Evaluation of Privacy for DNS Private Exchange

Gaurav Korde
gkorde@buffalo.edu

Pranav Jain
pranavja@buffalo.edu

Jim Smith
howardsm@buffalo.edu

Nikhil Selveraj
nikhilse@buffalo.edu

ABSTRACT

DNS security monitors process a lot of information from users, some of which can be considered extremely sensitive for an individual. The misuse of such information by an adversary may be harmful for more than just the communicating parties from where the information is leaked. Although various schemes have been proposed recently to counter the issues in privacy in DNS exchange, there is absence of a comprehensive work that evaluates these schemes from the privacy perspective. And choosing the most appropriate scheme for a given DNS setup can be hard with all these new schemes. Our work aims to facilitate this process for an individual who is looking for a comparison between these schemes. It also throws light on the potential privacy issues associated with these schemes. We evaluate the schemes based on the vulnerability of these schemes in an experimental environment with a well defined adversary having a set of eavesdropping powers on the network.

1. INTRODUCTION

The DNS (Domain Name System) protocol has long been used to serve its primary functionality of converting a given domain name or URL into its corresponding IP (Internet Protocol) address and vice-versa. However, these days, the DNS protocol is susceptible to a large number of attacks mainly due to weaknesses in its underlying IP protocol. Improper authentication, integrity, timing attacks etc. are some of the weaknesses that DNS is vulnerable to. Another example is DNS security monitors which eavesdrop on the information being communicated between two parties. This information could be confidential, and if misused, could prove to be harmful for both the communicating parties. There are no fully established mechanisms to counter these attacks and a lot of new schemes are being proposed in this regard. Our work analyzes some of these proposed schemes from the privacy perspective by evaluating them based on a set of privacy parameters in the presence of an adversary in the form of a pervasive monitor. We intend this document to be helpful for people who are looking for a comparison between different DNS schemes from privacy perspective and hope that it would help them in choosing the most appropriate scheme based on their requirements.

In our work, we have further progressed upon the draft [5] and have evaluated the private DNS schemes including Private DNS, Confidential DNS, DNS over TLS, IPSEC, Mixed networks, QName Minimization and dummy traffic. Out of these, Mixed networks and IPSEC are fairly old protocols, but their effectiveness with respect to DNS privacy are still unknown. Besides, IPSEC packets have the problem of exceeding the Maximum Transmission Unit when deployed and cause different routers to simply refuse transmission of oversized packets or perform incorrect Network Address Translation [6]. So even if it might be a great scheme from a privacy perspective, its practical deployment on a large network with a mix of standard and non-standard network devices seems infeasible. The other schemes are fairly new and are yet to be deployed on a large scale and hence there is lack of evidence of their effectiveness in the context of DNS privacy when widely deployed.

Our work evaluates the different DNS privacy schemes based on a probabilistic approach. We have defined different experiments that aim to emulate a real life network with different types of adversaries (defined in section 4.2). We then calculate the different probabilities of the system being compromised by the adversary based on various privacy parameters (defined in 3.2) in these experimental environments. The comparison of the various schemes based on these probabilities gives an insight into how private these schemes are. Different DNS setups would want to choose schemes based on different privacy parameters. The comparison chart (section 5.3) obtained by our work will help the reader choose the scheme that aims to maximize the privacy for a particular parameter based on their requirements for their setup. We also discuss the feasibility of deploying the various schemes. Although this does not come under privacy considerations, this discussion is necessary, since a scheme that makes high privacy promises, but isn't feasible to deploy widely is not so practically useful.

Our report is structured as follows. In the related work section (section 2), we discuss the various privacy promising schemes in short to give the reader a general idea on what each scheme does. We then go on to define the terms and definitions in the preliminaries (section 3) we use through-

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

out the report. It includes the general terms we use (section 3.1) and also the notions of privacy we use in our evaluation. Section 4 contains our work including a summary of each scheme and pseudocodes (section 4.1) wherever possible and the adversary definitions (section 4.2) we use to formally define the adversary in our privacy experiments. In section 5 we actually evaluate the different schemes based on the evaluation experiments (section 5.1) we define at the beginning of the section and our general comparison chart (section 5.2) for the schemes and a comparison chart based on privacy parameters (section 5.3). Section 6 provides our conclusion based on our evaluation results from section 5 and the ease and cost of implementing the schemes. Sections 7 and 8 contain individual contributions and bibliography respectively.

2. RELATED WORK

This report focuses on evaluating the following seven important schemes that can be used to provide privacy in DNS private exchange - Dummy traffic, Mixed networks, DNS over TLS, IPSEC, QName Minimization, Private DNS and Confidential DNS. Most of these schemes are evaluated for various privacy parameters in the IETF draft [5] where they analyze the various methods for DNS private exchange and measure their performance and effectiveness against pervasive monitoring. They use various risk models to evaluate different notions of privacy. The notions of privacy and other definitions as used in the security community are redefined from the privacy perspective in the draft. They claim that a binary notion of privacy is not feasible since it is not continuously obtainable and hence continuously quantify them based on negligible probabilities. These are then used to determine the probabilities for the various privacy parameters used in their discussion.

