社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－DPS－130 (8)
2007－CSEC－ 36 (8)
2007／3／1

PIRに基づくプライバシー保護を考慮した新しいDNSクエリフレームワーク

趙 方明† 堀 良彰‡ 櫻井 幸一‡

†九州大学大学院システム情報科学府
819-0395 福岡市西区元岡 744
zhao@itslab.csce.kyushu-u.ac.jp

‡九州大学大学院システム情報科学研究院
819-0395 福岡市西区元岡 744
{hori,sakurai}@csce.kyushu-u.ac.jp

あらまし 現在、プライバシーを侵害しつつある社会では、DNSクエリのプライバシー侵害は考慮する価値がある重要な問題である。本稿では、まず現在使用されている完全なDNSクエリプロセスを分析する。さらに、DNSクエリプロセスの三つのステップ、すなわち、クライアント側、クエリ通信プロセス、DNSサーバ側、でのプライバシー侵害の問題について論じる。これらの問題を解決するために、既知のクエリ方式を用いて、理論上でのプライバシー保護が証明された新しいクエリフレームワークを提案する。

A New Privacy-Preserving DNS Query Framework Based on PIR

Fangming Zhao† Yoshiaki Hori‡ Kouichi Sakurai‡

†‡Department of Computer Science and Communication Engineering, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka 819-0395 Japan
zhao@itslab.csce.kyushu-u.ac.jp {hori,sakurai}@csce.kyushu-u.ac.jp

Abstract In the society preoccupied with gradual erosion of (electronic) privacy, loss of privacy in current DNS query is an important issue worth considering. We firstly analyzed the complete DNS query process now in use; then, from each step of the DNS query process, we discussed the privacy disclosure problem in each step of the query: Client side, Query transmission process and DNS server side. For solving these problems, we propose a new DNS query framework with a well-known query scheme: Private Information Retrieval, which was proved to achieve theoretic privacy.

Keyword: DNS Query, Privacy-Preserving, Private Information Retrieval(PIR)

1 Introduction and Motivation

With the development of automatic information processing, it is necessary to consider privacy protection in relation to personal information. Moreover, the surveillance potential of powerful computer systems demands for rules governing the collection and sharing of personal information. An overview of the evolution of data protection is presented [1]. Expression of data protection in various declarations and laws vary. However, all require that personal data must be kept secure. Thus, information systems must take responsibility for the data they manage. Therefore the main challenge in data privacy is to share some data while protecting personal data.

However, an issue: "DNS Query Privacy" was ignored by those laws and declarations. Nowadays, when surfing on the Internet, DNS query is the most frequently used function. Moreover,

while enjoying the DNS query service, do you totally trust that service can securely protect your privacy? Most Internet users always don't know that is there anyone watching, peeking, or eavesdropping "what kind of websites I am attending to access?"

The primary motivation for this paper is current lack of privacy in DNS query. By reviewing the whole process of the DNS query, we analyse those privacy threats. Then, for avoiding these threats, we construct a simple and flexible privacy-preserving component based on a well-known client-to-server query scheme: Private Information Retrieval (PIR)[8].

Our new DNS query framework could provide users a privacy-preserving query scheme, by our scheme, users could protect their personal privacy to a new level. What's more, we believe that our research could make average users aware of both the need for effective DNS query and the need to protect their privacy.

The rest of the paper is organized as follows: first, we give a careful review of information privacy and of the whole DNS query process in order to show that there is little privacy protection in the query process and to show the importance of privacy protection in the DNS query. In section 4, we introduce One-Server based and Two-Server based query models: PIR. In section 5, by reviewing a One-Server PIR based query model(Range Query[10]), we present some prons and cons of that scheme and for the Cons, we propose a Two-Server PIR based DNS Query model, moreover, we also give a comparison of two schemes. Finally, we list the remaining issues and give further research directions.

2 DNS Query Process

DNS stands for Domain Name System. The DNS is used on the Internet to make correspondence between IP address and readable names. The part of the system sending the queries is called the resolver and is on the client side of the configuration. The name server answers the queries. Think of a DNS query as a client asking a server a two-part question, such as "Do you have any IP address for a computer named hostname.example.com.?" (Fig.1) When the client receives an answer from the server, the request is then passed to the DNS client service for resolution using locally cached information. If the queried name can be resolved, the query is answered and the process is completed. Else, the resolution process continues with the client querying a DNS server to resolve the name. When the DNS server receives a query, if the queried name matches a corresponding resource record in local zone information, the server answers it authoritatively. Else, the server then checks to see if it can resolve the name using locally cached information from previous queries. If a match is found here, the server answers with this information. Again, if the preferred server can answer with a positive matched response from its cache to the requesting client, the query is completed.

