NSA's MORECOWBELL: Knell for DNS

Christian Grothoff, Inria
Matthias Wachs, TU Munich
Monika Ermert, Heise Verlag
Jacob Appelbaum, Tor Project

1 Introduction

On the net, close to everything starts with a request to the Domain Name System (DNS), a core Internet protocol to allow users to access Internet services by names, such as `www.example.com`, instead of using numeric IP addresses, like 2001:DB8:4145::4242. Developed in the "Internet good old times" the contemporary DNS is like a large network activity chart for the visually impaired. Consequently, it now attracts not only all sorts of commercially-motivated surveillance, but, as new documents of the NSA spy program MORECOWBELL confirm, also the National Security Agency. Given the design weaknesses of DNS, this begs the question if DNS be secured and saved, or if it has to be replaced — at least for some use cases.

In the last two years, there has been a flurry of activity to address security and privacy in DNS at the Internet Engineering Task Force (IETF), the body that documents the DNS standards. The Internet Architecture Board, peer body of the IETF, just called on the engineers to use encryption everywhere, possibly including DNS. [4]

A recent draft [6] by the IETF on DNS privacy starts by acknowledging that the DNS

> "... is one of the most important infrastructure components of the Internet and one of the most often ignored or misunderstood. Almost every activity on the Internet starts with a DNS query (and often several). Its use has many privacy implications ..."

Despite seemingly quick consensus on this assessment, the IETF is not expecting that existing industry solutions will change the situation anytime soon:

> "It seems today that the possibility of massive encryption of DNS traffic is very remote." [5]

From a surveillance perspective, DNS currently treats all information in the DNS database as public data. The content of queries and answers is typically not encrypted. This allows passive attackers to monitor the queries of users and see which services they are using and which websites they are visiting. For an active attacker, DNS facilitates locating potentially vulnerable services, which is the first step to their subsequent exploitation with commercially available 0-day attacks.

The discussions in the IETF now include proposals for "query minimization", Confidential DNS, DNS over TLS, DNSCurve and more radical proposals for alternative name system designs to improve privacy. All of these designs take different approaches in reducing the role of DNS as the ultimate source of meta data in the digital panopticon known as the Internet.

2 MORECOWBELL: Listening in on the DNS

Given that DNS today is an open book, it is not surprising that in a new set of top secret documents published by Le Monde, it is revealed that the MORECOWBELL (MCB) program of the American spy agency NSA monitors DNS as a source of information about the Internet (Figure 2). NSA's MORECOWBELL program uses a dedicated covert monitoring infrastructure to actively query DNS servers and perform HTTP requests to obtain meta information about services and to check their availability (Figure 1).

Despite the open nature of DNS, the NSA does so covertly (Figure 3) to ensure the thousands of DNS lookups every hour are not attributed to the US government (USG). In fact, the servers the NSA rented for the purpose of monitoring DNS and checking Web servers using HTTP are located in Malaysia, Germany and Denmark (Figure 4), allowing the NSA to perform the monitoring covertly and to get a more global view on DNS name resolution and service availability. While the slides only list these three countries, the

PACKAGEDGOODS non-attributable monitoring infrastructure that MORECOWBELL builds on is known to span machines in at least 13 other countries, as described previously by Der Spiegel in a set of slides describing the NSA's TREASUREMAP program. [14]

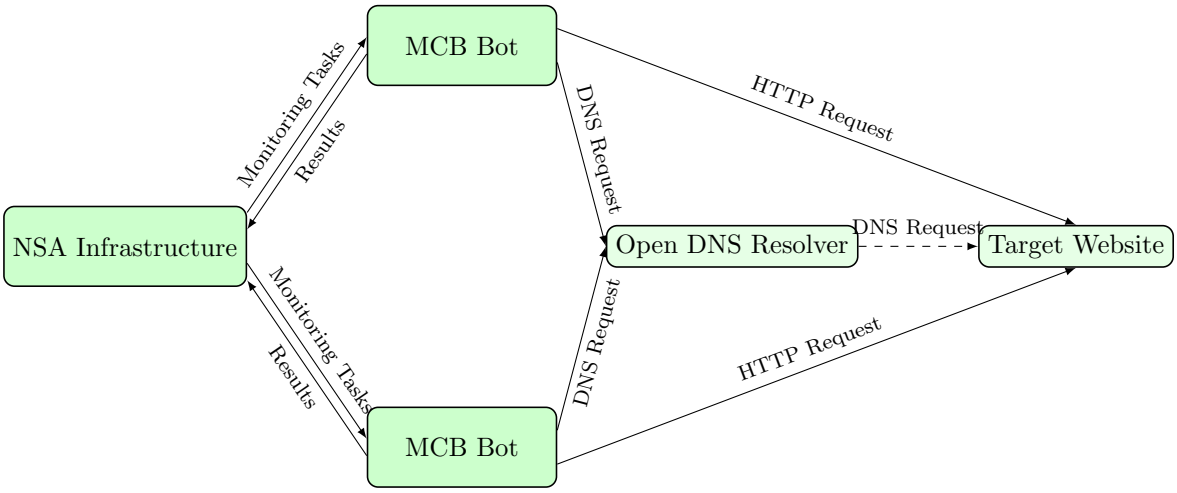


Figure 1: From `http://www.lemonde.fr/economie/visuel/2015/01/24/cowbells-nouvelles-revelations-sur-les-pratiques-de-la-nsa_4561547_3234.html`: NSA's MORECOWBELL infrastructure: a list of targets to monitor is deployed to geographically distributed bots performing DNS and HTTP requests against target websites to collect information about the availability of services. The resulting data are returned to the NSA in regular intervals.

What is interesting is that at the time, the NSA did not care much about the specific content of the Web servers or the DNS entries — as usual the NSA is after the meta data: The NSA wants to know if the DNS information has changed, and check on the availability of the service. The slides show that this simple check has some rather benign uses, for example it is used to monitor some of the US government's own websites.

A key justification for the need to make the active probing of DNS unattributable to the US government is most likely its use for "Battle Damage Indication" (Figure 5). Specifically, after "Computer Network Attacks (CNA)" are used against critical network infrastructure, the US may use such probes to confirm that its attacks have found their targets when the lights go out on the Internet systems, say of some foreign government. By monitoring for changes in the DNS, the attack could be repeated if the victim tries to shift its services to another system or network. By keeping the monitoring infrastructure covert and geographically distributed, the NSA gets a global view on the impact of an attack. This makes it harder for victims to identify the monitoring servers, which otherwise might enable victims to evade the attack by treating requests from monitors differently, an approach commonly used with DNS and known as *split view*.

While we have seen no proof for this, "battle damage indication" may also include damage from sources other than cyber attacks, such as bombing raids or cut cables. The US government does use the term "battle damage indication" for kinetic attacks:

> "BATTLE DAMAGE INDICATION
>
> The goal of this work is to develop low cost, innovative methods to quickly determine the affect an air-delivered munition has had on its intended target. This is especially important with regard to deeply buried targets where post-attack visual cues may be difficult to perceive. An onboard munition data link may be appropriate for obtaining an indication of damage to this type of target. Such a data link might be dependent on a trailing wire, or it might be completely wireless. Conversely, the **battle damage indicator** might be totally independent of the penetrating munition. The purpose of this study is to develop a low cost, efficient, and reliable means to

quickly provide the warfighter an accurate determination, or at least a reliable estimate, of the damage inflicted on a target - particularly a hardened and/or deeply buried one.

