DNSChain + okTurtles

Easy to use, state of the art security, for existing online communications

Author: Greg Slepak

1 Abstract

In this paper, we'll briefly describe some of the shortcomings of today's Internet security (SSL/TLS, HTTPS, and Certificate Authorities), and why these systems do not provide the security that they claim to provide. We'll focus on the importance that authentication plays in securing communication systems, why a system that fails to provide strong authentication does not provide *meaningful security*, and how these problems can be addressed. We then introduce two projects that address these issues. The first project, DNSChain, deprecates today's insecure and fraudulent[1] public key infrastructure (PKI) by gracefully transitioning DNS from its hierarchical design, to one that is based on a globally distributed, peer-to-peer network that successfully "squares Zooko's triangle".[1] We then use the strong authentication provided by DNSChain to introduce okTurtles: a protocol and browser extension that protects the content of almost all online text-based communication from a variety of threats (such as MITM).

KEYWORDS: Namecoin, DNS, SSL/TLS, security, authentication, X.509, HTTPS, P2P, MITM, DNSChain, DNSNMC, okTurtles, OTR, Certificate Authorities, PKI, Zooko's Triangle, plausible deniability

2 Motivation

Thanks to the bravery of whistleblowers, it may very well be common knowledge that many governments around the world—with the United States in particular—record virtually all electronic communication without obtaining a warrant, or permission from the individuals and groups they are recording. In the United States, this surveillance was found to violate the nation's highest law (the Constitution).[2] For reasons unknown to this author, this practice continues to this day, unabated.

---

[1] Companies that sell SSL certificates usually claim that their certificates provide customers with "security." Customers are led to believe that these certificates protect browser-server communication from eavesdropping and tampering. As elaborated in this paper, this simply isn't true today.

Today's surveillance is made possible because most protocols that facilitate online communication do not provide all of the following properties:

1. End-to-end encryption
2. Secure authentication
3. Perfect Forward Secrecy (PFS)

We use the term "meaningful security" to refer to the security provided by protocols that employ all of these features for communication between individuals. The security that such protocols provide is *meaningful* because it protects users from most of the known life-threatening and *life-damaging* dangers they may encounter online. Any entity that sits between (or next to) the user and the endpoint they're communicating with represents a potential threat. The list of threatening actors includes institutions of all sorts (businesses, universities, etc.), governments, internet service providers, malevolent network administrators, and random hackers.

Governments pose the most significant threat to users because of their fantastic surveillance capabilities, virtually unlimited resources, and their ability to operate above the law or modify it to suit their needs. As an illustrative example, it was revealed that the NSA appears to be storing all of the information that they are technically capable of getting their hands on.[3][4][5][6] The NSA gives encrypted traffic special attention, storing it for a longer period of time.[7] The existence of such pervasive data retention means that it is no longer sufficient for online communications protocols to simply encrypt data. Communication protocols that do no provide *Perfect Forward Secrecy* (PFS) can no longer be said to provide any meaningful security to their users. PFS employs the use of ephemeral keys to encrypt data that is transmitted between endpoints. This renders the stored encrypted communication worthless in the face of a compromised long-term private key.[8]

A fourth property, called *plausible deniability,* is sometimes desirable when the communication is between individuals:

4. Plausible deniability (sometimes)

Plausible deniability, in the context of computer security, represents feature(s) that give users a "legitimate out", or a way to deny responsibility or ownership of incriminating data.[9] Governments, and other entities, will often employ the threat of physical harm or imprisonment to manipulate and threaten the livelihoods of non-violent individuals. This often happens in the context where they need the secret key to decrypt some encrypted data that they believe can be leveraged against said individuals. Unfortunately, this is becoming an increasingly common scenario today, threatening the lives of journalists, human rights activists, individuals accused of petty drug crimes, and even their relatives or partners.[27][28][29][30] If laws and courts were rational and just, and governments responsible and ethical, plausible deniability would not be a necessary, or even a desirable feature.

