On the Feasibility of a Censorship Resistant Decentralized Name System

Matthias Wachs Martin Schanzenbach Christian Grothoff
Email: {wachs,schanzen,grothoff}@in.tum.de

Technische Universität München

Abstract. A central problem on the Internet today is that key infrastructure for security is concentrated in a few places. This is particularly true in the areas of naming and public key infrastructure. Secret services and other government organizations can use this fact to block access to information or monitor communications. One of the most popular and easy to perform techniques is to make information on the Web inaccessible by censoring or manipulating the Domain Name System (DNS). With the introduction of DNSSEC, the DNS is furthermore posed to become an alternative PKI to the failing X.509 CA system, further cementing the power of those in charge of operating DNS.

This paper maps the design space and gives design requirements for censorship resistant name systems. We survey the existing range of ideas for the realization of such a system and discuss the challenges these systems have to overcome in practice. Finally, we present the results from a survey on browser usage, which supports the idea that delegation should be a key ingredient in any censorship resistant name system.

1 Introduction

> "The Domain Name System is the Achilles heel of the Web. The important thing is that it is managed responsibly." – Tim Berners-Lee

Recent global news [1] on extensive espionage and cyberwar efforts by the US government and its "second class" allies, in particular the UK, have been met by some with calls to "encrypt everything" [2]. While this is hardly a solution for governments monitoring communication patterns (meta-data) and accessing data stored in plaintext at major service providers, encryption is clearly the baseline defense against government intrusions and industrial espionage [3]. However, encryption is useless without a secure public key infrastructure, and existing PKIs (DNSSEC, X.509 or the German ePa) are easily controlled and bypassed by major intelligence agencies. To realize the vision of an Internet where dissent is possible, we thus need to create an alternative, decentralized method for secure name resolution. Given a secure decentralized name system, we can then begin to build secure decentralized solutions for communication (e-mail, voice) and social networking applications and liberate the network from comprehensive government surveillance.

Today, the Domain Name System (DNS) is a key service for the Internet. DNS is primarily used to map names to IP addresses. Names are easier to remember for humans than IP addresses, which are used for routing but generally not meaningful for humans. DNS thus plays a central rôle for access to information on the Web; consequently, various institutions are using their power — including legal means — to censor or modify DNS information. These attacks on the DNS are sufficient to threaten the availability and integrity of information on the Web [4]. Furthermore, tampering with the DNS can have dramatic side effects, as a recent study about the worldwide effects of China's DNS censorship in China shows [5]: Chinese censorship of DNS can result in invalid results for parties that are far away from China. Many institutions like the European Parliament [6] or the OpenNet initiative [7] realize the dangers arising from DNS censorship, especially with respect to the importance that obtaining free information on the Web had in in recent events as the Arab Spring or the Green Revolution in Iran.

Significant efforts have been made to harden DNS against attacks with DNSSEC providing data integrity and authenticity. These efforts are limited in their effect against institutional attackers performing censorship using their oppressive or legal powers to modify the results of an DNS request; even if end-to-end security between authoritative DNS servers and DNS clients were deployed, legal attacks coercing DNS authorities to hand over control of names would still be possible. A hypothetical mandatory DNSSEC deployment with end-to-end security providing integrity and authenticity cannot prevent or even detect such attacks, as the censored results would still be signed by a valid (albeit coerced) authority.

To prevent such attacks, we need a censorship resistant name system ensuring availability and resilience of names. For such a censorship resistant name systems, this paper advocates a solution in line with the ideas of the GNU project. Richard Stallman, founder of the GNU project, writes [8]: "When a program has an owner, the users lose freedom to control part of their own lives." Similarly, ownership of a name implies the existence of some authority to exercise control over the property, and thus implies the possibility of coercion of that authority. Cryptographic identifiers can be created without the need for an authority; similarly, when users locally assign values to private labels, as done in *petname* systems, such personal labels also cannot be owned or confiscated.