—Dr. Alex Cash AFRL/MNMI (850) 882-0391 cash@eglin.af.mil"[1]

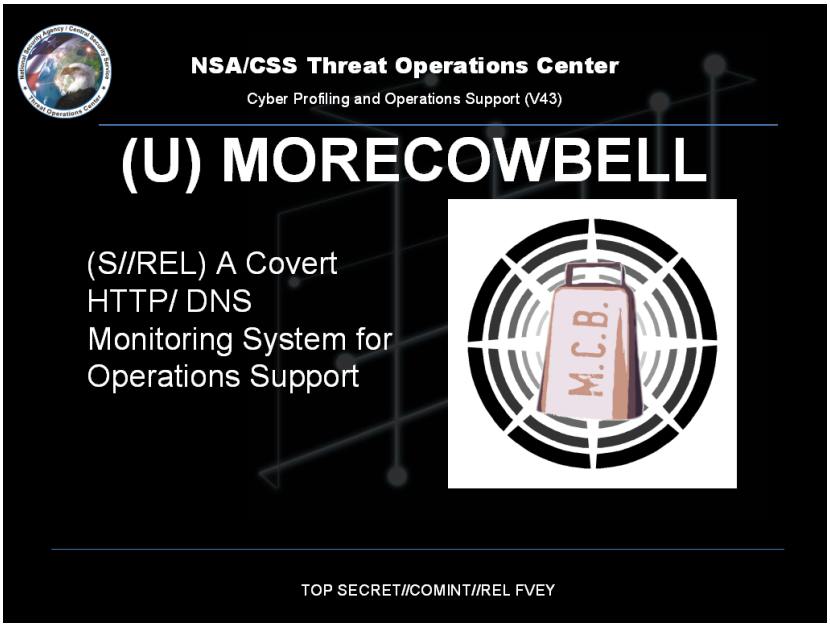


Figure 2: From `http://www.lemonde.fr/economie/visuel/2015/01/24/cowbells-nouvelles-revelations-sur-les-pratiques-de-la-nsa_4561547_3234.html`: MORECOWBELL: A Covert HTTP/DNS Monitoring System

The various documents of the NSA relating to DNS show that existing covert attacks on DNS go beyond mass surveillance and into a support role for active attacks. [20] With the revelation about the NSA's QUANTUMTHEORY family of projects with subprojects like QUANTUMDNS, we know that powerful attackers like nation states can not only eavesdrop DNS traffic but also inject DNS responses to modify the result of name resolution or make it even completely fail. [15] With DNS not providing confidentiality to protect a user's privacy, it is easy to create a profile of the users and their surfing behaviour on the Web. [9] This information could then also be used to perform QUANTUMTHEORY attacks against the target. NSA programmes like QUANTUMBOT have the purpose to monitor IRC botnets and detect computers operating as bots for a botnet and hijack the command and control channel to instrument the bots. These programmes are evaluated by the NSA to be *highly successful* according to their documents. [13]

Thus, the Internet community needs to work towards resolving the privacy and security issues with name resolution and the current Domain Name System (DNS). In the remainder of the article, we will review the existing DNS architecture and a range of current proposals that have been made to improve the security of this critical Internet service.

3 Background: DNS

The Domain Name System (DNS) is an essential part of the Internet as it provides mappings from host names to IP addresses, providing memorable names for users. DNS is hierarchical and stores name-value mappings

[1] Emphasis ours. Cited according to `http://www.darkgovernment.com/airforcedev.html`.

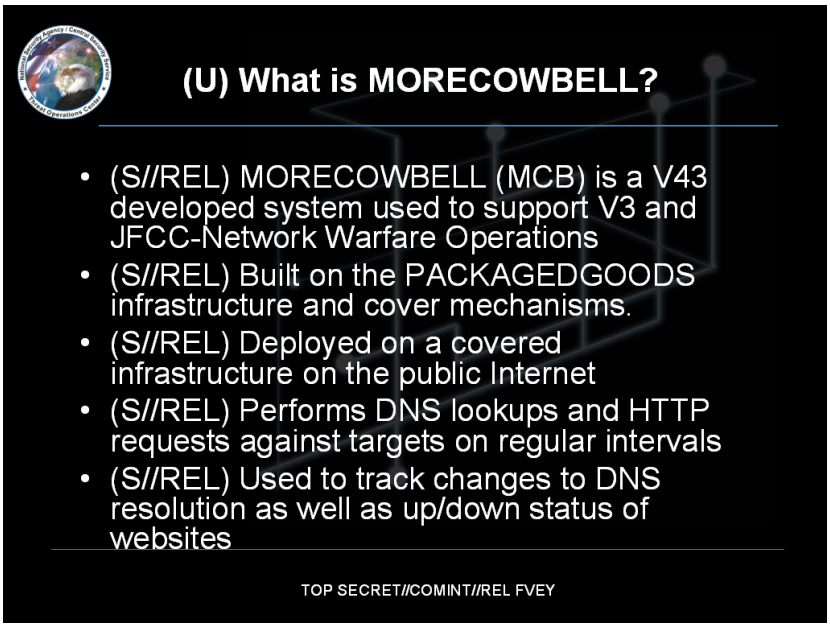


Figure 3: From `http://www.lemonde.fr/economie/visuel/2015/01/24/cowbells-nouvelles-revelations-sur-les-pratiques-de-la-nsa_4561547_3234.html`: What is MORECOWBELL.

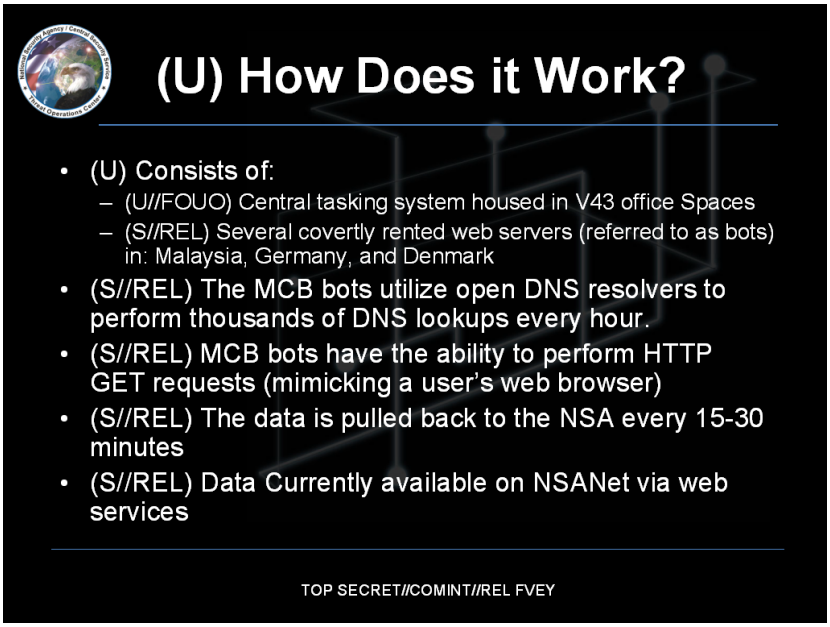


Figure 4: From `http://www.lemonde.fr/economie/visuel/2015/01/24/cowbells-nouvelles-revelations-sur-les-pratiques-de-la-nsa_4561547_3234.html`: How does MORECOWBELL work?