The discussion in the draft is limited to the link between a stub and recursive and that between recursive resolver and an authoritative name server. They then define the risk models which are basically different types of adversaries in a particular DNS setup. The two risk models they use are passive pervasive monitor and active monitor. Then to evaluate, they use risk templates in a pseudo code style and output the probabilities based on various privacy parameters. These evaluations consider the basic components of each scheme, analyze them, and then output the different probabilities for each privacy parameter as the evaluation metric. A few interesting findings (essentially probabilities) emerge from these evaluations that help them draw conclusions in their evaluation.

3. PRELIMINARIES

In order to evaluate the privacy of the different DNS schemes, we define below the various risk models, formal definitions for different types of adversaries, and security and privacy terms used throughout this work.

3.1 General terms:

DNS - The Domain Name System is a system that converts between a domain name or URL and its IP address.

Client/Stub Resolver - This is an entity requesting a DNS resolution.

Proxy/Forwarder - This is an entity that sits in between the stub and recursive resolver or between the recursive resolver and authoritative server that implements caching of DNS resolutions for quicker responses to the client.

Recursive Resolver - This is an entity that implements the recursive function for the client to get a DNS resolution.

Authoritative Server - This is an entity that holds the DNS resolution that the recursive resolver queries.

TLS - The transport layer security is a protocol that ensures the privacy of traffic over the internet.

Risk Model/Adversary - A risk model or an adversary is a set of capabilities that the adversary has in order to breach the privacy of an individual.

Privacy Mechanism - These are mechanisms that enhance the privacy for communication over networks.

TOR - The Onion Router is a software that routes internet traffic through a network of other Tor users creating anonymous communication.

Eavesdropper - This is an adversary that can listen to the user's communications without the knowledge of the user.

Observer - This is an authorized entity that can observe the user's communications.

Individual - This is a user on a network.

Enabler - This is an entity that helps the communication on a network without being directly part of that communication.

Initiator - This is an entity that starts the communication with an individual.

3.2 Privacy related definitions:

We define the privacy parameters we will use to evaluate the schemes as follows:
1. Unlinkability: Any two given entities A and B are said to be unlinkable, if one of the entities cannot be linked to the other given the other entity and vice versa. In other words, given A, it cannot be linked to B and vice versa. Thus, the probability of finding B given A, or of finding A given B is zero if A and B are unlinkable.

In the context of our discussion, any two entities are said to be unlinkable if the chance of an adversary identifying them as associated to each other (linkable to each other) is negligible. On the other hand, they are said to be linkable if this probability is close to 1.

2. Undetectability: In the context of our work, it is defined as the property of an entity being undetectable by an adversary. It thus makes detectability, which is the probability of the adversary being able to determine the existence of that entity, negligible.

In other words, undetectability property means that the probability of the adversary being able to determine the existence of an entity is negligible. This makes the undetectability property of that entity close to 1.

3. Unobservability: It is the property that the entity under discussion is also anonymous - meaning that an adversary cannot detect or observe the entity in the network. The probability of the adversary being able to detect or identify a particular entity on the network is thus negligible for an unobservable entity.

4. Identifiability: This is the degree measure of the ability of an adversary to identify an entity or entities on a network which it is eavesdropping on.

4. OUR WORK

4.1 Summary of Different Schemes:

Before we evaluate the schemes based on various privacy parameters, with a view to give the readers an insight on what each scheme does, we have summarized them below. We have also given a short pseudocode to give readers to skim through faster through the section.

The draft [4], with a view to compliment [8], introduces a new scheme based on DNSE-JX to secure DNS connection between a client - resolver or resolver - authoritative server and wrapping it in a new framing protocol. Its motive was to address the issues in DNS private exchange under the constraints of legacy deployment limitations, integrity attacks and limited message size, and also providing the CIA requirements of computer security. The draft [2] describes an algorithm called as Confidential DNS protocol which encrypts data for DNS query/response scenario. The motive of this protocol is to make monitoring DNS queries too expensive. However, the protocol does not offer any authentication. The draft [1] introduces a new protocol called DNS-over-TLS which provides privacy from eavesdropping for two communicating parties over a DNS channel. The protocol makes use of persistent TCP connections along with the TLS protocol in order to achieve this privacy.

Mix networks [9] are networks where a mix of input messages result into an output in such a way that it is infeasible to link an output to a corresponding input or vice versa. To do this, the mixer changes the appearance and the flow of messages by encrypting or padding them and reordering and delaying them respectively. Dummy traffic [10] is the introduction of fake messages in a mix network in order to make it difficult for an attacker to deploy active and passive attacks on the network. QNAME minimization [7] is the technique where the DNS resolver obscures the full original QNAME when sending it on the upstream server. In the original draft[11], they claim it to serve as one of the tools amongst many to solve the DNS privacy problem. They say that the less data you send out, the fewer privacy problems you have.

