on an Intel PIII 1 GHz with 512 MB of memory; and S_3, that runs on an Intel Xeon 2.4 GHz with 1 GB of memory. Servers in G are located on different networks and on different countries: server S_1 is located in North America; and servers S_2 and S_3 are located in Europe. DNS and DNSSEC services configured on each one of these hosts are based on BIND 9.4.2 (cf. `http://www.isc.org/products/BIND/`). The configuration of each server in G consists of a database $\mathcal{N}$ that contains more than twenty thousand NAPTR records generated at random. Each one of these records are linked moreover with appropriate DNSSEC signatures. We use for this purpose the *dnssec-keygen* and *dnssec-signzone* tools that come with BIND 9.4.2. The key sizes are 1200 bits for the generation of Key Signing Keys (KSKs) and 1024 bits for Zone Signing Keys (ZSKs). The generation of keys is based on the RSA implementation of *dnssec-keygen*. Although the use of ECC signatures seems to reduce the storage space

of signed zones [1], the algorithm we use is RSA instead of ECC since the latter is not yet implemented in BIND 9.4.2.

We measured in our evaluations the time required for resolving queries from R to G with different testbeds, where the size of the query range of each testbed increments from thirty to more than one hundred. Each testbed consists indeed on the generation of three sets of random queries, one for each $S_i \in G$. Each testbed is launched multiple times towards cumulative series of NAPTR queries. Each series is created at random during the execution of the first testbed, but persistently stored. It is then loaded into the rest of testbeds to allow comparison of results. We split our whole evaluation in four different stages. During the first two stages, the transport layer utilized between R and G is based on the TCP protocol. First stage is used for the resolution of DNS queries, while stage two is used to resolve DNSSEC queries. Similarly, stage three and four are based on UDP traffic for the resolution of, respectively, DNS and DNSSEC queries. During these two last experiments based on DNSSEC, R verifies the integrity and the authenticity of the queries received from the different servers in G. The verification procedures have been implemented as defined in DNSSEC RFCs (cf. Section 4). We show in Figure 1 the results that we obtained during the execution of these four experiments.



(a) UDP tests

(b) TCP tests

Fig. 1. Evaluation of our proposal.

We can appreciate by looking at Figure 1 that the latency increases linearly with the size of the range of queries. TCP-based experiments show worst performance than UDP-based queries — due to the overhead imposed by the establishment of sessions. UDP protocol is clearly the best choice for the deployment of our proposal. Given an acceptable latency of no more than two seconds, UDP results show that the probability of guessing the true query is $P_i = \frac{1}{3 \cdot 80} = \frac{1}{240} \simeq 0.004167$. We consider this result as satisfactory. In general terms, we should expect that the certainty for obtaining a query i within a range of size n and m different servers is $P_i = \frac{1}{n \cdot m}$.

Besides the difficulties imposed by our model for predicting the original petition, we are conscious of the high bandwidth increase that it represents. This is an important drawback in scenarios where the bandwidth consumption is a critical factor. However, if this is the case, it is possible to reduce the size of the range of queries. Since there is a clear relation between both parameters, i.e., the bandwidth consumption is inversely proportional to the prediction probability, we believe that a proper balance between bandwidth consumption and prediction probability can be enough to enhance the privacy of the service. Let us recall that reducing the size of each range of queries to a fifty per cent, the prediction probability for the attacker is proportionally increased by two. On the other hand, let us observe how the penalty in the response times introduced by DNSSEC is not specially significant, solving the integrity and authenticity problems that appeared in the other approaches. This is the reason why we consider the activation of DNSSEC as a decisive factor for avoiding manipulation network traffic attacks.

6 Conclusion

The use of the DNS (*Domain Name System*) as the underlying technology of new lookup services might have unwanted consequences in their security and privacy. We have analyzed in the first part of this paper privacy issues regarding the use of DNS procedures in the ENUM (*tElephone NUmber Mapping*) service. The loss of privacy due to the lack of security mechanisms of the DNS data in transit over insecure channels or with dishonest servers is, from our point of view, the main peculiarity of the threat model associated to the ENUM service — compared with the threat model of traditional DNS applications. We have then analyzed in the second part of our work, the use of statistical noise and the construction of range of queries as a possible countermeasure to reduce the risk associated to this threat.

The implementation of our proposal is inspired on a PIR (*Privacy Information Retrieval*) model introducing random noise in the DNS queries. The goal of our model is to reduce privacy threats at both channel (e.g., eavesdroppers trying to infer sensible information from people's queries) and server level (e.g., dishonest servers from silently recording people's queries or habits). The proposal is indeed inspired on two previous works surveyed in Section 3. Security deficiencies detected in both contributions have been addressed, such as response manipulation and range intersections. The combination of our model with the use of DNSSEC allows us to prevent, moreover, from authenticity and integrity threats. The main drawback of our contribution is still a high increase on the bandwidth consumption of the service. We are working on an improvement of our model to address this limitation.