In today's present reality, however, this simply is not the case. For example, in the United States, which has the world's highest incarceration rate, **victimless so-called "crimes" constitute 86% of the federal prison population.**[31] This human catastrophe must be taken into account by engineers when designing communication protocols, and by executives and policymakers in order to fulfill their ethical and moral commitments to their respective professions, customers, stakeholders, and to fellow citizens. Communication protocols, therefore, should attempt to address the threat of coercion. Plausible deniability feature(s) can protect users from such threats, and therefore protocols that facilitate communication between individuals should provide plausible deniability features when possible. As we'll discuss later, this is not always possible, however, and when it is, it is often difficult to achieve the desired *degree* of deniability.

The counterargument often raised against protocols that provide meaningful security is that they might enable the activities of truly violent criminals that we actually do want punished. This argument is completely legitimate and must be considered. The quantitative data necessary to perform a "Calculus of Suffering" portrays a rather stark picture. On one side are government officials and agencies that have built for themselves a reputation for providing false information [32], violating laws [2][33], and even their own rules[2] [34], justify their need to invade the individual's privacy "Because Terrorism." On the other side are the victims (either dead or rotting in prison) of the policies that allow such agencies to continue to exist and do the things they do. In terms of numbers, the "Threat of Terrorism" was found to cause approximately the same

[2] The author has reached such a point of exhaustion with one specific agency that he is no longer paying attention to, or collecting news articles about, the novel methods this agency discovers for compromising its own integrity.

number of casualties as furniture (per year) [35], and roughly *eight times less* than the "Threat of Police Officers" [36]. A final nail in the coffin of the "Because Terrorism" argument is the fact that governments are often responsible for the very terrorism that is their excuse to impinge on people's rights (privacy and otherwise). Though usually indirectly responsible, occasionally they'll stage a terrorist act in so-called "false flag" operations (sometimes killing their own citizens). They then publicly blame someone else for the incident, and use the ensuing furor for their own selfish goals (for example, to gain public approval for a war against another country).[41][42]

These facts, especially when weighed against the incredible human toll caused by misguided policies and legislation [31][37], indicate that some balancing force is urgently needed to restore a semblance of sanity to the world we find ourselves in.

3 Certificate Authorities make HTTPS and SSL/TLS insecure

Today, HTTPS is the primary means by which the connection between a user and a website is secured. HTTPS does not provide meaningful security (as has been defined in this paper). HTTPS relies on several underlying technologies for its meaningless security: SSL / TLS and X.509.

According to T. Dierks et. al, "Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols which are designed to provide communication security over the Internet."[10][11] The latest version of TLS provides encryption and PFS, but it does not provide secure authentication or plausible deniability, and therefore does not provide meaningful security. For authentication, TLS relies on *certificates* to authenticate end-points as described by X.509 PKI.[10][12] Certificates are electronic documents that can contain various information (a public key and information about some entity), and employ cryptographic signatures that prove their authenticity.

When a user connects to a website over HTTPS, they are given the website's certificate. The user's web browser has no way to tell whether the certificate is legitimate. It is possible that the certificate was replaced with a fraudulent one enroute to the browser via a Man-in-the-Middle (MITM) attack.[13]

To verify the legitimacy of the certificate, web browsers rely on “root certificates” that belong to *Certificate Authorities* (CAs). These certificates are shipped with the browser itself, and therefore can (supposedly) be trusted.[3] These “root certs” are used to verify the certificates provided by HTTPS servers. Server operators must purchase a certificate that is signed by one of these CAs. If *any one* of the 600+ root certs that browsers trust[4] declares the website’s certificate legitimate, the web browser will actually declare to the user (via graphical elements such as those shown in Figure 1) that the connection to the website is secure, and that no MITM is taking place.