A. Private-DNS:

In [4], with a view to compliment DNSSEC, a new scheme was introduced based on DNSE-JX to secure DNS connection between a client - resolver or resolver - authoritative server and wrapping in a new framing protocol. Its motive was to address the issues in DNS private exchange under the constraints of legacy deployment limitations, integrity attacks and limited message size, and also providing the CIA requirements of computer security.

The draft describes the architecture of the scheme in terms of the service connection being provided by [4, some citation, find out from the previous report] and describing the new DNS message encapsulation method that supports encryption and authentication. The service connection mechanism is responsible for establishing a connection context between a client and a service comprising of – a security context between the client and the connection service, and one or more query host connection contexts, each consisting of network connection description (IP address, Port, Protocol, transport) and Security Context (opaque identifier, key, algorithm choice) between the client and the query. It contains entities like service identifier, protocol name, presentation, and transport in case UDP is used; and protocol, presentation and transport in case HTTP is used.

The encapsulation part supports encryption, authentication, multiple DNS queries and responses per Private-DNS query, and multiple DNS packet responses. The draft also describes the various components of the encapsulation part in detail - the request, the response and the payload. The design choice to be made is between a tagged data format (provides greater flexibility) like JSON and a position based format (provides more efficiency) like the one used in TLS. Presently, TLS position based approach is preferred since it's compatible with the traditional approaches in DNS.

Simple Algorithm for Payload Encapsulation:

1. Open Socket connection between client and server.
2. Client will write data to server with packet type REQUEST for type of encryption.
3. Server will read the REQUEST file and encrypt payload with requested type.
4. Close connection.

B. Confidential DNS:

The draft [2] describes an algorithm described as Confidential DNS protocol which encrypts data for DNS query/response scenario. The motive of this protocol is to make monitoring DNS queries too expensive. The protocol does not offer any authentication.

It uses a cacheable ENCRYPT Resource Record (ENCRYPT RR) and optionally a cacheable shared secret. The server has the ability to advertise which crypto suites and key lengths may be used in the ENCRYPT RR and the client can choose a crypto suite from this list and include it in the subsequent queries.

Since this is an opportunistic encryption, the key is refetched if an exchange fails. The key from the client can be cached by the client, using the TTL specified in the ENCRYPT RR. The IP address of the server distinguishes keys in the cache. The server may also cache shared secrets and keys from clients.

The encryption happens in the following manner. If a client wants to fetch the keys for the server from the server, it performs a query with query type ENCRYPT and query

name "." which is a root label. If a client wants to perform an encrypted query, it sends an unencrypted outer packet, with query type ENCRYPT and query name "."(root label). In encrypted queries, the ENCRYPT RR has a type set of RRS which has a size of 2 bits and encrypts the data with a selected algorithm and key ID. If the client wants to use a symmetric key, it will use the ENCRYPT of type SYM which has a size of 3 bits and contains encrypted data in the form of rdata of an ENCRYPT of type KEY and has the symmetric key.

Opportunistic encryption is good against passive adversaries. However, it does come up short when faced with an active adversary. This technique does not protect against timing, traffic analysis (what IP address is contacted), and the packet size, RR count, header flags and header RCODE can be observed.

Simple Algorithm for Confidential DNS:

1. Open Server Client connection.
2. Client sends message to server with ENCRYPT request.
3. If the transmission fails, client will send message again.
4. Server reads the ENCRYPT type and sends corresponding key.
5. Close connection.

C. Specification for DNS over TLS:

The paper[1] basically addresses the problem of DNS privacy which is of huge importance since a lot of the queries today are sent unencrypted over network channels. This makes them vulnerable to eavesdropping which in turn reduces privacy. The DNS-over-TLS protocol provides this privacy to both parties.

This protocol relies on establishing a TCP connection between the two hosts by the Handshake protocol, followed by TLS negotiation and establishment after which the connection is said to be protected from eavesdropping. The following diagram describes the establishment of a TCP connection followed by the TLS session establishment using the handshake protocol.

usage policies are described; firstly, opportunistic privacy profile where privacy is not guaranteed and the channel is susceptible to on-path attacks and secondly, out-of-band key-pinned privacy profile where the client establishes a connection only with a server which it can validate using the SPKI Fingerprints.

The paper then talks about the performance considerations of this scheme with regards to latency, number of connections, state and processing. Since this scheme makes use of TCP connections, it requires an initial handshake and this will increase the latency. Since the scheme uses persistent TCP, this requires maintaining the state on the server. The use of TLS encryption algorithms accounts for the increased processing overhead. Finally, the number of TCP connections should be reduced.