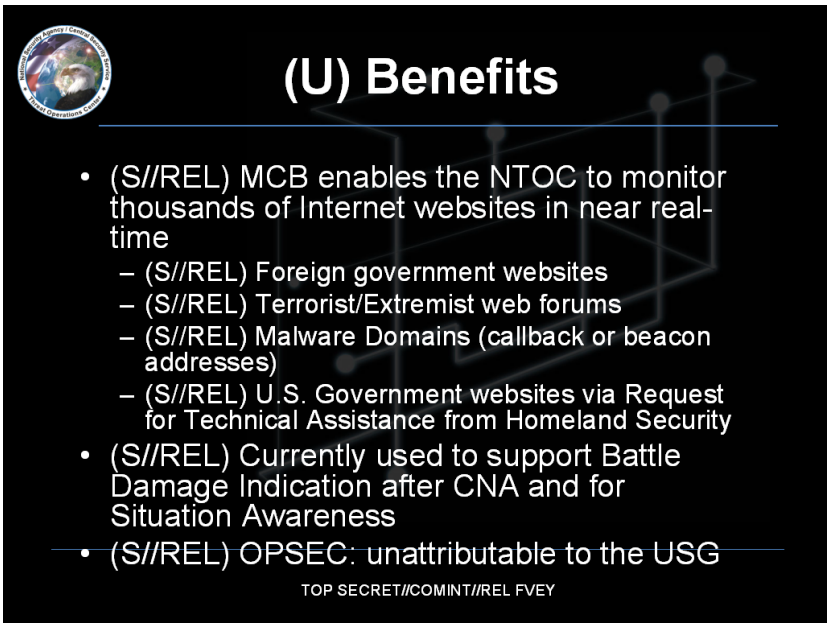


Figure 5: From `http://www.lemonde.fr/economie/visuel/2015/01/24/cowbells-nouvelles-revelations-sur-les-pratiques-de-la-nsa_4561547_3234.html`: "Benefits" of MORECOWBELL.

in so-called *records* in a distributed database. A record consists of a name, type, value and expiration time. Names consist of *labels* delimited by dots. The root of the hierarchy is the empty label, and the right-most label in a name is known as the top-level domain (TLD). Names with a common suffix are said to be in the same *domain*. The *record type* specifies what kind of value is associated with a name, and a name can have many records with various types. The most well-known record type is the "A" record, which maps names to IPv4 addresses.

The DNS database is partitioned into *zones*. A *zone* is a portion of the namespace where the administrative responsibility belongs to one particular authority. A zone has unrestricted autonomy to manage the records in one or more domains. Very importantly, an authority can delegate responsibility for particular *subdomains* to other authorities. This is achieved with an "NS" record, whose value is the name of a DNS server of the authority for the subdomain. The *root zone* is the zone corresponding to the empty label. It is managed by the Internet Assigned Numbers Authority (IANA), which is currently operated by the Internet Corporation for Assigned Names and Numbers (ICANN) under a contract with the National Telecommunications and Information Administration (NTIA). NTIA which is an agency of the US Department of Commerce and as such has also a tiny yet significant operational role: it checks every addition and change to the root zone file. The NTIA-IANA contract ends September 30, 2015 and the NTIA has announced its intent to transition out of its current role in an effort to handover the control to a global multistakeholder oversight. The root zone contains "NS" records which specify names for the authoritative DNS servers for all TLDs, such as ".de" or ".berlin".

Names in DNS are resolved using *resolvers*. Many modern operating systems do not provide a full implementation of a DNS resolver but only so called *stub resolvers*. These stub resolvers do not resolve names directly but forward the request to a *forward resolver*, which is typically provided by the Internet Service Provider (ISP), as shown in Figure 6. These resolvers resolve the name by first querying the root servers for the required name. If the queried DNS server cannot provide the final answer, it at least provides the resolver with an "NS" record which refers the resolver to the next DNS server. This *iterative* process is repeated, and terminates for sure when the resolver queries the *authoritative name server* which is responsible for a

particular domain. DNS strongly benefits from caching of DNS information: many *caching resolvers* store information previously requested to improve lookup performance. They use cached record data to skip some or all of the iterations, and thus can return information more quickly to the client.

With the use of forwarding resolvers, the IP address of the client is hidden from authoritative name servers. This gives the user a certain degree of privacy as it prevents operators of authoritative name servers to monitor the source of DNS requests. Naturally, the operators of the forwarding resolvers can still trivially monitor and censor users' requests. Passive dragnet monitoring with systems such as TURMOIL and XKEYSCORE are also able to see any part of the transaction that is available in the ingestion filter.

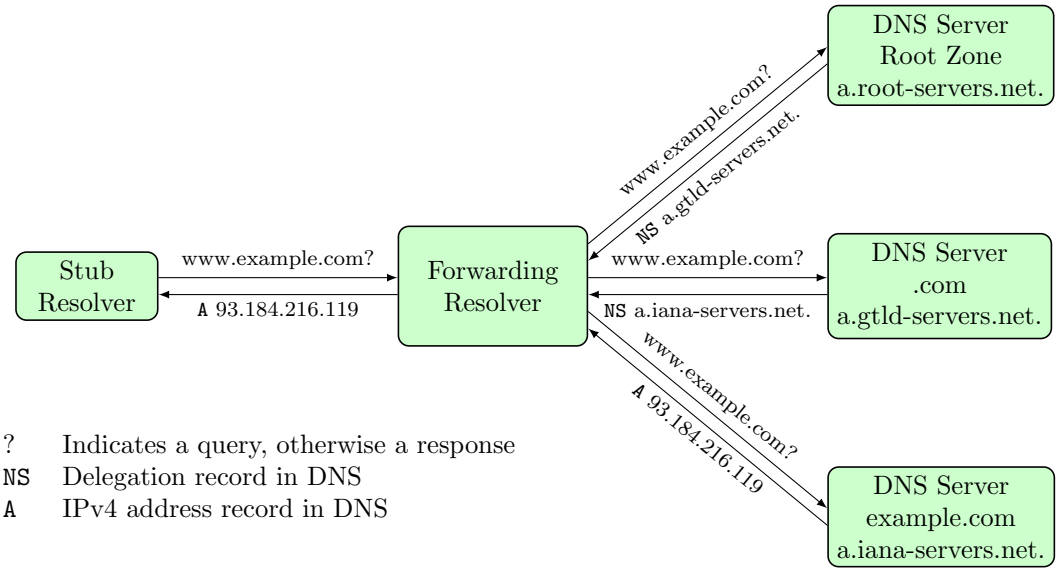


Figure 6: Resolving the name `www.example.com` with DNS. Many operating systems only provide minimal *stub resolvers* forwarding requests to full resolvers. To resolve a name, these resolvers start with querying the name servers of the root zone. If a server cannot provide the required information, it refers the resolver to the next server to query until the server *authoritative* for the respective zone is found.

4 DNSSEC

DNS was originally not designed to provide any security when used over an insecure network. The Domain Name System Security Extensions (DNSSEC) [1] add integrity protection and data origin authentication for DNS records. DNSSEC does not add confidentiality or denial-of-service protection, and thus does not protect at all against passive surveillance. It adds record types for public keys ("DNSKEY"), signer delegation ("DS"), and for signatures on resource records ("RRSIG"). Figure 7 illustrates the interactions among resolvers using DNSSEC. DNSSEC creates a hierarchical public-key infrastructure in which all DNSSEC operators must participate. It establishes a trust chain from a zone's authoritative server to the trust anchor, which is associated with the root zone. This association is achieved by distributing the root zone's public key out-of-band with, for example, operating systems. The trust chains established by DNSSEC mirror the zone delegations of DNS. With TLD operators typically subjected to the same jurisdiction as the domain operators in their zone, these trust chains are at risk of attacks using both legal and technical means.