Unfortunately, this method of authentication is broken.[14][15] Because browsers will trust any one of hundreds of CAs, it is only necessary to compromise a single one (whether it be via hacking, a court order, or physical intimidation) to compromise the security of HTTPS worldwide, for all websites.

Tthere are many documented cases where fraudulent certificates have been used to MITM internet traffic, often to large companies like Google.[14][20][21]

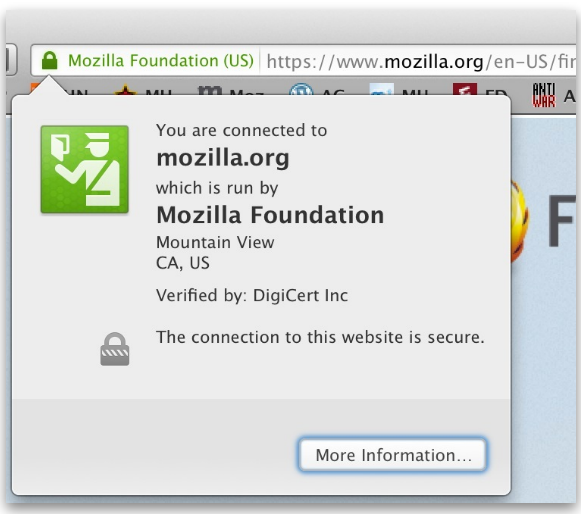

Figure 1. Firefox believes that “The connection to this website is secure.”[5]
Unfortunately, so do most users.

[3] Or so the theory goes. Few consider the possibility that the browser itself was compromised enroute to the user. Even if the browser remained untouched, glaring security holes still exist because of CAs.

[4] Phillip Hallam-Baker disputes the EFF’s figure on [therightkey] mailing list, but did not provided citable references for his claims.

[5] The sad part is, Mozilla (and others) pay hundreds of dollars for their certificates. Every year! :-O

4 Existing attempts to fix this problem

This document would not be complete if it failed to mention some of the existing proposals to address these problems. The table below mentions a few, along with their respective shortcomings:

Proposal Summary	Problem(s) with the proposal
Google's "Certificate Transparency"[22] wants CAs to note all of the certs that they issue into a log somewhere protected in someway and verified via some mechanism. The original proposal is almost this vague. Website owners are then asked to monitor these logs to see if their clients were hacked, in addition to continuing to pay the CAs money for the worthless certificates they provide.	The proposal summary seems to do an adequate job of summarizing the problems with it. This is not surprising. Companies that require access to your information for their survival will never provide meaningful leadership on security.
DNSSEC[23] suggests a complicated mechanism to essentially re-create many of the same problems with X509 and CAs in DNS itself, by providing a chain of trust to untrustworthy entities. Its intended goal is to secure DNS and thereby assure clients that when they ask for "**apple.com**", they are actually visiting **apple.com**.	This is another proposal that seems like a great argument against itself. It does not appear to protect against MITM.[25] Unnecessary complexity breeds security problems. DNSSEC appears to be a terrible idea for various other reasons.[26]
In their words, **Convergence** "is a secure replacement for the Certificate Authority System. Rather than employing a traditionally hard-coded list of immutable CAs, Convergence allows you to configure a dynamic set of Notaries which use network perspective to validate your communication."[24] In our words: Convergence is similar to having a known_hosts ssh key file for your browser, and comparing it against your friend's file. Not a terrible idea.	The website claims it's simple to use, but we have to disagree because users are asked to manage a list of notaries. It depends on group consensus, but the group might not be very bright. What happens then? It does not provide MITM protection on first-visit. All of the notary info appears to be stored locally to the computer, or even the browser. Rather inconvenient for just about everyone.