Based on these two central concepts, this paper discusses the design space and requirements for the "GNU Name System", which would be a fully-decentralized, ownership-less name system that provides censorship-resistance, privacy and security against a wide range of attacks by strong adversaries. We also discuss challenges alternative name systems face in practice and present the results of a survey characterizing common usage patterns of name systems on the Web.

2 Requirements Analysis

To analyze the requirements a censorship resistant name system has to fulfill, we start with a practical adversary model and the attacks a system has to withstand. Based on these, we then develop functional requirements for a censorship resistant name system.

2.1 Adversary Model

The adversary used in this paper is modeled after nation state trying to limit access to information on the Internet. Our adversary can participate in any role in the system and can also assume multiple identities (Sybils) without an upper bound in relation to the total number of participants. The adversary can take over control of names using judicial or executive powers and is allowed to have more computational power then all benign users. This model excludes the use of a trusted third party.

On the other hand, the adversary cannot break cryptographic primitives and not prevent the usage of cryptography or encrypted communication. The adversary is also not able to take direct control of the systems of individual users, or at least if he does so, the system does not have to protect the users that are directly affected by such actions. As far as network communication is concerned, we assume that the adversary cannot generally prevent communication between benign participants.

Our adversary's goal is to prevent access to information on the Web by affecting the name resolution process, either by making it fail or by changing the value associated with an existing name. He can do so by influencing or controlling parties participating in the name system.

Some name systems were designed with a weaker adversary model in mind; in particular, the assumption that the adversary does not control the majority of the nodes or the majority of the computing power is a popular model in computer security in general. However, censorship resistance is typically an issue for activists, and thus hardly a topic for the majority of Internet users. As a result, it is unlikely that any censorship resistant name system is going to be used widely enough to compete with the computational power available to major governments. Thus, we advocate using the assumption that the adversary might have more computational power than all other participants combined.

2.2 Functional Requirements

The basic functionality of a name system for the Internet is to map memorable names to correct values. After all, name resolution provides names for systems such that human beings can easily remember them, instead of having to remember the more complicated (and possibly frequently changing) address values used by the network.

One of the most important Internet services is the Web, and a fundamental building block for Web services is the ability to link to information hosted on different systems; as humans often manually create these links, links are specified using names. Thus, a name system should be designed to support link resolution: a service provider must be able to link to a foreign resource, and the users of the service must then be able to resolve the name to an address for the intended destination.

3 Design Space for Name Systems

This section explores the theoretical design space for name systems; we will structure our discussion on how a name system can be realized using Zooko's triangle [9], an insightful conjecture on the design space for name systems (Figure 1).

Definition 1 (Memorable). *A name is memorable if it is feasible for an attacker in our adversary model to obtain it by enumerating names (bit strings). In other words, the number of bits of entropy in a memorable name is insufficient against enumeration attacks.*

Definition 2 (Secure). *A secure name system must enable benign participants to register and retrieve correct name-value mappings while experiencing active, malicious participants (which are assumed to follow the adversary model described in Section 2.1).*

Definition 3 (Global). *The system supports an unlimited number of participants without prior coordination or certification of participants. All benign participants receive the same (global) values for the same names.*

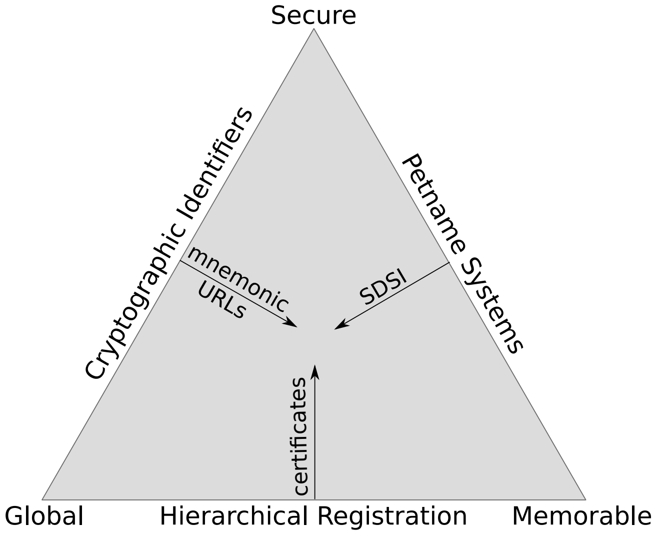

Fig. 1. Illustration of Zooko's triangle and key approaches to name systems.