If the queried name does not find a matched answer at its preferred server-either from its cache or zone information, the query process can continue. This involves assistance from other DNS servers to help resolve the name. By default, the DNS client service asks the server to use a process of recursion to fully resolve names on behalf of the client before returning an answer. In most cases, the DNS server is configured by default to support the recursion process.

From the analysis of the DNS query process, we can clearly get the result: the client's privacy could be leaked in the whole query process. A client only query one specific hostname(target) on the Internet, and the target hostname and the returned IP address may be passed through several servers and resolvers. If anyone of those third parties (agents, servers, responders or distribution points) is compromised, then we say that our privacy was disclosed to others illegally. In the following discussions, we will introduce our analyses of several privacy attacks to DNS servers and DNS queries.

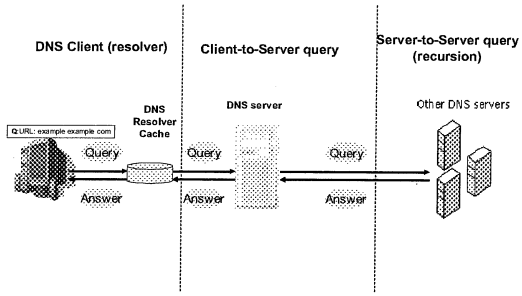

Figure 1: The complete DNS query process

3 Privacy Threat Analysis

Based on the analyses to the whole DNS query process, in this section, we analyze the privacy disclosure of each step of the query process: client side, data transmission between clients and widespread DNS servers, DNS servers side.

3.1 Privacy disclosure at the client side

The privacy disclosure of DNS query at the client side has been recognized for a long time. This threat does not like the form of direct attacks by viruses or hackers, but rather the form of monitoring programs surreptitiously installed on computers. These monitoring applications are called spyware, and serve to record and transmit a user's computer uses and behaviors to third parties. They are normally not directly malicious as the secret spywares, but they do send out information from user's computer to a third party. Most commonly some kind of habit tracing or statistics of surfing or similar. It has been estimated that, excluding cookies, almost 70 percent of consumers ' computers contain some form of spyware [4].

Spywares differ from virus and worms in that they do not usually self-replicate. Like many recent viruses, we found that spyware infected computers for commercial gain. Moreover, it has been found that toolbars from any other then the big players like Google, Yahoo, MSN and similar very often contains spyware to some degree today. Several US states have enacted anti-spyware legislation but currently, without being creative and applying anti-hacking laws to spyware, there are no federal laws. Internationally, laws have been and will continue to be enacted.

In current condition: with the lack of specific laws about the E-Privacy and without a perfect secure enough anti-spy software, the best protection for client is: Do not dowload peer-to-peer application bundles; do not install suspicious software, even if your are familiar with it; and finally, which is the most important, integrate a spyware monitoring and sweeping program just as you have become accustomed to do with anti-virus measures.

3.2 Privacy disclosure during the query transmission

Nowadays, based on cryptographic techniques, such as Transport Layer Security, digital signature and authentication techniques, some of data transportation problems can be easily remedied at the technical level, however, at the same time, we have to concede that there are still some privacy-disclosure problem which users and many institutions ignored. Here, we discuss the most often privacy-referred DNS query attacks: Eavesdropping and Man In The Middle(MITM).

Eavesdropping attacks are enabled by the use of shared media in networks. In an eavesdropping attack, the attacker configures the respective network interface in promiscuous mode. In this mode, the attacker's computer receives any packets sent on the network, including packets destined to other nodes. If packets are unencrypted, the attacker can read packets' data, possibly including passwords and other credentials. Many easily available applications can be used for eavesdropping, including *tcpdump* [5] and *wireshark* [6].

A secure version of DNS, DNSSEC[7], uses cryptographic electronic signatures signed with a trusted digital certificate to determine the authenticity of

data. DNSSEC is an Internet standard that extends the DNS technology for name look-ups. What DNSSEC adds is primarily more secure name look-ups and reduced risk for manipulation of information and forged domain names. However, when related to the domain name query, it is similar to the DNS, it still query only 1 unencrypted name at 1 time. So what DNSSEC Provides are only: authenticity and integrity, without privacy protection. Moreover, as of 2007 DNSSEC is not yet widely deployed.

3.3 Privacy disclosure at the DNS server side

In this part, we discuss the threat to DNS query privacy disclosure at the DNS servers side. Here, we would like to focus on a "not illegal" DNS query privacy dislosure problem.