The following are some of the most severe weaknesses that the DNS exhibits even in the presence of the DNS Security Extensions (DNSSEC). DNSSEC fails to provide any level of query privacy: the content of DNS requests and responses can still be read by any adversary with access to the communication channel and can subsequently be correlated with users, especially if the adversary can observe the link between the

user's stub resolver and the forward resolver. On a technical level, current DNSSEC deployment suffers from the use of the RSA crypto system (the root zone uses RSA-1024), which is required to be supported by every DNSSEC-enabled resolver and leads to large key sizes, especially as response includes the signatures for all of the signature schemes supported by the authoritative server. This can result in message sizes that exceed size restrictions on DNS packets, leading to additional vulnerabilities [8]. Finally, DNSSEC is not designed to withstand legal attacks. Depending on their reach, governments, corporations and their lobbies can legally compel operators of DNS authorities to manipulate entries and certify the changes. This is particularly relevant as DNSSEC maintains the hierarchical structure of DNS and thus places extensive trust in the root zone and TLD operators.

DNSSEC also effectively lifts the few traditional limitations on bulk acquisition of zone data, such as restrictions on zone transfers. Before DNSSEC, DNS zone administrators could disallow zone transfers, making it difficult for an adversary to systematically enumerate all of the DNS records in a zone. However, as DNS allows for negative replies (NXDOMAIN), DNSSEC needed a way to create a signed statement that records did not exist. As DNSSEC was designed to keep the signing key offline, "NSEC" records were introduced to certify that an entire range of names was not in use. By looking at the boundaries of those ranges, an adversary can quickly enumerate all names in a zone that are in use. An attempt to fix this via the introduction of "NSEC3" records was shown to be kaput before reaching significant deployment. As a result, DNSSEC makes it even easier for an adversary to discover vulnerable services and systems.

5 Query minimization

The recent discussions in the IETF to improve privacy in DNS include a proposal for so-called *query minimization* [7], which has good chances of being adopted quickly as it does not actually require changes to the DNS protocol. Query minimization would slightly improve privacy by changing forwarding resolvers to not send the full query to the DNS servers contacted in each resolution step. Instead, each DNS server only receives as much of the DNS name as is necessary for making progress in the resolution process (Figure 8). Consequently, the full name being queried is typically only exposed to the final authoritative DNS server.

Query minimization can simply be implemented by changing how forwarding resolvers construct their iterative queries. Query minimization may negatively impact caching, as at least in theory the full query may enable the DNS servers to respond with the ultimate answer, for example due to cached information from recursive queries, or because they are already the authority for the full name. Even with query minimization, forward resolvers still learn the full query and reply of a user.

Query minimization has the advantage that its deployment only requires changes to the forward resolver, and the disadvantage that this possible change to improve user privacy is entirely outside of their control. Query minimization can be combined with the various approaches to encrypt DNS traffic presented in the next sections; without query minimization, simply encrypting DNS traffic continues to expose the full query to many DNS servers. Finally, Verisign Inc. may hinder query minimization adoption by exploiting the software patent racket [18].

6 T-DNS: DNS over TLS

Discussions to use Transport Layer Security (TLS) for encrypting DNS traffic were previously often rejected because of the performance loss associated with such a change. In the context of a recent IETF draft on starting DNS over TLS, the authors point out that using TLS would not only be beneficial in supporting privacy, but also that switching to TCP — and therefore from connectionless UDP to connection-oriented TCP — might help mitigate against amplification attacks on (or by) DNS servers. [10]

By re-using a TCP connection for multiple DNS requests with moderate timeouts, pipelining requests and allowing out of order processing, the T-DNS proposal promises reasonable performance despite the overheads from TCP and TLS.

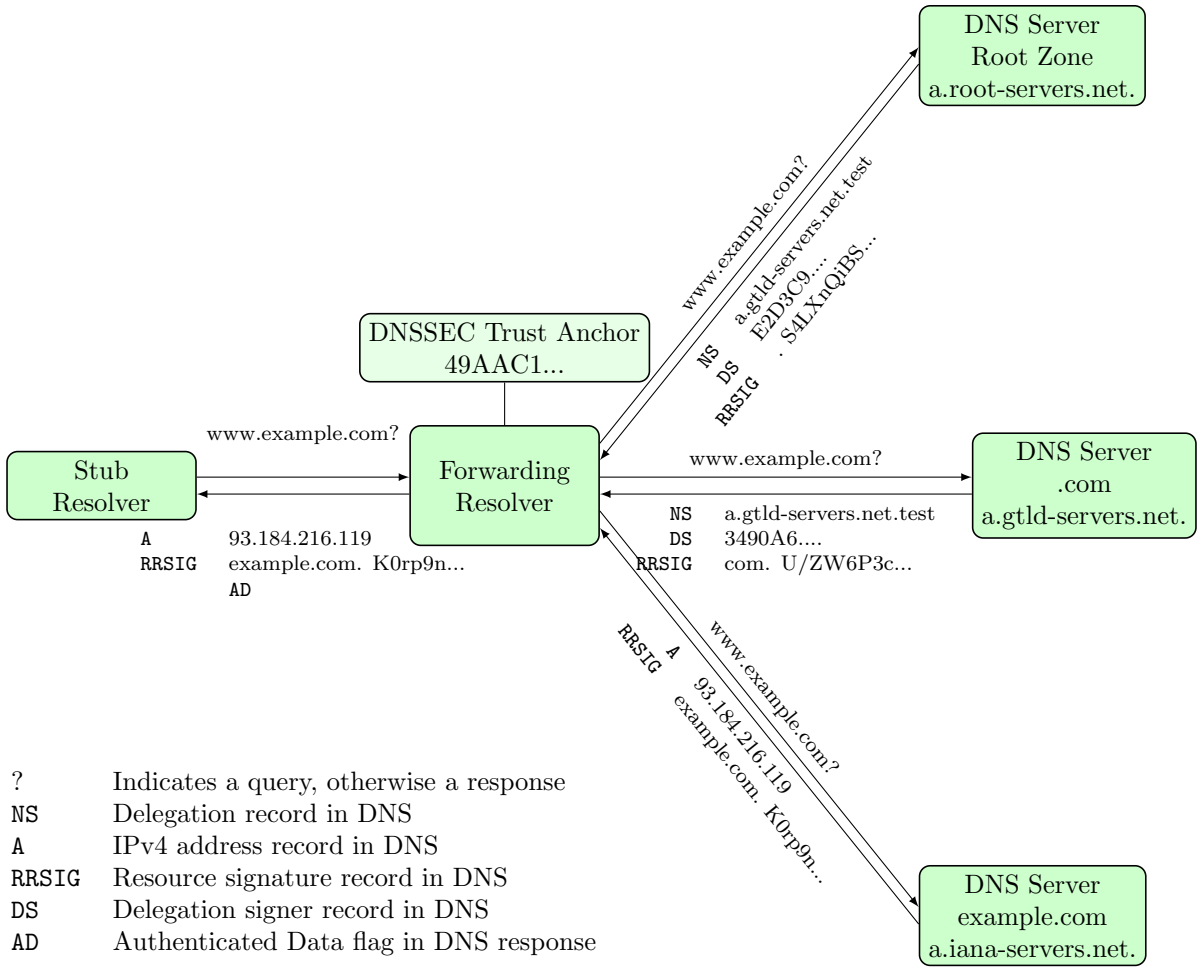


Figure 7: Resolving the name `www.example.com` with DNS and DNSSEC: information returned by name servers is cryptographically signed to ensure authenticity and integrity. This information is stored in "RRSIG" records and information about the parent zone stored in "DS" records. A resolver can verify a signature by following this trust chain and using the *trust anchor* shipped out-of-band. Stub resolvers cannot verify this chain and the resolver therefore indicates to the stub resolver that it checked authenticity by setting the `AD` bit in the response given to the client.