Acknowledgments — The authors graciously acknowledge the financial support received from the following organizations: Spanish Ministry of Science and Education (projects *CONSOLIDER CSD2007-00004 "ARES"* and *TSI2006-03481*).

References

1. Ager, B., Dreger, H., and Feldmann, A. Predicting the DNSSEC Overhead Using DNS Traces. *40th Annual Conf. on Information Sciences and Systems*, pp. 1484–1489, 2006.
2. Atkins, D. and Austein, R. Threats Analysis of the Domain Name System (DNS). *Request for Comments, RFC 3833*, IETF, 2004.
3. Chor, B., Kushilevitz, E., Goldreich, O., and Sudan, M. Private Information Retrieval *In: Journal of the ACM*, pp. 965–981, New York, USA, 1998
4. Eric A. Young, and Tim J. Hudson OpenSSL: The Open Source Toolkit for SSL/TLS http://www.openssl.org/
5. ETSI, Methods and Protocols for Security; part 1: Threat analysis. Technical Specification ETSI TS 102 165-1 V4.1.1, 2003.
6. Faltstrom, P. and Mealling, M. The E.164 to Uniform Resource Identifiers Dynamic Delegation Discovery System Application. *Request for Comments, RFC 3761*, IETF, 2004.
7. Federal Trade Commission. Protecting Consumers from Spam, Spyware, and Fraud. A Legislative Recommendation to Congress, 2005.
8. Garcia-Alfaro, J., Barbeau, M., Kranakis, E. Evaluation of Anonymized ONS Queries. 1rst Workshop on Security of Autonomous and Spontaneous Networks (SETOP 2008). Loctudy, Brittany, France, October, 2008.
9. Mealling, M. and Daniel, R. The Naming Authority Pointer (NAPTR) DNS Resource Record. *Request for Comments, RFC 2915*, IETF, 2000.
10. Mockapetris, P. Domain Names - Implementation and Specification. *Request for Comments, RFC 1035*, IETF, 1987.
11. Nomium Inc. A DNS Toolkit for Python http://www.dnspython.org/
12. Siong, N. P. and Toivonen, H. Mee Too Crypto http://chandlerproject.org/bin/view/Projects/MeTooCrypto.
13. Rosenberg J., et al. Session Initiation Protocol. *Request for Comments, RFC 3261*, 2002.
14. Dingledine, R., Mathewson, N., and Syverson, P. F. Tor: The second-generation Onion Router. *In: 13th conference on USENIX Security Symposium*, 2004.
15. DNSSEC Deployment Initiative. Available from: http://dnssec-deployment.org/
16. IETF IPsec. Available from: http://www.ietf.org/ids.by.wg/ipsec.html
17. Meenakshi, S.P. and Raghavan, S.V. Impact of IPSec Overhead on Web Application Servers. *Advanced Computing and Communications (ADCOM2006)*, pp. 652–657, 2006.
18. Ostrovsky, R. and Skeith, W.E. A Survey of Single Database PIR: Techniques and Applications. *Proceedings of Public Key Cryptography (PKC-2007)*, 2007.
19. Rossebø, J., Cadzow, S., and Sijben, P. eTVRA, a Threat, Vulnerability and Risk Assessment Method and Tool for eEurope. *In: 2nd Int'l Conf. on Availability, Reliability and Security, ARES 2007, pp. 925–933, Vienna, Austria, 2007*.
20. Rossebø, J., Cadzow, S., and Sijben, P. *e*TVRA, a Threat, Vulnerability and Risk Assessment Tool for *e*Europe. *In: 4th Int'l Conf. on Trust Management (iTrust 2006), Springer, LNCS 3986, pp. 467–471, Pisa, Italy, 2006*.
21. Reed, M. G., Syverson, P. F., and Goldschlag, D. M. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, 1998.
22. Sion, R. and Carbunar, B. On the Computational Practicality of Private Information Retrieval. *Network and Distributed Systems Security Symposium (NDSS)*, 2007.
23. Zhao, F., Hori, Y., and Sakurai, K. Analysis of Privacy Disclosure in DNS Query. *In: IEEE Int'l Conf. on Multimedia and Ubiquitous Engineering*, pp. 952–957, 2007.
24. Zhao, F., Hori, Y., and Sakurai, K. Two-Servers PIR Based DNS Query Scheme with Privacy-Preserving. *In: IEEE Int'l Conf. on Intelligent Pervasive Computing*, pp. 299–302, 2007.