Theorem 1 (Zooko's triangle). *It is impossible to have a name system that achieves* **memorable***,* **secure** *and* **global** *names at the same time.*

We confirmed with Zooko Wilcox-O'Hearn that these definitions represent the intended interpretation of his formulation. We show now that Zooko's triangle is a valid conjecture *in our adversary model*:

Proof. All participants, including the adversary, are supposed to be able to register names under the "secure" property of the name system. As names are memorable, an adversary can enumerate all possible names. Thus, the adversary can perform a squatting attack by (if necessary) assuming the identities of name system components that restrict registration and performing the necessary computations (we assumed he is able to do those faster than the rest of the network combined). The adversary can use this attack to register *all* memorable names. As names are global, once the adversary has registered a name, that name can no longer be registered by anyone else.

Thus, the squatting attack can prevent the registration of memorable names by normal participants. Thus, in our security model, it is impossible to create a secure, global name system where memorable names are guaranteed to be available for registration by normal users.

A trusted authority in control of name assignments would easily prevent such an attacker from being successful; however, the existence of such an authority is outside of our security model. We would like to point out that the proof given above is controversial in the research community. We have had comments from reviewers ranging from assertions that the theorem is a trivial (or at least well-known) fact that does not require proof, to those questioning its veracity. We believe that this is the first formalization of Zooko's hypothesis and that the theorem holds in our security model — and that it is false under weaker assumptions. Thus, any name system *in our security model* must deemphasize one of the properties from Definitions 1–3. Figure 1 describes the three major design approaches in this context. The edges of the triangle represent the three simple designs, and the arrows towards the middle represent the three main designs which move toward satisfying all three properties.

3.1 Hierarchical Registration

In Zooko's triangle, a name system using hierarchical registration is a name system providing global and memorable names; however, in the hierarchical structure names are owned by organizations. These organizations receive the power to manage a subspace of the namespace by delegation from an organization ranked higher in the tree, which enables censorship. The well-known DNS is a distributed database realizing access to a name system with such a hierarchical structure.

In the original DNS, globally unique memorable names are managed by a handful of organizations with no security guarantees [10]. DNSSEC improves security by providing authenticity and integrity protection using cryptographic certificates; however, DNSSEC still requires trusted authorities and is thus open to certain types of attacks in our adversary model, as governments can typically easily compel a limited number of easily identified service providers. In particular, given the delegation hierarchy, an adversary can put pressure on organization to obtain control over all subdomains. As a result, top-level domain providers are both extremely powerful and extremely high-value targets, as is the control over the root zone itself.

3.2 Cryptographic IDs and Mnemonics

A name system that can securely map globally unique identifiers to values can be achieved using cryptographic identifiers as names. In such a system security is achieved by certifying values using the private key associated with the cryptographic identifier. Names are with high probability globally unique. However, as cryptographic identifiers are long random bitstrings, they are not memorable. An example for a deployed name system with cryptographic identifiers is Tor's ".onion" namespace [11].

The proposed Tor mnemonic URL system [12] aims to make the ".onion" names more memorable by encoding the hashes names into "human-meaningful" sentences. However, the resulting names will not be memorable by our Definition 1 as the high entropy of the original cryptographic identifiers remains. As (assuming sufficiently strong cryptographic primitives are used) an adversary would not be able to enumerate all cryptographic identifiers, Tor's mnemonic URL system would not result in memorable names as those names correspond to cryptographic identifiers and thus could also not be enumerated. Finally, it is important to note that Tor's mnemonic URLs are still work in progress; it is thus difficult to assess the usability of this approach.