5 Introducing DNSChain, your connection Namecoin's blockchain

DNSChain fixes the authentication problems previously described, and it addresses all of the problems that with the previously mentioned proposals. It does this first by combining DNS with Namecoin (NMC), and then by encouraging a "trust only those you know" policy.[6]

"Namecoin is an open source decentralized key/value registration and transfer system based on Bitcoin technology".[16] Namecoin "squares Zooko's Triangle", meaning, it makes it possible to have domain names (and other types of identifiers) that are:

- **Authenticated:** users can be certain that they are not speaking to an impostor
- **Decentralized:** there is no central authority controlling all the names
- **Human-readable:** names look just like today's domain names

However, by itself, Namecoin does not provide the means by which ordinary users can take advantage of the features it provides. Using Namecoin is far too cumbersome for the vast majority of internet users, even those with years of computer expertise. For one, it cannot be used on mobile devices (like iPhones) in its current state because of its network requirements.

DNSChain provides the missing "glue" to the Namecoin blockchain that makes it immediately accessible to clients of all types with *zero configuration*. A network administrator need only enter the IP address of a DNSChain-compliant DNS server to instantly make the information within the blockchain accessible to all of the users that she (or he) provides internet access to.

To be assured of the authenticity of answers provided by a DNSChain server, clients must have its public key fingerprint. With these two pieces (the server's IP address and the server's fingerprint), users are given strong authentication for all of the information that resides within the blockchain. Of course, we do not claim that this system provides *perfect authentication*[7], but rather it provides authentication that is *meaningful.* Once this relationship has been established between the DNSChain server and its clients, the

[6] In contrast to the "trust a bunch of untrustworthy strangers" policy that today's browsers implement.

[7] Such a system is most likely impossible. Even if the person is standing right in front of you, they might not be themselves. Watch *The Exorcist* if you don't believe me. ;-)

clients are *guaranteed* to receive accurate values from the blockchain, *so long as the software involved (both server & client) and their respective keys (public and private) are not compromised.*

Ensuring the integrity of the software and keys involved is an orthogonal problem that affects all authentication systems, and thus is outside the scope of this paper.

The magic doesn't stop there. DNSChain isn't just DNS + NMC, it's also an HTTP server. DNSChain provides its clients with secure access to the Namecoin blockchain itself through a RESTful API. 😊

It accomplishes this by exposing a *metaTLD*: **.dns**

When a request is made to **namecoin.dns**, the DNSChain server responds with its own public-facing IP address. This is what makes okTurtles possible.

Clients wanting to know Joe Biden's public key fingerprint need only load to the following URL:

http(s)://namecoin.dns/id/jbiden

The DNSChain server will then respond with a cryptographically signed JSON message containing all of the relevant info. Clients can optionally avoid retrieving a cached IP address by prepending a random subdomain:

http(s)://ajpow8jwojfsdjkl.namecoin.dns/id/jbiden

The precise details of the JSON format and the RESTful API are to be decided. From these preliminaries, however, it is clear that this design not only allows JavaScript clients to securely read data from *any* blockchain (without the burden of running a P2P node locally), but also the potential to securely write data back to the blockchain.

6 okTurtles + DNSChain

okTurtles takes the authentication provided by DNSChain, and uses it to provide secure communication through virtually any website. It will initially be implemented as a web browser extension. okTurtles uses DNSChain to authenticate individuals (instead of websites). This makes it possible to communicate with individuals around the world through almost any website, and without any action or intervention on the part of the site operator. Specific use cases for okTurtles include establishing secure communication with individuals on social networks like Facebook, and the sending of secure email via web email clients like Google's Gmail.

okTurtles employs DNSChain for secure authentication, and uses asynchronous OTR to provide end-to-end encryption and PFS. Thus, okTurtles fulfills the criteria outlined in this document for meaningful security. We shall explore the sort of plausible deniability offered by okTurtles in more detail in the next section, but for now sufficed to say that using okTurtles over a password-protected account on a third-party website greatly reduces the degree of plausible deniability that is available to the user.

When a user installs the okTurtles browser extension, they will be asked to claim an identity and to choose a password (known only to them) that will be the foundation upon which their identity is verified. A public key is generated on their behalf, and the fingerprint to this key is sent to the Namecoin blockchain (along with any supplied information about their identity, such as their full name, their online handles, etc.).