However, even if TLS were to be deployed for DNS, this would still leak meta data, allowing third parties to easily determine which DNS data a user accesses: In the IETF proposal, TLS is combined with the use of forward resolvers, and while forward resolvers hide the user's IP address from the DNS servers, they themselves have to be trusted to not spy on the user. Furthermore, TLS itself does not have the best security track record, with dozens of issues in recent years ranging from high-profile certificate authority compromises to broken implementations and insecure cipher modes.

TLS is not the only possible method for encrypting DNS queries and replies as they traverse the network. DNSCurve and Confidential DNS are alternative proposals to protect the content of DNS queries and replies from network-level monitoring.

DNS-over-TLS is available the Unbound DNS server.

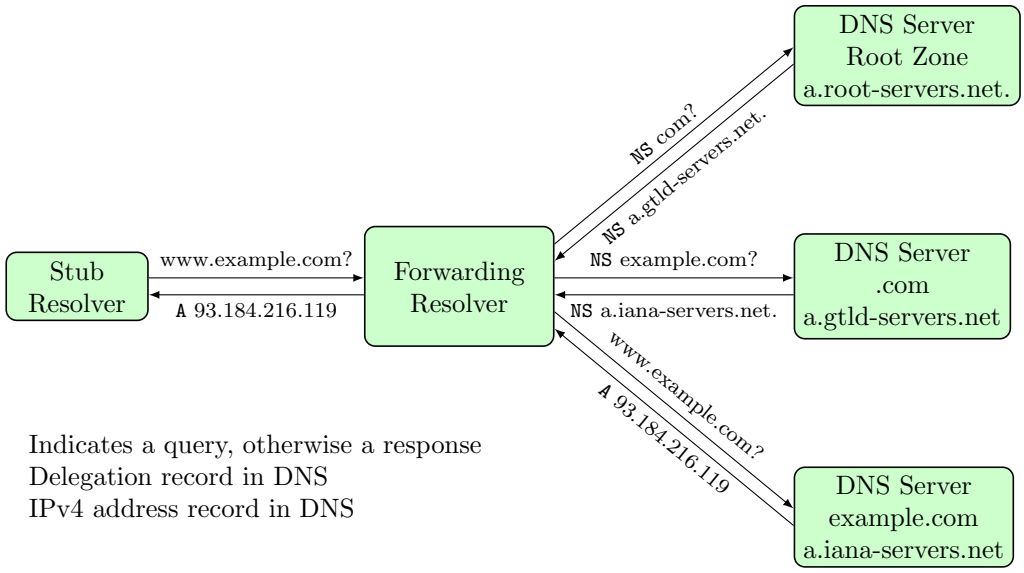


Figure 8: With query minimization, resolving the name `www.example.com` no longer exposes the full name and query type to the root zone and the `.com` authority. Naturally, this scheme still leaks quite a bit of sensitive information to the TLD's DNS server. Furthermore, the effect is even weaker in practice, as root zone is already often not contacted as information about TLD name servers is typically cached at forwarding resolvers.

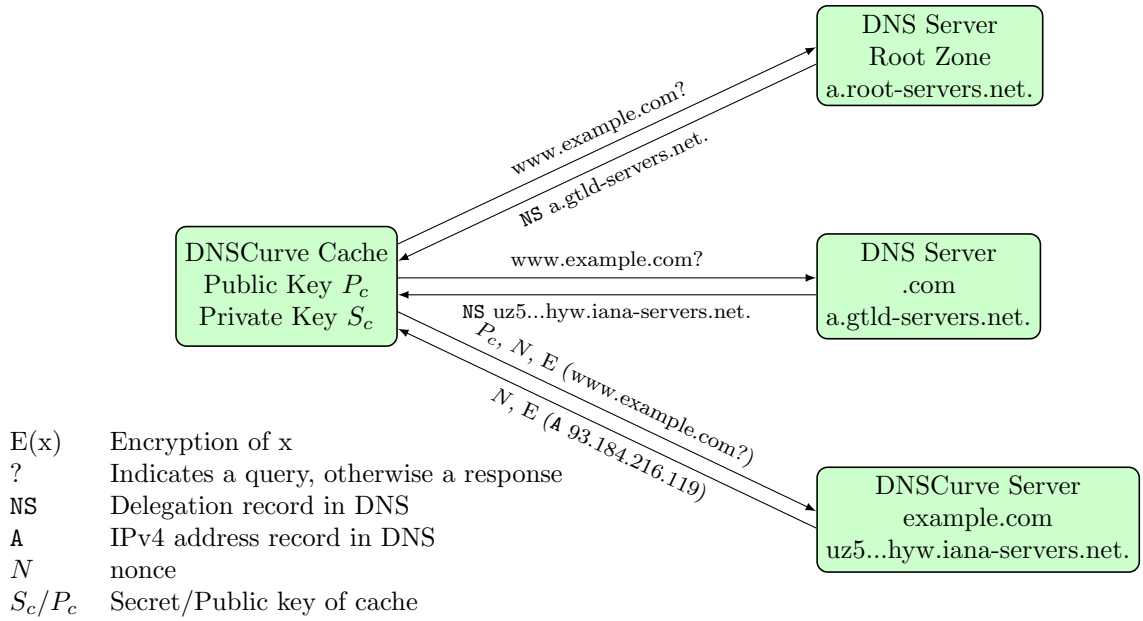
7 DNSCurve

The first practical system that improves confidentiality with respect to DNS queries and responses was DNSCurve [3]. In DNSCurve, session keys are exchanged using Curve25519 [2] and then used to provide authentication and encryption between caches and servers. DNSCurve improves the existing Domain Name System with confidentiality and integrity without the need to create expensive signatures or (D)TLS sessions. Specifically, DNSCurve achieves the same round trip time (RTT) as DNS by embedding the public key of the server in the "NS" record.

DNSCurve creates an authenticated and encrypted association between a *DNSCurve server* and a *DNSCurve cache*, the latter being a caching recursive DNS resolver running at the endpoint instead of a DNS stub resolver (Figure 9). As DNSCurve does not use signatures, the DNSCurve cache cannot prove the authenticity of the cached records to other users, limiting the utility of each cache to the respective endpoint.

While in DNSCurve the user no longer has to trust a forward resolver, the endpoint's IP address is now directly exposed to the authoritative DNS servers: it is no longer obscured by forward resolvers operated by network service providers. Thus, DNSCurve can increase privacy against an adversary monitoring DNS traffic on intermediary systems or with other cable tapping, but reduces privacy with respect to authoritative DNS servers, as they learn both the full query and the identity (IP address) of the user. Another commonly voiced concern about DNSCurve is the need to keep private keys online. DNSCurve also cannot protect against censorship, as certain governments continue to effectively control the hierarchy of registrars and can thus make domains disappear. With respect to attacks from the NSA, DNSCurve only helps users against passive surveillance on the wire by protecting the confidentiality of at least the DNS payload.