3.3 Petnames and SDSI

A secure name system with memorable names can be created using so called *petnames*. In a petname system, each user establishes names of his choice for other entities [13]. Each user or service would be identified using a cryptographic identifier based on a public key; the service provider can then sign mapping information to certify integrity and authenticity of the data. Memorable names are achieved by mapping petnames to cryptographic identifiers. While such a system can provide security and memorability, the mappings are only local and petnames are meaningless (or have a different meaning) for other users. A simple example of a petname system is the `/etc/hosts` file that allows administrators to create a mapping from hostnames to addresses for the local system.

Extending petname systems with ideas from Rivest's Simple Distributed Security Infrastructure (SDSI) [14] adds the possibility of (secure) *delegation*, allowing users to reference the petnames of other users. SDSI based delegation enables users to resolve other participant's names and thus enables linking to external resources. Delegation essentially adds another user's namespace in a subtree under a specific name. This creates an hierarchical namespace from the point of view of each user; globally, the resulting structure is simply a directed graph. While delegation broadens the accessibility of mappings, it does not achieve global names.

3.4 Timeline-based Name Systems

Timeline-based name systems, such as the Namecoin system [15], manage to combine global names, memorable names and security. In these systems, a global timeline with the domain registrations is secured by users performing proof-of-work computations, which in turn are used as "payment" for name registration.

Their existence does not contradict Zooko's triangle as their security depends on the adversary not having more computational power than the honest nodes; an adversary with sufficient computational power can create an alternative timeline with different domain registrations and a stronger proof-of-work, which would ultimately result in the system switching to the adversarial timeline. Thus, timeline-based systems do not fit the realistic adversary model we assumed for this paper (Section 2.1).

4 Practical Considerations

The previous section has outlined the design space for censorship resistant name systems. However, implementations of these alternatives will have to address a range of technical and practical concerns which be will discussed here.

4.1 Interoperability with DNS

To be accepted by users, a censorship resistant name system should respect user's usage patterns and integrate with existing technologies. Users should not have to manually switch between alternative name systems and DNS. Syntax and semantics of the different name systems should also be similar to not confuse the user about the meaning of names.

Thus a central requirement for any alternative name system will be interoperability with DNS. Users are used to DNS names and virtually all network applications today use DNS for name resolution. Thus, being interoperable with DNS will allow censorship-resistant alternatives to be used with a large body of legacy applications and facilitate adoption by end users.

Interoperability with DNS largely implies that alternative name systems should follow DNS restrictions on names, such as limiting names to 253 ASCII characters, limiting labels to 63 characters and using Internationalizing Domain Names in Applications (IDNA) [16] for internationalization. Furthermore, the name system should be prepared to return standard DNS records (such as "A" and "AAAA") to typical applications.

Interoperability with DNS should also include accessing the information of DNS from within the namespace of the censorship resistant name system. For example, it is conceivable that a censor might block access to `www.example.com` by removing the nameserver information for `example.com` in the `.com` TLD, without blocking access to the nameserver of `example.com`. In this case, a censorship resistant name system only needs to provide an alternative way to learn the nameserver for `example.com` — the lookup of `www` can then still be transmitted directly to the authoritative nameserver. In an alternative name system supporting delegation, this simply requires support for delegating subdomains back to DNS. This allows users to bypass censorship closer to the root of the DNS hierarchy even if the operators of the censored service do not explicitly support the censorship resistant name system.

Finally, for good interoperability users must not be required to exclusively use an alternative domain name system — alternating between accessing DNS for domain names that are not censored and using the censorship resistant name system should not require the user to reconfigure his system each time!