The DNS-over-TLS scheme, although it provides privacy, it is still prone to some attacks. For instance, it is susceptible to man-in-the-middle attacks. It is also largely prone to traffic analysis and side channel leaks. For example, an adversary like a pervasive monitor could glean information by observing message timings and sizes many times. Using this information, it can learn a lot about the messages being sent. This shows that although the scheme uses encrypted data, it is still vulnerable to such attacks. To sum up, the paper clearly describes a new scheme called DNS-over-TLS which provides privacy of data to users by ensuring that only encrypted data is sent over the channel, however it fails to address the issues mentioned above like man-in-the-middle attacks and timing attacks which occur in the presence of even a pervasive monitor.

Simple algorithm for Persistent TCP connection:

1. Open Server socket connection by client sending SYN message.
2. If Server accepts message, it will send ACK message.
3. If delivery from client fails, send again for n number of attempts
4. If client receives ACK message, send ACK again and establish persistent connection.

D. Mixed Networks:

Mixed networks[10] output the input messages in such a way that it is infeasible for an adversary to link them together. It changes the flow of messages by reordering and randomly delaying messages, and the appearance by encrypting and padding them. To randomize, it uses different cryptographic primitives like padding and encryption. It also uses different integrity mechanisms to ensure message integrity. The modification in flow of messages is done to make it difficult for an attacker to link a message to a user - unlinkability property. To increase the anonymity of the users on the network, these networks use mixes in the mix network. This ensures that even if certain mixes are corrupted or manipulated by an attacker, it does not break the anonymity of the users. It also improves the reliability of the system since failure of a mix does not lead to DoS. There are various types of options that have been explored in order to achieve this mix. But a discussion on these in detail is out of scope of our work. Please refer [10] for a detailed analysis of the techniques.

Simple algorithm for Mixing packets:

1. Write class Mixer for client packets.
2. Introduce reordering and random delaying.
3. Define padding and encryption for every packet.
4. Open Server Client connection and transfer packets.

E. Dummy Traffic:

In simple terms, dummy traffic[10] is redundant traffic introduced by the sender in the form of additional packets to disguise the actual data packet from the adversary. This can be achieved by the sender - who generates k dummy messages to disguise the original message - or by the mixes[11] - which provide a better disguise for the original message by introducing a fake mix destination for the dummy messages, thereby providing anonymity and unobservability. Even though due to the enormous amount of resources it requires this system is not practical yet, we discuss it from the privacy perspective as it could be feasible to deploy in the near future. Again, the different techniques of this method are discussed in detail in [10] and are out of scope of our work.

Simple algorithm for Dummy Traffic:

1. Open Server Client connection.
2. Create class Dummy to parse packets for dummy packets generation.
3. Generate k dummy messages per packet for every transaction.

4. Transmit data and close connection.

F. Qname Minimization:
QName Minimization[7] is a technique that aims to minimize the amount of information being sent from the resolver to the authoritative name server. In general, when a user wants to know the IP address of a particular domain name, he will send the DNS query to many name servers, starting with the root server. If the root server is able to respond to the query i.e. if the root server is authoritative for that domain name, it will respond immediately. Otherwise it will have to refer the user to another name server that can answer the query directly or refer the user to another name server. For example, if the domain name is www.abc.com, the root server will refer the user to a name server for .com which in turn refers the user to another name server for abc.com and so on.

However, this does not provide privacy since the full domain name is being exposed each time a query is made. So to provide more privacy, the QName minimization technique is used. For example, the root server does not need to know the full domain name www.abc.com to know that it has to route the request to another name server. It only needs the last label of the domain name which is .com. It does not need to know the labels before .com to make this decision. This is called zone descendance in the DNS domain name hierarchy because each time a referral is made to the authoritative name server for a particular zone, we descend a zone in the domain name hierarchy followed by prepending of labels from the original domain name.

For example, we start with .com, followed by abc.com, followed by www.abc.com. This ensures that only a minimal portion of the domain name is exposed at each referral to the name server and this accounts for the increased privacy from the normal DNS which exposes the full domain name.
Simple algorithm for implementing Qname Minimization:
1. Create class Truncate to parse domain names.
2A. Store each domain hierarchy in a temp variable
2B. Transmit variable to DNS servers.
3. If reply is obtained, append higher hierarchy to temp variable and repeat step 2B.
4. Obtain IP address of the entire domain name after repeating steps 2 and 3 for all hierarchies.

G. IPSEC:
IPSEC[12] is a protocol suite for secure Internet Protocol communications. It defines authentication and encryption for IP packets on a secure communication channel. It lies in the Internet Layer of the Internet Protocol Suite. It protects all other security protocols from the above layers, and can be seen as the first line of defense for the protection of an network packet.

IPSEC offers two modes of operation. The first is called the Transport mode and in this mode, only the payload is encrypted and authentication is optional. The IP header is not encrypted, which means the address of the destination is visible to a monitor. The second mode of operation is called Tunnel mode and in this mode, the entire packet is encrypted. The whole packet is then encapsulated in a new packet with a new IP header. This is often used to create virtual private networks between two hosts or between hosts and routers.