With DNSCurve, DNS servers remain a juicy target for mass surveillance. Furthermore, as with DNS, the well-known and easily located DNS servers remain a target and confirmation vector for attacks on critical infrastructure. With DNSCurve, the need for online public key cryptography by the DNS authorities may open up an additional vulnerability to computational denial of service attacks if a small CPU is used to handle a high-speed link.



E(x)	Encryption of x
?	Indicates a query, otherwise a response
`NS`	Delegation record in DNS
`A`	IPv4 address record in DNS
N	nonce
S_c/P_c	Secret/Public key of cache

Figure 9: Resolving the name `www.example.com` with DNSCurve. With DNSCurve, the resolving cache and the DNSCurve server exchange a shared secret to encrypt their communication. The DNSCurve server's public key is encoded in the name of the name server itself using Base32. When a DNSCurve cache resolves a name and finds the name server to support DNSCurve, the cache creates a shared secret based on the server's public key, the cache's private key, and a one-time nonce. The cache sends its public key, the nonce and the query encrypted with the shared secret. The server will respond with the result of the query encrypted with the shared secret. The first two lookups to the root zone and the ".com" TLD do not use DNSCurve in the illustration as those currently do not support DNSCurve.

DNSCrypt

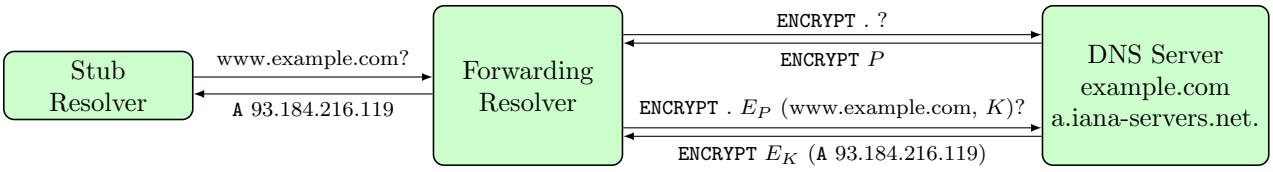
DNSCrypt is an unstandardized but documented protocol largely based on DNSCurve. It protects the end user's stub resolver queries from network surveillance and tampering. At it is based on DNSCurve, it does not solve any of the major other privacy or security issues present in DNS. The largest known resolver to support DNSCrypt is OpenDNS. There are a number of open DNSCrypt resolvers run by the DNSCrypt community. Today, DNSCrypt remains the most widely deployed DNS encryption protocol designed to prevent surveillance of end users from the network. However, it only helps to solve half of the privacy problem, and it is not widely adopted or standardized.

8 Confidential DNS

Another recent IETF draft suggests an alternative method for adding encryption to DNS that actually uses the main extension mechanism of DNS, the introduction of additional record types, to encrypt DNS traffic [21]. With Confidential DNS, a new "ENCRYPT" record type is introduced to provide the necessary public key that would allow the recursive resolver to encrypt the connection to the DNS server. This "ENCRYPT" record contains the public key of the DNS server to be used to encrypt communication initiated by the resolver. This avoids the hack used by DNSCurve where the public key was added into the "NS" response of the delegating zone.

The current draft supports two different operation modes: an *opportunistic* mode which is easier to realize since it does not require major changes to DNS infrastructure and an *authenticated* mode, where a domain's

public keys are also stored in the respective parent zone, thus requiring support from the parent zone's DNS infrastructure.



?	Indicates a query, otherwise a response
.	Query for the root zone
P	Public key of server
K	Encryption Key
E_P (x)	Encryption of x with P
E_K (x)	Encryption of x with K
`A`	IPv4 address record in DNS
`ENCRYPT`	Encrypt record in DNS

Figure 10: Resolving the name `www.example.com` with opportunistic Confidential DNS. The resolver retrieves the DNS servers public key querying for the new "ENCRYPT" record. This public key can then be used to encrypt the query to the server. The resolver sends the query encrypted with the server's public key containing the query and the key to encrypt the reply with.

With the opportunistic mode, the public key is no longer associated with the parent zone and instead served separately in the clear and possibly without authentication as a record with the target zone. As a result, Confidential DNS using the "ENCRYPT" record only supports so-called *opportunistic encryption* — which is newspeak for encryption that is trivially bypassed by a man-in-the-middle attack, as it uses unauthenticated keys for encryption.

The use of a new record type also creates the opportunity for the necessary complexity of a committee-engineered solution: Confidential DNS can use symmetric or asymmetric cryptography, and sports support for 512-bit RSA and AES in CBC mode (which was recently used to finally kill off SSL3 [11]). The draft fails to set a strong minimum baseline and ensure that this minimum will be updated to reflect new security considerations in due course.

The draft on Confidential DNS provides a second method to achieve "real" authenticated encryption by storing a domain's public key in the respective parent zone. To do so, Confidential DNS extends DNSSEC's Delegation Signer ("DS") resource records with a hack to provide the encryption key for the zone, similar to the hack of the "NS" record used by DNSCurve. The draft provides for a variety of failure modes, such as "fallback to insecure" allowing clients to relapse to insecure modes with "leaps of faith" even after secure connections used to be available. Combined with the possibility of "fallback to insecure" due to the possibility of unsupported cryptographic algorithms, Confidential DNS provides unpredictable security instead of any kind of strict assurances. Making no guarantees and offering many options facilitates deployment and migration, which is the guiding principle for the industry-driven IETF engineering process.

9 Namecoin

Alternative peer-to-peer name systems provide more radical solutions to secure name resolution. Timeline-based systems in the style of Bitcoin [12] have been proposed to create a global, secure and memorable name system [16]. Here, the idea is to create a single, globally accessible timeline of name registrations that

is append-only. Timeline-based systems rely on a peer-to-peer network to manage updates and store the timeline. In the Namecoin system [17], modifications to key-value mappings are attached to transactions which are committed to the timeline by mining. Mining is the use of brute-force methods to find (partial) hash collisions with a state summary (fingerprint) representing the complete global state — including the full history — of the timeline.

Given two timelines with possibly conflicting mappings, the network accepts the timeline with the longest chain as valid, as it represents the largest expense of computational power. This is supposed to make it computationally infeasible for an adversary to produce an alternative valid timeline. This assumes limited computational power and may not actually be binding for certain adversaries.

To perform a lookup for a name with Namecoin, the client has to check the timeline if it contains an entry for the desired name and check the timeline for correctness to ensure that the timeline is valid. To do so, the user has to possess a full copy of the timeline (Figure 11), which had a size of 2 GB in November 2014.[2] Alternatively, users may use a trusted name server participating in the Namecoin network.

Namecoin can improve user privacy if the full block chain is replicated at the user's end system. In this case, resolving a name does not involve the lookup and is thus perfectly private. However, replicating the full block chain at each user may be impractical for some devices should Namecoin ever grow to be a serious competitor for DNS. Namecoin also does not protect the zone information from monitoring, and in particular zone enumeration is trivial. However, the decentralised nature of Namecoin does ensure that at least battle damage indication against a name server no longer makes sense.