Suppose one of the DNS servers is interested in aggregate, statistically signaficant, properties of his clients. For example some companies wants to construct an aggregate model of its employees' web access interests, or want to statistic each employee's web-suring activity during their free time. However, many people are becoming increasingly concered about the privacy of their personal data. They would like to avoid giving out much more about themselves than is required to be aggregated by the local DNS service. In this situation, how can we protect the dislosure of our private data? Since until now, there is no specific protection aims at this kind of privacy protection problem.

After giving a carefully threats analysis in this section, in the next section of this paper, we would introduce a famous query mode–PIR. And we construct a new query model based on this model. Since the perfect privacy is very intractable [2], our new query model aims at maximally decreasing the DNS query privacy disclosure.

4 Private Information Retrieval

The notion of private information retrieval PIR was introduced by Chor, Goldreich, Kushilevitz and Sudan [8] and has already received a lot of attention. The study of PIR is motivated by the growing concern about the user's privacy when querying a large commercial database. The problem was independently studied by Cooper and Birman [9] to implement an anonymous messaging service for mobile users. Next, we will introduce the One-Server and Two-Server PIR, and then we will present our framework's prototype.

4.1 One-Server PIR Scheme

The One-server PIR problem consists of devising a communication protocol involving just two parties, the database server and the user, each having a secret input. The database's secret input is called the *datastring*, an n-bit string $X = (x_1, x_2, ..., x_n)$. The user's secret input is an integer i between 1 and n. The protocol should enable the user to learn x_i in a communication-efficient way and at the same time hide i from the database(the trivial and inefficient solution is having the database send the entire string B to the user), the communication in this scheme is n. For more details about it, please read the paper [8].

4.2 Two-Server PIR Scheme

In this section, we introduce a two-server PIR scheme that allows user to privately obtain the bit x_i by receiving a single *bit* from each of two servers. The user uniformly selects a random set $S \subseteq [n]$ (i.e. each index $j \in [n]$ is selected with probability 1/2). The user sends S to Server1 and $S \oplus i$ to Server2. Each server, upon receipt of the message $I \subseteq [n]$, replies with a single bit which is the XOR of the bits with indices in I). The user XOR the answers it has received, thus retrieving the desired bit x_i. Clearly, none of the servers has obtained any information regarding which index was desired by the user (as each of the servers obtains a uniformly distributed subset of [n]), the communication in this scheme is $n^{1/3}$.

5 Discussion on PIR and DNS Query

Based on the PIR introduction, in this section, we analyze a new DNS Query model which based on the One-Server PIR, and then we propose a new DNS Query framework which based on the Two-Server PIR, and at last we give a comparation of the two query schemes.

5.1 DNS Query with One-Server PIR

We propose the query model based on One-Server PIR [10], in that scheme, a new query model "Range Query" was proposed: instead of querying by a specific host name, the client queries a range of hostnames [1, n] (n: max index number of the hostname in DNS query). The only information divulged to the server and other third parties is that the target query lies in the interval [1, n] which translates into the probability of correctly guessing i: $P_i = \frac{1}{n}$. Only one 1 hostname is the target, and others are all generated randomly.

5.2 DNS Query with Two-Server PIR

Our new query model is also based on the Range Query: the client queries two hostname range to two separate servers (Fig 2.), $Q_1 = (H_1, H_2, ..., H_m)$: hostname range without target; and $Q_2 = (H_1, ..., H_m, H_{m+1})$: hostname range with target. H_{ave}: Average length of hostname; we assume that common cost does not include DNS query protocol overhead. After received two range of hostnames separatly, servers compute $A_i = \oplus x_{ij}$(i=1, 2, server numbers), then send back to client. By two response from two separate servers, client could retrieval the target IP address $(x_{1hm}, x_{2hm}, ..., x_{nhm})$ $= A_1 \oplus A_2$ (XOR answer of two servers' response).

5.3 Comparison of two schemes

For a perfect privacy situation with One-Server PIR, as everyone knows, the server should send all n bits(the whole DNS Database) to the user.

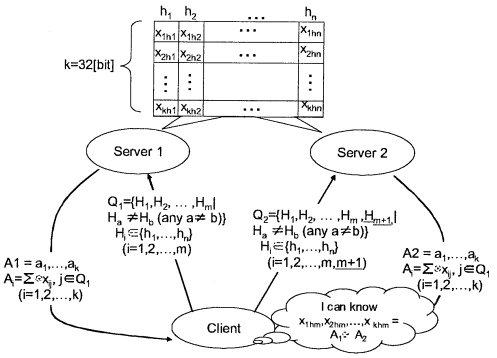

Figure 2: Two-Server PIR based DNS Query

However, this requests a higher hardware-demand, such as huge disks to store such a large amount of data, a larger cache to receive the entire database. So, for common users, it is very intractable.