IPSEC however does have an increased overhead due to the encryption algorithm. The tunnel mode is quite costly to implement. Time complexity does raise an issue with regards to implementation.
Simple algorithm to encapsulate packet:
1. Select mode of operation: Tunnel or Transport
2A. For Transport mode:
2B. Encrypt payload and append IP header to the front and transmit.
3A. For Tunnel mode:
3B. Obtain entire packet from upper layers and encrypt the entire packet.
3C. Append new IP header to the entire encrypted packet.
4. Transmit packet obtained from 2 or 3.

4.2 Adversary:

The types of adversaries we are using are defined below. Depending on the powers defined, the adversary is an actor with the power of eavesdropping on the traffic between two entities with or without modifying or inserting traffic with the goal of breaching the privacy of individuals. Specifically, the adversary eavesdrops on the DNS channels Stub Resolver-Recursive Resolver, Stub Resolver-Proxy, Proxy-Recursive Resolver and the adversary has won when he is able to determine partially or fully the identity of the individual/network entities and the contents of that traffic. For the schemes to be private, the adversary should also not be able to determine the change of communicating users or identify a packet for a specific individual when the DNS packets are exchanged. They should just be random bits for the adversary (indistinguishable, undetectable and unlinkable for different users). We define adversaries and the evaluation experiments inspired by the way they are defined formally in the book [14]. The types of adversaries based on the different attack types are defined below:

Based on the set of capabilities of the attacker and the amount of information the attacker has with the goal of breaching the privacy of an individual with respect to a particular privacy metric, we denote the adversary as follows:

$$Adv^{Type,Globalflag}_{Compromisedlinks}$$

Where,
Type indicates the type of adversary as defined in the following paragraphs.
Global flag is an indicator of whether the adversary is local or global - that is whether its powers are limited to a part of the network or the entire network.
Compromised links indicates the links that are compromised as a result of the adversary eavesdropping/listening over the different channels/network entities.

Adversary 1 (Pervasive monitor):

Relies on the monitoring capabilities of an adversary capable of eavesdropping on traffic between any two end points (shown in the diagram), to breach the privacy of individuals from the DNS traffic. This attacker has the abilities to

eavesdrop pervasively on many links at once. These attackers are passive – they don't modify or insert traffic. They are subdivided into:

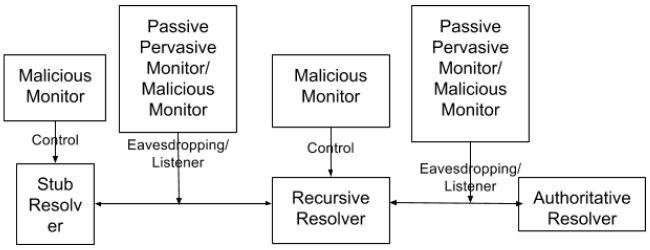
P1: $Adv^{P1}_{S-P,P-R,S-R}$ - One who can monitor links carrying individual's traffic through access to traffic along the links shown but does not target any specific individual. It tries to learn maximum possible data about all the individuals using the links and the system location to launch attacks.

P2: $Adv^{P2}_{S-P,P-R,S-R}$ - Similar to P1 except that it targets links to target specific individuals by using techniques like putting intermediaries between links to obtain traffic on them.

Adversary 2 (Malicious monitor):

M: $Adv^{M}_{S-P,P-R,S-R}$ - It is an attacker with all the capabilities of a pervasive monitor along with the additional capability of being able to control one or more infrastructure elements, which give it the power to modify traffic in both directions on that link. It could have any of the network entities given in the diagram under its control.

Diagram:



5. EVALUATION AND DISCUSSION

We now evaluate the schemes we summarized in section 4.1. But before we do that, we define an evaluation experiment as follows:

1. Let $Priv = Parameter^{Adversary,Attack\ type}_{Protocol\ details}$ be the experiment where, with the set of powers defined as per the adversary type, the adversary $Adv^{Type,Global}_{Compromised\ links}$ is able to obtain/infer information about the user(s) on the network. Parameter indicates the privacy parameter we are evaluating - namely, (un)detectability or (un)linkability.

2. Let $Pr[Priv = Parameter^{Adversary,Attack\ type}_{Protocol\ details}]$ be the probability of the attacker winning/losing.