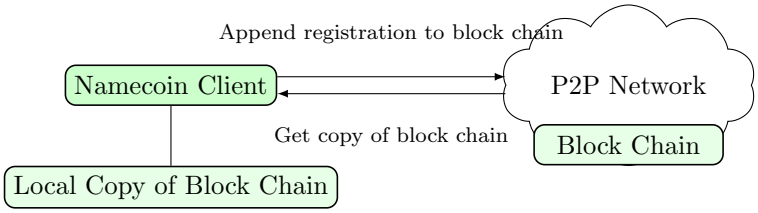


Figure 11: The Namecoin name system is decentralized and uses a peer-to-peer network. To achieve a consensus about names registered, Namecoin uses a *block chain* stored in the peer-to-peer network. To register a name, clients have to perform some computational work to get their name appended to the chain. To resolve a name, clients have to possess a full copy of the block chain and search for the name to resolve in the block chain.

10 The GNU name system

The authors of this article are working on the GNU Name System (GNS) [19], which is a more radical proposal to address DNS privacy and security issues, and which like Namecoin significantly departs from DNS's name resolution process. The GNS resolution process does not use resolvers querying DNS authorities. Instead, GNS uses a peer-to-peer network and a distributed hash table (DHT) to enable resolvers to lookup key-value mappings.

GNS is privacy-preserving since queries and responses are encrypted such that even an active and participating adversary can at best perform a confirmation attack, and otherwise only learn the expiration time of a response. Note that the queries and responses themselves are encrypted, not the connections between a resolver and some authority. As all replies are not just encrypted but also cryptographically signed, peers in the DHT cannot tamper with the results without immediate detection.

Due to the use of a DHT, GNS avoids DNS complications such as glue records and out-of-bailiwick lookups. In GNS, the labels of a name correspond precisely to the lookup sequence, making the complete trust path obvious to the user. Finally, the use of a DHT to distribute records also makes it possible for GNS

[2] `https://bitinfocharts.com/de/namecoin/`

authorities to operate zones without visible, attributable critical infrastructure that could be used for battle damage indication.

GNS can securely resolve names to any kind of cryptographic token. Thus, it can be used for addressing, identity management and as an alternative for today's battered public key infrastructures.

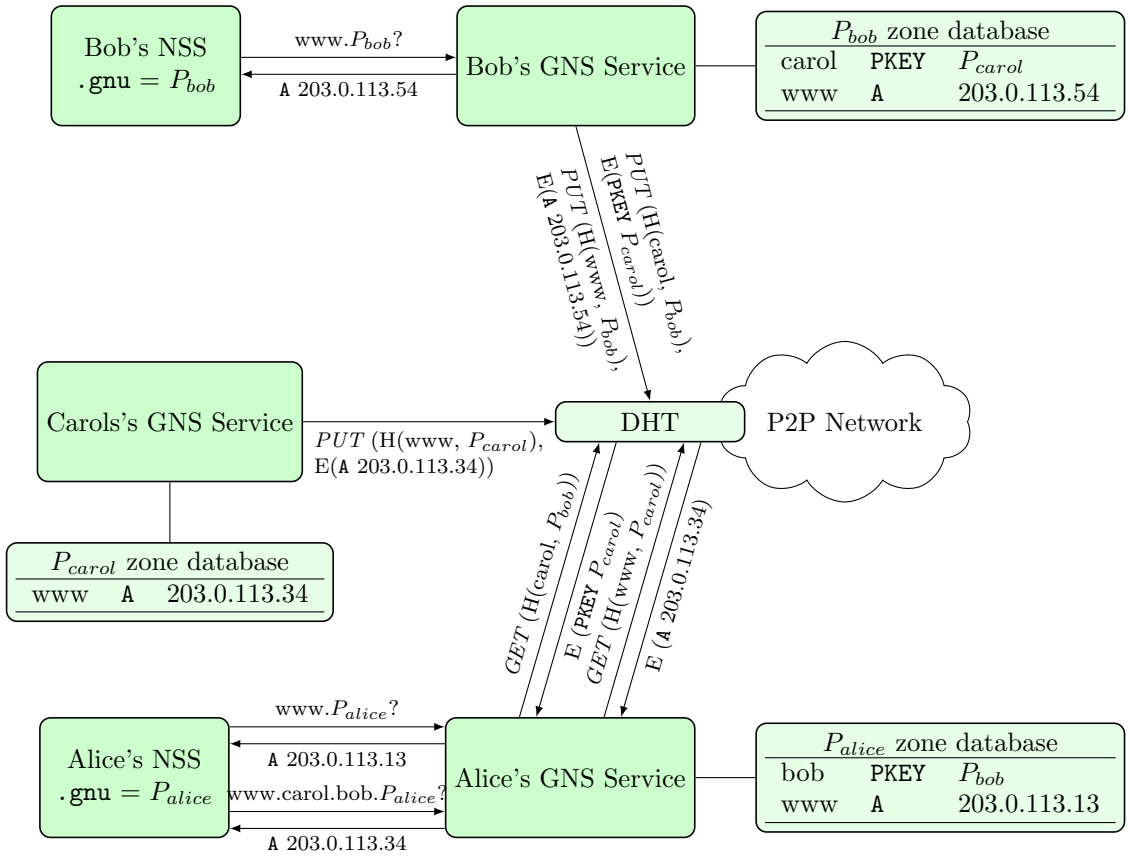


Figure 12: The GNU name system: with GNS, every user maintains their own databases containing record sets under labels organized in zones. A zone is referenced by a public-key pair. Here Alice, Bob and Carol have web servers all reachable under `www.gnu`. For Alice `www.gnu` resolves to a different address than for Bob or Carol, as their respective local name service switches (NSS) associate a user-specific public key with `.gnu`. To allow other users to resolve the names, a user's public zone information is encrypted and published in a DHT under an obfuscated query key. A user can *delegate* to another user's namespace from his local namespace to resolve foreign names. Alice can access Bob's namespace by delegating control over the name `bob` to P_{bob} in her namespace using a GNS-specific "PKEY" record. This way Alice can access Carols's webserver using the name `www.carol.bob.gnu`.

10.1 Names, zones and delegations

A GNS zone is a public-private key pair and a set of associated records. The GNS name resolution process basically resolves a chain of public keys. In the absence of a widely recognized and operational *root zone*, but also as an inherent alternative to hierarchical addressing, GNS uses the pseudo-TLD ".gnu" to refer to the user's own zone, which is called the *master zone*. Each user can create any number of zones, but one must be designated as the master zone. Users can freely manage mappings for the labels in their zones. Most importantly, they can delegate control over a subdomain to any other zone (including those operated by other

users) using a "PKEY" record, which simply specifies the public key of the target zone. "PKEY" records are used to establish the aforementioned delegation path. Due to the use of a DHT, it is not necessary to specify the address of some system that is responsible for operating the target zone. Record validity in the DHT is established using signatures and controlled using expiration values.

10.2 Cryptography for privacy

To enable other users to look up records of a zone, all records for a given label are stored in a cryptographically signed block in the DHT. To maximize user privacy when using the DHT to look up records, both queries and replies are encrypted and replies are signed using a public key derived from the public key of the zone and the label (Figure 12). Any peer can easily validate the signature but not decrypt the reply without prior knowledge of the public key and label of the zone. Consequently, users can use passwords for labels or use public keys that are not publicly known to effectively restrict access to zone information to authorized parties.

Due to the use of a DHT, all GNS queries go to the same fully decentralised and shared global infrastructure instead of operator-specific servers. This makes it impossible to target a zone-specific server because all machines in the DHT are jointly responsible for all zones — in fact, the key-value pairs do not reveal which zone they belong to. At the same time, encryption and authentication of the records is critical as it helps protect the users from effective censorship or surveillance. However, unlike the other less radical proposals to overhaul DNS, deploying GNS will be a significant challenge: GNS requires more significant changes to software, as well as a community effort to operate a DHT as a new public infrastructure.