Whenever an okTurtles user clicks on an HTML `<textarea>` field, okTurtles will scan the entire page to try and use site-specific plugins to figure out who they are trying to talk to. If it is able to find some form of identity (for example, on Facebook this would be their friend's name), it will then check the blockchain to see if it can find a matching identity, and if not, the user can manually enter their friend's unique Namecoin id (id), or invite their friend to install okTurtles and create a global identity.

Similarly to how web browsers come with root certs, okTurtles will have a list of DNSChain server IP addresses and their corresponding public key fingerprints. To avoid falling into the same trap that web browsers enjoy today with Certificate Authorities, okTurtles will encourage the user to use their own DNSChain server, whether it's one that they have setup themselves, or one that belongs to someone they trust. Defaults are provided to encourage adoption and to make the software function immediately upon install.

6.1 Plausible deniability, anonymity, and how they relate to okTurtles

okTurtles does not offer any intrinsic anonymity features by itself. Although it is certainly possible to use okTurtles anonymously (over Tor, etc.), providing anonymity has never been a design goal of okTurtles[8].

Since okTurtles is designed to send messages over existing websites and platforms, the level of plausible deniability (PD) it can offer is somewhat limited. This is true, however, for all such software[9].

One of the most popular communications protocols to offer plausible deniability features is OTR (Off-the-Record). OTRv3 has a few problems, however. For one, it does not offer automatic secure authentication on first encounter[39]. Also, the PD it offers users is limited to those they've had conversations with. If Alice uses OTRv3 to converse with Bob, then if Bob (later on) produces quotes allegedly made by Alice (from stored logs), Alice has the ability to claim that Bob fabricated them[38]. If, however, Bob's intentions were malicious from the start, PD disappears. To prove the authenticity his recorded messages, Bob need only save all the ephemeral encryption keys, and then submit them to the police along with the ciphertext that was recorded by the third-party chat service he and Alice were using.[40]

A recently proposed modification to the OTR protocol by Trevor Perrin eliminates DSA signatures[38], thus slightly improving OTR's PD by making it possible for *anyone* (third-parties included) to manufacture "quotes from Alice." However, in the contexts that we've been exploring, these improvements do not significantly alter the degree of PD that OTR offers. The set of individuals capable of damaging Alice (or Bob) remains unchanged: those individuals they've actually spoken to over a third-party's network. The previously described "malicious Bob" would still be able to prove the authenticity

---

[8] The author sees good reason to make *encryption* a common sense practice (like putting on your seat belt before driving). He does not hold this viewpoint for anonymity.

[9] Pidgin (Linux), Adium (OS X) and TextSecure (Android) are good examples of this.

of his conversation with Alice if he saved the ephemeral keys. While it is certainly open to debate, this author thinks that a judge who is presented with ciphertext from Facebook's (or Verizon's, or the NSA's) servers, along with all the keys to decrypt it, will be hard pressed to believe that Bob *and* Facebook conspired against Alice.[10]

Finally, it is important to emphasize (in case it's not clear) that the majority of these conversations occur in a non-anonymous fashion over the networks and chat services of many third-parties. Therefore, the simple fact that a conversation occurred, and that it occurred between these two (or more) specific individuals, remains undeniable.

6.2 Plausible deniability when participants are trustworthy

A final consideration of PD should examine the situation where Alice and Bob aren't out to get each other. In this case, there isn't much PD to consider that we haven't already considered (for the ciphertext saved by a third-party).

There is, however, a **very** serious PD issue, and that is what to do with the plaintext that's decrypted on either side of the conversation? In traditional OTR situations, the plaintext is simply discarded, forever lost. For most users, this "feature" is undesirable. Even in the context of ordinary chats, users often save their conversations, either locally or online. This is certainly true for email.