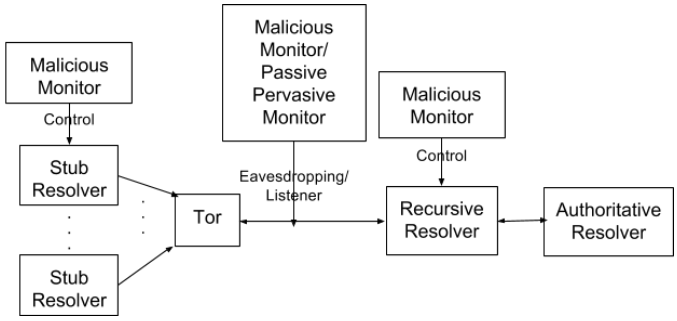
$Pr[Priv = Parameter^{Adversary,Attack\ type}_{Protocol\ details} = 1]$ indicates that the adversary has won for the given privacy parameter and

$Pr[Priv = Parameter^{Adversary,Attack\ type}_{Protocol\ details} = 0]$ indicates that the adversary has lost for that privacy parameter.

5.1 Evaluation:

A. Mix Networks:
These use systems such as Tor to route DNS queries to the DNS server. This makes it difficult for a monitor to determine which individual or subject sent which DNS query. This is because Tor is such a huge network, even if it's not represented as one, that the adversary has to somehow link entering traffic into Tor with exiting traffic from Tor to a single stub resolver.

Diagram:



The evaluation experiment is given as:
$Priv = Parameter^{Adv^{P1}_{S-R},Eav,Global}_{Mix,u,uniform}$

Here, the P1 type of attacker eavesdrops on the S-R link. While the queries here are detectable, they cannot be linked to the same individual - unlinkability. u indicates the number of users in the network. Assuming that the traffic from each of these users is uniformly random, the probability that a given query comes from a given user is $1/u1/u$. This goes on decreasing as the number of queries goes on increasing, ie, the probability that any two given queries are from a single user is $1/u * 1/u = 1/u21/u * 1/u = 1/u^2$. Thus the unlinkability for this becomes $1-1/u21 - 1/u^2$, giving a fair amount of unlinkability. This remains the same irrespective of the adversary type. For a large number of users, the unlinkability probability approaches 1.

In other words, the linkability probability approaches negl *negl*with square inverse proportionality. In terms of our experiment, this is given as:

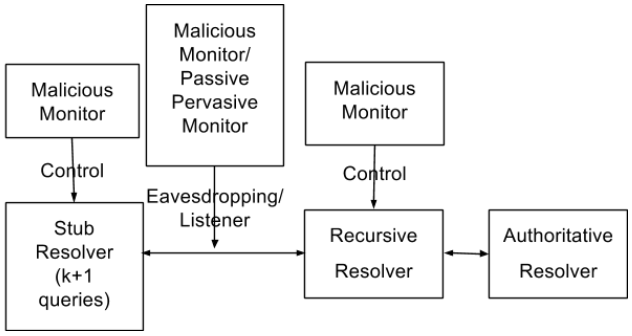
$$Pr[Priv = Linkability^{Adv^{P1}_{S-R},Eav,Global}_{Mix,u,uniform} = 1]$$

$$=1\text{-}Pr[Priv=Linkability^{Adv^{P1}_{S-R},Eav,Global}_{Mix,u,uniform} = 0]$$

$$=negl$$

B. Dummy Traffic:
This is where the initiator generates k dummy queries along with the actual query. This makes it difficult for the monitor to determine which query is actually useful.

Diagram:



The evaluation is given as:

$$Priv^{Adv^{P1}_{S-R},Eav,Global}_{Dummy\ traffic,k,uniform}$$

Here, k indicates the number of queries, that are used to disguise the original query, selected uniformly at random from a pool of queries, for every original query. The attacker here is Passive Pervasive Monitor (P1) and the link it can eavesdrop upon is S-R.
It provides undetectability with a probability of $1-(1/(queries+1))$. It goes on increasing with the increase in the number of traffic, and approaches 1 for a large number of traffic.

In other words, the detectability probability approaches *negl* with inverse proportionality. In terms of our experiment, this is given as:

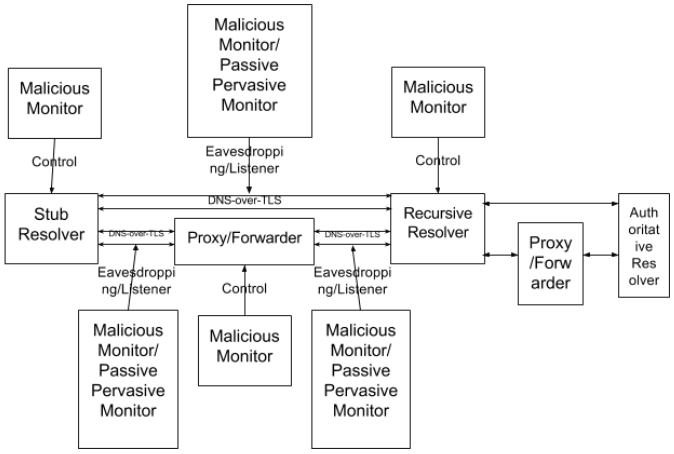
$$Pr[Priv = Detectability^{Adv^{P1}_{S-R},Eav,Global}_{Dummy\ traffic,k,uniform} = 1]$$

$$=1\text{-}Pr[Priv=Detectability^{Adv^{P1}_{S-R},Eav,Global}_{Dummy\ traffic,k,uniform} = 0]$$
$$=negl$$

This probability decreases when P1 is replaced by M, since it can inject responses to arbitrary requests and then track them.