Interoperability and using multiple name systems with the same configuration can be easily achieved using pseudo-TLDs. A pseudo-TLD is a top level domain that is not actually participating in the official DNS. For example, using the pseudo-TLD ".key", a user might specify "ID.key" to access a name system based on cryptographic identifiers, or "NICK.pet" to access a pseudo-TLD ".pet" for petnames. Naturally, this only works as long as the names chosen for the pseudo-TLDs are not used by the global DNS.

Once pseudo-TLDs have been selected, the local DNS stub resolver can be configured (for example, using the Name Service Switch [17]) to apply special resolution logic for names in the pseudo-TLDs. The special logic can then use alternative means to obtain and validate mappings, which will work as long as the final results returned can be again expressed as a DNS response.

4.2 End-to-End Security and Errors

Today, client systems typically only include a DNS stub resolver, delegating the name resolution process to a DNS resolver operated by their Internet Service Provider (ISP). As ISPs might be involved in censorship, they cannot be trusted to perform proper name resolution. Thus, secure name systems (including DNSSEC) must be deployed end-to-end to achieve the desired security.

This may not only require updating operating system resolvers. Existing applications sometimes implement their own DNS clients, and typical DNS APIs (such as POSIX's name resolution functions) do not include error reporting that incorporates security attributes. Browsers will thus be unable to benefit from TLSA records [18] until they either implement full DNSSEC resolver functions, or until operating system APIs are enhanced to allow returning additional information. A particularly critical example is the possibility to return unsigned records even within a DNSSEC deployment. As a result, DNSSEC protections can easily be disabled by replacing signed valid records with a set of invalid records without signature information.

4.3 Petnames and Legacy Applications

In addition to integration with existing systems an alternative name system also has to consider assumptions made by applications in higher layers, for example existing applications assuming globally unique names. Existing support for virtual hosting of websites in HTTP-based applications and TLS/SSL certificate validation both assume that the names given by the client match exactly the (DNS) name of the respective server. Links to external websites are typically specified using (globally unique) DNS names; as a result, relative names involving delegation from a SDSI-based name system would not be properly understood by today's browsers.

In lieu of directly modifying legacy applications, it might be possible to perform the necessary adaptations using proxies. Proxies might be used to translate hostnames from websites using delegation, and to perform SSL certificate validation (for example, by looking at TLSA [18] records from the secure name system instead of hostnames). Reverse proxies could be used to generate the virtual host names expected by the server, and to translate links with absolute links to those using the delegation chains provided by a SDSI-based name system. Additional records in the name system might be used to aid the conversion between relative names and legacy names by the proxies. In order to achieve end-to-end security, these proxies would naturally have to be operated within the trusted zone of the respective endpoints in the system.

4.4 Censorship-Resistant Lookup

Censorship resistant distributed name systems need to consult name information from other participants and thus require a network protocol to perform censorship resistant lookups. The most common method for implementing key-based searches in decentralized overlay networks is the use of a *distributed hash table* (DHT).

Typical attacks on DHTs include poisoning and eclipse attacks. In a poisoning attack, the adversary attempts to make interesting mappings hard to find by placing many invalid mappings into the DHT. A censorship-resistant DHT for a name system that uses public keys to lookup values signed by the respective private key can easily defeat this type of attack by checking signatures. In an eclipse attack, the adversary tries to isolate particular key-value mappings from the rest of the network. Modern DHTs defend against this type of attack by replicating values at multiple locations [19].

Some censorship resistant DHTs such as X-Vine [20] and R^5N [21] additionally accept limited connectivity between the peers in the DHT, making it harder for the adversary to disrupt DHT operations in the IP layer. Furthermore, this also allows peers to restrict connections to known friends, making the DHTs more robust against Sybil attacks [22] by building the overlay topology using existing social relationships.

One important property in this context will be query privacy. In existing centralized name systems, infrastructure providers can easily observe which names are used by which users. When the database is decentralized in a DHT, these central observation points are eliminated; however, now ordinary users can observe other user's queries, which maybe even more problematic for some applications. Thus, it is desirable to have encryption for queries and responses in the DHT. The encryption could be based on secrets only known to the user performing the resolution (such as the label and the zone); as a result, other users could only decrypt the resolution traffic with a confirmation attack where they would have to guess the label and zone of a query (or response). This would strengthen censorship-resistance as participants would typically not know which requests they are routing. Additional query privacy might be achieved by anonymizing the source of the request, for example by using onion routing [11]. Naturally, using such anonymization techniques increases latency.