As we have mentioned, a DNS Query method–Range Query had been proposed [10], In the proposal which based on the One-server PIR, an issue is the bandwidth, when client send queries to server, the communication cost is H_{ave}*$(m + 1)$; so the response is $(m+1)$*32$[bit]$.(H_{ave} means the length of IP address)

After an anylysis of it, we found the pros and cons of this scheme are very obvious, *Pros*: sending m+1(n) queries together, could hide the target in the range; client can protect privacy of their data by perturbing it with a randomization algorithm; for the response information from servers, the returned IP addresses are also a range with equal probability of its memebers. *Cons*: In brief, huge computation at servers' side, and huge communication cost is also a serious problem.

With a Two-Server PIR shceme, we could get a diffrent communication cost. For Queries, it is the sum of two servers: H_{ave}*$(m+1)$+ H_{ave}*m. And for Response, from the fig 2, it is 32$[bit]$+32$[bit]$ (length of 2 IP address). This shceme inherits all metrits from one-server query scheme, however, compared with the cons(communication cost) of one-server scheme, there is some diffrence.

When comparing the communication cost of two

schemes, in the query part, the cost rises from H_{ave}*(m+1)–(1-server) to H_{ave}*(m+1)+ H_{ave}*m–(2-servers); However, in the response part, the cost declines from (m + 1)*32[bit]–(1-server) to 32[bit] +32[bit]–(2-servers).

However, when concerning about privacy-preserving, we can get a more optimal value with the Two-Server scheme. The client sends hostname ranges to two separate servers, any single server can not make sure whether the target lies in the range that it has received. It is more secure than One-Server based range query under the assumption that servers can not communicate with each other.

On another side, we have to discuss a weakness of the 2-server PIR scheme with a special condition. If we suppose one server conspires with another one, the server can obtain a hostname which the clients wants to know. Also if the attacker can monitor all of the queries from a client, the attacker can know the hostname.

6 Concluding Remarks

Getting some inspiration from the Two-Server PIR query model, we proposed a new DNS query framework based on a Two-Server PIR scheme, and we proved that we could get a more optimal privacy-preserving value by the new model. We also gave a careful comparation with One-Server PIR based DNS query model and gave some pros and cons of two schemes.

As with many simple solutions, the challenge lies in the details(our new shceme only fit the 32[bit] length IP address). We must concede that there is still a long way from applying the scheme since protocols we are using must be modified respectively to support new proposals. Many wonderful security and privacy enhancing techniques have been proposed and launched by the research community only to quietly fade into obscurity due to usability issues. As mentioned earlier in the paper, DNS query is unfortunately ignored by the majority of Internet users. For this reason, finding simple and unobtrusive ways of making average users aware of both the need for effective DNS query and the need to protect their privacy is a major challenge. We hope our work could be an initial step in this line of research and will attract more attention from the whole society.

参考文献

[1] Privacy International. *Overview of privacy, 2004*, Available at http://privacyinternational.org/privhroverview2004.

[2] G. Miklau and D. Suciu. *A formal analysis of information disclosure in data exchange.* In Proceedings of Computer Security Symposium 2006, In SIGMOD, 2004.

[3] C. Cachin, S. Micali, and M. Stadler. *Computationally private information retrieval with polylog communication.* In Proceedings of Eurocrypt ' 99, LNCS. IACR, Springer Verlag, 1999.

[4] Federal Trade Commission. *Protecting Consumers from Spam, Spyware, and Fraud.* A Legislative Recommendation to Congress, 2005.

[5] tcpdump. *tcpdump Online.* http://www.tcpdump.org/.

[6] wireshark. http://www.wireshark.org/.

[7] NIC-SE *DNSSEC - What is it and what does it do?.* http://www.dnssec.org/, April, 2006.

[8] B. Chor, O. Goldreich, E. Kushilevitz and M. Sudan. *Private Information Retrieval*, Proc. 36th IEEE Symposium on Foundations of Computer Science(FOCS), 1995.

[9] D. A. Cooper and K. P. Birman. *Preserving privacy in a network of mobile computers.* Proc. IEEE Symposium on Security and Privacy, pp. 26-38, 1995.

[10] Fangming Zhao, Hori Yoshiaki Hori and Kouichi Sakurai. *DNS Query with Privacy-Preserving.* In Proceedings of Computer Security Symposium 2006, pages 19-24, October, 2006.