C. Encrypted Channel Mechanisms-DNS over TLS:
These are where an initiator has an encrypted channel using DNS-over-TLS with an enabler to send queries over that cannot be monitored.

Diagram:



The evaluation is given as:

$$Priv^{Adv^{P2}_{S-R},Eav/Lis,Global}_{DNS-over-TLS,SHA256,ECDSA,port\ 53,uniform}$$

Here, the attacker type is Directed Monitor attack model (P2) and thus it can apply extra resources to the encrypted channel. Irrespective of using upgrade-based TLS on port 53 or port-based TLS on a dedicated port, since both indicate the use of DNS, it's not undetectable. The source address of the user is exposed in all the cases.

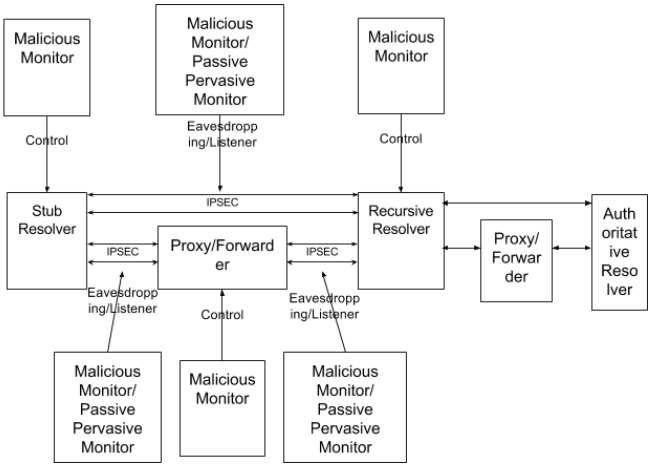
Thus, the detectability probability in terms of our experiment is given as:

$$Pr[Priv = Detectability^{Adv^{P2}_{S-R},Eav/Lis,Global}_{DNS-over-TLS,SHA256,ECDSA,port\ 53,uniform} = 1]$$
$$= 1$$

A possible method to increase undetectability is IPSEC since it hides the fact that all the traffic from S-R was DNS.

D. Encrypted Channel Mechanisms-IPSEC:
These are where an initiator has an encrypted channel using IPSEC with an enabler to send queries over that cannot be monitored.

Diagram:



The evaluation is given as:

$$Priv_{IPSEC,uniform}^{Adv_{S-R}^{P2},Eav/Lis,Global}$$

Here, the attacker type is Directed Monitor attack model (M) and thus it can apply extra resources to the encrypted channel. The type of attacker is irrelevant since IPSEC hides the fact that the traffic is DNS traffic, providing undetectability and unlinkability.

Thus, the detectability probability in terms of our experiment is given as:

$$Pr[Priv = Detectability_{IPSEC,uniform}^{Adv_{S-R}^{P2},Eav/Lis,Global} = 1] = 0$$

and, the linkability probability in terms of our experiment is given as:

$$Pr[Priv = Linkability_{IPSEC,uniform}^{Adv_{S-R}^{P2},Eav/Lis,Global} = 1] = 0$$

Therefore, IPSEC provides the best privacy guarantees compared to the other mechanisms. The only problem is that IPSEC is extremely difficult to implement and is not practical in an actual environment.

E. QName Minimization:

This technique uses a resolver that has the query name minimizing capability. The full domain name is not provided in each query to a name server so that this provides more privacy.
The evaluation is given as:

$$Priv_{QNameMin}^{Adv_{R-A}^{M},Attacker,Global}$$

Here the attacker type is an active monitor. So it can target specific links to target specific individuals. It can also introduce intermediaries into these links to obtain traffic.

The goodness of the privacy is measured based on the linkability capability of the monitor i.e. its ability to link the labels of two minimized queries to each other and thereby to the original query. So the probability of privacy here would depend on the number of labels in the queries, the number of queries and the number of users using a particular resolver. The privacy measure here is analytical since it depends on all these factors. Also, this technique does not provide privacy for an S-R link but only for an R-A link.

F. Private DNS:

In this technique, the entities taken into consideration are S, R, A and the authenticating function for the user of S before the encrypted channel to R or A is formed. In case of S-R link, the identifiability of S as an individual is the same as if it is identified from the source address of S. On the other hand, in case of S-A link, the source address is provided to A, making unlinkability property as 0.