11 Political developments

The Domain Name System and IANA's IP address registry are the two key databases that tie together the global Internet. Given the reckless exploitation of the Internet as a surveillance machine by its current steward, the US government, the trend for "national internets" can further accelerate.

Some countries, especially those that use more heavy-handed approaches to Internet censorship such as China and Iran, have closed off their national internet as a means to restrict the flow of information for some time. However, especially since Snowden's revelations, debates about national routing and national infrastructure building have flurried even in countries that have been viewed traditionally as strong US allies: Brazil spoke of obligating big Internet platforms to establish a presence in Brazil and to confine Brazilian data within Brazil. In Germany there were calls for national or Schengen routing. Disvestiture of the IANA function, demanded since 2003 at the first UN conference on the Information Society (World Summit of the Information Society), finally was announced by the NTIA in April 2014.

As usual, the spy agencies are ahead of the game when it comes to isolating themselves: both the NSA and GCHQ are known to internally operate a non-public DNS system with their own unofficial TLDs, `.nsa` and `.gchq`. However, unlike the developers of Tor, the spy agencies have not yet followed RFC 6761 to try to reserve those names.

The strategic use of non-public TLDs to make Internet services less accessible is logical, and a clear step towards Internet "Balkanization". At a global scale, this trend is not appreciated by the US government, as decentralisation may limit the reach of US surveillance. To ward against this development, a "multi-stakeholder" process is used to obscure the issue of who runs the system, and to deflect the question of accountability while maintaining control indirectly via the "stakeholders".

In recent years, ICANN tried to increase competition in domain name offerings with the proliferation of GTLDs. However, it remains a US incorporated organization which controls process and profits. Thus, a key question is if ICANN/IANA or some successor organization will — under whatever governance structure — continue to sit at the helm. Alternatively, we may see technologies developed and deployed that fully decentralize the allocation of addresses and names, making a global steward and the associated political fighting over control unnecessary. It appears that the Internet is headed in both directions at the same time.

12 Conclusion

In "Culture Is Our Business" Marshall McLuhan stated presciently:

> "World War III is a guerrilla information war with no division between military and civilian participation."

It appears that his prediction from 1970 remains relevant when we consider the Internet's architecture as it is woven through our everyday lives.

DNS was never designed with privacy or security as a design goal. In the battle of nation states for global dominance, any Internet infrastructure that serves a specific audience is a target for state attackers. Critical infrastructure needs to be logically decentralised and should ideally be shared globally to reduce the value of harming it. Merely encrypting DNS and Web traffic may not sufficiently reduce the effectiveness of targeted attacks against insecure designs.

While awareness exists in the DNS community that privacy is an issue, the diverse interests in the community make it virtually impossible to quickly make significant progress by consensus. Modifications to a deployed system like DNS, following the general ossification trend of the Internet, are met with inertia and usually end up with death by committee, as any significant change could not only result in serious malfunctioning, but may also impact somebody's business model or nation state interest.

In a world where the NSA hunts system administrators[3] and ICANN becomes an easy victim[4], the proposed band aids by the IETF fail to address the scope of the problem: surveillance of users, commercial censorship and the danger of a new reign of terror where DNS operators are legitimate targets must be addressed better in future designs.

Acknowledgements

We thank Laura Poitras, Ludovic Courtès, Dan Bernstein, Luca Saiu and Hellekin Wolf for their help and support in preparing this report.

References

[1] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS security introduction and requirements. *IETF RFC 4033*, March 2005.

[2] Daniel J. Bernstein. Curve25519: new Diffie-Hellman speed records. In *In Public Key Cryptography (PKC), Springer-Verlag LNCS 3958*, 2006.

[3] Daniel J. Bernstein. DNSCurve: Usable security for DNS. `http://dnscurve.org/`, 2008.

[4] Internet Architecture Board. IAB statement on Internet confidentiality. `https://mailarchive.ietf.org/arch/msg/ietf-announce/ObCNmWcsFPNTIdMX5fmbuJoKFR8`, 2014.

[5] S. Bortzmeyer. Possible solutions to DNS privacy issues. `http://tools.ietf.org/html/draft-bortzmeyer-dnsop-privacy-sol-00`, December 2013.

[6] S. Bortzmeyer. DNS privacy considerations. `https://datatracker.ietf.org/doc/draft-ietf-dprive-problem-statement/`, 2014.

[7] S. Bortzmeyer. DNS query name minimisation to improve privacy. `https://tools.ietf.org/html/draft-bortzmeyer-dns-qname-minimisation-02`, May 2014.

[8] Amir Herzberg and Haya Shulman. Fragmentation considered poisonous: or one-domain-to-rule-them-all.org. In *CNS 2013. The Conference on Communications and Network Security*. IEEE, 2013.

[3] `http://cryptome.org/2014/03/nsa-hunt-sysadmins.pdf`

[4] `http://www.heise.de/security/meldung/Erfolgreicher-Angriff-auf-Internet-Verwaltung-ICANN-2499609.html`

[9] Srinivas Krishnan and Fabian Monrose. Dns prefetching and its privacy implications: When good things go bad. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, LEET'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.

[10] Allison Mankin, Duane Wessels, John Heidemann, Liang Zhu, and Zi Hu. t-DNS: DNS over TCP/TLS. `http://www.isi.edu/ant/tdns/`, 2014.

[11] Bodo Möller, Thai Duong, and Krzysztof Kotowicz. This POODLE bites: exploiting the SSL 3.0 fallback. `https://www.openssl.org/~bodo/ssl-poodle.pdf`, 2014.

[12] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. `http://bitcoin.org/bitcoin.pdf`, 2008.

[13] Anonymous (NSA). There is more than one way to quantum. `https://www.documentcloud.org/documents/1076891-there-is-more-than-one-way-to-quantum.html#document/p1`, 2014.

[14] NSA/CSS Thread Operations Center (NTOC). Bad guys are everywhere, good guys are somewhere! `http://www.spiegel.de/media/media-34757.pdf`, 2014.

[15] Redacted (NSA, S32X). QUANTUMTHEORY. `https://firstlook.org/theintercept/document/2014/03/12/nsa-gchqs-quantumtheory-hacking-tactics/`, 2014.

[16] Aaron Swartz. Squaring the triangle: Secure, decentralized, human-readable names. `http://www.aaronsw.com/weblog/squarezooko`, 2011.

[17] `http://dot-bit.org/`. The Dot-BIT project, a decentralized, open DNS system based on the Bitcoin technology. `http://dot-bit.org/`, 2013.

[18] Inc. Verisign. Verisign, Inc.'s statement about IPR related to draft-bortzmeyer-dns-qname-minimisation-02. `https://datatracker.ietf.org/ipr/2469/`, October 2014.

[19] Matthias Wachs, Martin Schanzenbach, and Christian Grothoff. A censorship-resistant, privacy-enhancing and fully decentralized name system. In *13th International Conference on Cryptology and Network Security (CANS 2014)*, pages 127–142, 2014.

[20] Nicholas Weaver. A close look at the NSA's most powerful Internet attack tool. *Wired*, 2014.

[21] W. Wijngaards. Confidential DNS. `http://tools.ietf.org/html/draft-wijngaards-dnsop-confidentialdns-00`, 2013.