4.5 Case study: Usability

Unlike DNS, the user's experience when using a name system based on SDSI depends on high-level user behavior: following a link corresponds to traversing the delegation graph and resolution is fully automatic. However, when users want to visit a fresh domain that is not discovered via a link, SDSI requires a trust anchor to be supplied via a registrar or out-of-band mechanisms, such as QR codes. This raises the question: how often are these inconvenient methods needed in practice?

To answer this question, we did a survey on surfing behavior. Specifically, we wanted to find out how often users would typically type in a new domain name for a site. A domain name is "new" if the user has never visited it before, and if the user is typing it in the name is also not easily available via some link. Typed in new domain names are thus the case where a SDSI-based name system (or PKI) would need to use some external mechanism to obtain the public key of the zone.

Based on a limited and most likely biased survey where users volunteered the output of a simple shell script that inspected their browsers history database, we determined that given current Internet behavior, approximately 8% of domain names would require introduction via some out-of-band exchange. A key limitation of the survey's methodology was that we did not attempt to control who submitted results; we simply used the data of anyone who was willing and able to download and run the shell script that performed the analysis. This limited the sample to somewhat more technologically sophisticated users. The complete results from our survey and details on the methodology can be found in [23]. Our conclusion is that a name system based on petnames and SDSI-style delegation stands a chance of being an acceptable choice if communication is hindered by censorship or strong security assurances (beyond those offered by the X.509 PKI or DNSSEC) are required.

5 Censorship in Other Layers

Censorship does not stop with the name system. For example, censors can also attempt to block information by destination IP address. Blocking IP addresses is actually easier than censoring DNS; however, there is an increased chance of collateral damage as with virtual hosting, a single IP address can host many sites and services. Tools that help users circumvent IP-level censorship can also benefit from censorship resistant name systems.

For example, the Tor network [11] is an anonymizing public virtual network for tunneling TCP connections over the P2P overlay network. While Tor is often associated with the goal of providing anonymity for HTTP clients, it can also be used to circumvent censorship by tunneling (the Tor overlay) traffic in other protocols, such as TLS. Tor also offers the possibility of hosting services within the Tor network, here with the primary goal of providing anonymity to the operators of the servers. Accessing these "hidden services" using cryptographic identifiers is not particularly user-friendly.

Given a decentralized censorship resistant name system, it should be easy to provide names for services offered within such P2P overlays. The name system would map names to a new record type that identifies the respective service and peer (instead of using “A” or “AAAA” records to reference a host on the Internet). Such service endpoint addresses can then again be translated to IP addresses in the entry node’s private address range to enable communication of legacy applications with the P2P service. The result would be still close to hidden services in Tor, though it would not necessarily have to also provide support for anonymity.

6 Conclusion

We have outlined the limitations of censorship resistant name systems and shown that it is not possible to achieve memorable, secure and global names in a unified name system. However, it is possible to use pseudo-TLDs to allow users to cherry-pick between multiple name systems, offering combinations of two of the three desirable properties. Among the theoretical ideas, the SDSI-design using delegation is the only which has so far not been attempted in practice. Here, the lack of globally unique names creates additional issues for legacy applications that need to be mitigated. Focusing on Web applications, we have performed a survey which shows that a delegation-based name system would offer significant benefits over simpler petname systems, as most name resolutions in practice arise from users following links. As each design offers unique advantages, developers of censorship circumvention tools should consider the integration or interoperability of their systems with *multiple* secure name systems via pseudo-TLDs, including DNS/DNSSEC, cryptographic identifiers and petnames with delegation.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) under ENP GR 3688/1-1. We thank everyone who submitted information about their browser history for our study of surfing behavior. We thank Jacob Appelbaum, Daniel Bernstein, Ludovic Courtès, Ralph Holz, Luke Leighton, Simon Josefsson, Nikos Mavrogiannopoulos, Ondrej Mikle, Stefan Monnier, Niels Möller, Chris Palmer, Martin Pool, Richard Stallman, Neal Walfield and Zooko Wilcox-O’Hearn and the anonymous reviewers for FPS’2013 for insightful comments and discussions on an earlier draft of the paper. We thank Krista Grothoff for editing the paper.