5.2 General Comparison chart for schemes:

SCHEME	Privacy and Security	Cost and Complexity	Infrastructure	Feasibility and Difficulty
DNS-over-TLS	TLS provides privacy by encrypting DNS queries. Insecure against man in the middle and timing attacks.	TCP connections lead to increased latency and processing. Increased time complexity because of initial TCP handshake and TLS encryption algorithms.	Uses TLS protocol which runs on TCP. Uses persistent TCP connections to reduce latency.	Feasible to implement although it is prone to some attacks. Requires maintaining persistent TCP connections and state on the server.
Private DNS	Uses DNS query encapsulation to provide privacy. Provides authentication, encryption and security against integrity attacks.	Requires usage of a message encapsulation method which leads to increased cost.	Has 2 parts: 1. service connection which establishes connection between client and server and 2. query encapsulation which provides encryption and authentication.	It is feasible. Can be implemented.
Mix Networks	Use Tor to route DNS queries. This makes it difficult for the monitor to link a particular query with its sender. Provide unlinkability.	More cost because of increased number of nodes. Increased space complexity due to more number of nodes.	Uses Tor network along with the existing nodes.	Not quite feasible. Requires maintaining large number of nodes.
Dummy Traffic	Uses k dummy queries which disguise the actual query. Provides undetectability.	Requires introducing additional queries. No change in complexity.	Uses the initiator to generate k dummy queries.	It is feasible. Not difficult to implement.
IPSEC	Uses an encrypted channel for sending queries which provides privacy and security.	Increased processing overhead due to the encryption algorithm. Time complexity increases.	Uses an initiator with an enabler to send queries over the encrypted channel.	Not very feasible. Difficult to implement.
Confidential DNS	Uses opportunistic encryption where the key to be used for encryption is fetched in each exchange	Increased time complexity because of initial communication between client and server which involves agreeing upon the	Uses Encrypt Resource Record and a cacheable shared secret. The cryptographic suites and the key lengths to be	Feasible to implement.

5.3 Comparison chart based on privacy mechanisms:

SCHEME	PRIVACY PROBABILITY BASED ON PARAMETERS		
	Unlinkability	**Undetectability**	**Unobservability**
Mix Networks	~1 for P1 or M	Uncertain	Uncertain
Dummy Traffic	~1 if mix network is used	~1 for P1, decreases for M	~1
DNS-over-TLS	1	0	0
IPSEC	1	1	1
QName Minimization	1	0	0
Private DNS	0 (in case of S-A link)	Uncertain	Uncertain

6. CONCLUDING REMARKS

The purpose of this report is to help a naive user understand the basics of DNS Private Exchange and the privacy issue that it addresses. A brief description of the protocols currently being evaluated has been written in this report with the efforts to keep it as simple as possible for the reader to understand. Similarly, the privacy notions used for evaluation and the preliminaries have been defined in a simple yet concise manner for the average user.

Though the report is by no means an exhaustive journal on the work done so far on the topic of Privacy in DNS Private Exchange, it hopefully helps in laying a basic foundation for

the same.

Amongst the protocols described in this report, IPSEC shows the most promising statistics for deployment on a large scale provided it can be developed for a time and cost optimal solution. Currently, as shown in the report above, both these factors coupled with the lack of cutting edge infrastructure are the major hurdles in the way of IPSEC implementation.

As per our observations, Dummy traffic also seems to be a promising scheme for ensuring privacy in DNS exchange. But due to lack of implementation and evidence supporting its success, we are skeptical as to how successful it can be.

In conclusion, DNS over TLS seems to be the most viable option for now.

7. BIBLIOGRAPHY

[1] Zhu, L., Hu, Z., Heidemann, J., Wessels, D., Mankin, A., and P. Hoffman, "TLS for DNS: Initiation and Performance Considerations", draft-hzhwm-dprive-start-tls-for-dns-01.txt.

[2] Wijngaards, W. and G. Wiley, "Confidential DNS", draft-wijngaards-dnsop-confidentialdns-03.

[3] Osterweil, E., Wiley, G., Mitchell, D., and A. Newton, "Opportunistic Encryption with DANE Semantics and IPsec: IPSECA".

[4] Hallam-Baker, P., "Private-DNS", draft-hallambaker-wsconnect-08.

[5] A. Mohaisen, A. Mankin, "Evaluation of Privacy for DNS Private Exchange", draft-am-dprive-eval-00.

[6] http://www.uninet.edu/6fevu/text/IPSEC-NAT.SGML.html

[7] https://tools.ietf.org/html/draft-ietf-dnsop-qname-minimisation-09

[8] https://www.icann.org/resources/pages/dnssec-qaa-2014-01-29-en

[9] https://www.ee.washington.edu/research/nsl/papers/proceedings-06.pdf

[10] http://freehaven.net/anonbib/cache/taxonomy-dummy.pdf

[11] Dai, Wei (1996). Pipenet 1.1. Usenet post

[12] https://en.wikipedia.org/wiki/IPsec

[13] Connection-Oriented DNS to Improve Privacy and Security

[14] Introduction to Modern Cryptography, Jonathan Katz and Yehuda Lindell