References

1. Greenwald, G., MacAskill, E.: NSA Prism program taps in to user data of Apple, Google and others. The Guardian (June 2013)
2. Times, R.: Post PRISM: Encrypted communications boom after NSA leaks. `http://www.youtube.com/watch?v=JJY3EXVdyiM` (June 2013)
3. Schmid, G.: Report on the existence of a global system for the interception of private and commercial communications (ECHELON interception system). European Parliament Session Document 2001/2098(INI) (July 2001)
4. `http://politi.dk/`: Fejl blokerede internetsider kortvarigt. `http://goo.gl/beQFm` (March 2012)
5. Anonymous: The collateral damage of internet censorship by dns injection. ACM SIGCOMM Comp. Comm. Review **42**(3) (July 2012) 22–27
6. European Parliament: Resolution on the EU-US Summit of 28 November 2011 (November 2011) P7-RC-2011-0577.
7. `http://opennet.net/`: The OpenNet Initiative (January 2013)
8. Stallman, R.: Why software should not have owners. `http://www.gnu.org/philosophy/why-free.html` (2012)
9. Wilcox-O'Hearn, Z.: Names: Decentralized, secure, human-meaningful: Choose two. `http://zooko.com/distnames.html` (Jan 2006)
10. Mockapetris, P.: Rfc 1035: Domain names - implementation and specification. Technical report, Network Working Group (November 1987)
11. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proc. 13th USENIX Security Symposium. (August 2004)
12. Sai, A.F.: Mnemonic .onion urls. `http://goo.gl/aOpKo` (February 2012)
13. Stiegler, M.: An introduction to petname systems. `http://www.skyhunter.com/marcs/petnames/IntroPetNames.html` (February 2005)
14. Rivest, R.L., Lampson, B.: SDSI – a simple distributed security infrastructure. `http://groups.csail.mit.edu/cis/sdsi.html` (1996)
15. `http://dot-bit.org/`: The Dot-BIT project, A decentralized, open DNS system based on the bitcoin technology. `http://dot-bit.org/` (April 2013)
16. Faltstrom, P., Hoffman, P., Costello, A.: RFC 3490: Internationalizing Domain Names in Applications (IDNA). Technical report, Network Working Group (March 2003)
17. Foundation, F.S.: The GNU C Library - System Databases and Name Service Switch. `http://goo.gl/gQYOw`
18. Hoffman, P., Schlyter, J.: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. IETF RFC 6698 (Aug. 2012)
19. Polot, B.: Adapting blackhat approaches to increase the resilience of whitehat application scenarios. Master's thesis, Technische Universität München (2010)
20. Mittal, P., Caesar, M., Borisov, N.: X-vine: Secure and pseudonymous routing using social networks. CoRR **abs/1109.0971** (2011)
21. Evans, N., Grothoff, C.: R^5N: Randomized Recursive Routing for Restricted-Route Networks. In: 5th Int. Conf. on Network and System Security. (2011) 316–321
22. Douceur, J.R.: The Sybil Attack. In Druschel, P., Kaashoek, M.F., Rowstron, A.I.T., eds.: IPTPS. Volume 2429 of Lecture Notes in Computer Science., Springer (2002) 251–260
23. Schanzenbach, M.: A Censorship Resistant and Fully Decentralized Replacement for DNS. Master's thesis, Technische Universität München (2012)