Though it is possible to use okTurtles in the traditional OTR way (forgetting decrypted messages after some time), most users will probably want their conversations archived for future reference. To maintain the security of these conversations, okTurtles re-encrypts them with the user's password and strips the messages of their digital signatures. To preserve PD in the event that a user is forced to give up their password, okTurtles employs a few tricks from Espionage[11], and gives users the ability to have multiple passwords. Each password unlocks a different set of encrypted data. Thus, a "potential out" remains, and users are able to hold on to their memories. The OTR ciphertext that was recorded by intermediary servers is now worthless to everyone, including the chat participants. Of course, many questions about how to properly store encrypted messages on third-party servers still need to be worked out.

---

[10] Might make for a fantastic Hollywood film! If you decide to make it, don't forget to credit this paper!

[11] Self-promotion warning! Author's company sells this product! http://www.espionageapp.com/

6.3 Note on JavaScript Cryptography

okTurtles makes extensive use of JavaScript to enhance the functionality of HTML `<textarea>` tags, and to transparently encrypt and decrypt messages. Ideally, we would like to do as much of this in JavaScript as possible, because JavaScript is easier to maintain and actually has a few security advantages over low-level languages[12].

We are aware that some concerns have been raised about JavaScript cryptography.[17] Most of these concerns have to do with the integrity of the code during two phases: (1) distribution (updates to the extension), and (2) when JavaScript is injected into a potentially malicious webpage.

Verifying code updates to the extension is easily accomplished thanks to DNSChain, so that is not a concern. Verifying the integrity of JavaScript that has been injected into an arbitrary webpage is a more challenging problem and must be adequately addressed. Our plan is simple and consists of two primary approaches:

1. Investigate the capabilities of DefensiveJS [18] and JavaScript sandboxing [19].
2. For any crypto that cannot be safeguarded on the page, we will implement it in the browser extension in a manner that shields it from the page.

We'd like to remind the reader that this document is not a complete specification of okTurtles and DNSChain, but rather an incomplete design overview. These projects remain very much works in progress.

7 Conclusion

In this paper we covered a lot of ground. We described the severe issues plaguing today's online security and how they can be traced to the lack of a secure authentication mechanism. We explained why this is an urgent matter, why it must be addressed, and

[12] No NULL pointer exceptions and other memory-related bugs, for example.

then introduced three criteria that a communication protocol needs in order to have what we refer to as "meaningful security" in today's world.

We discussed a few of the existing proposals to fix today's "authentication crisis", and enumerated various reasons that (hopefully) explained why these proposals do not address the problem in any meaningful way.

Finally, we introduced two free and open source projects: DNSChain and okTurtles, and gave an overview of what they are and why we think they are good solutions.

For more information, please visit the project website, sign up for the project newsletter, and if you'd like, follow @okTurtles on twitter. The .bit link below is for those already using Namecoin for DNS.

okturtles.com or okturtles.bit

We look forward to the day that all domains (not just .bit), are looked up via the blockchain. That will be the day that you actually own your domains[13], pay almost nothing for them, and your connection will be MITM-free! 😊

8 Acknowledgements

The author would like to thank the following individuals for their comments and for proof-reading this document:

- Nicolas Seriot (@nst021) for proof-reading the document. Nicolas is a developer with experience analyzing the security/privacy implications of various platforms

[13] Unlike traditional domains, domains (and other identifiers) in the blockchain cannot be "seized".

(notably iOS and Twitter's API). Per his twitter bio: *"Cocoa developer, happy father, a free thinker"*. His work can be found at his website: http://seriot.ch/

- Frederic Jacobs (@FredericJacobs) for proof-reading this document. Per his twitter bio: *"Hacker at @WhisperSystems, @cryptocatapp & @HackEPFL. Networks Research in Syria and Iran. Ethics & Crypto enthusiast. Building and breaking iOS & Android."*
- Bob Jesse for his insightful comments and advice. Bob does not claim to be a software security professional.
- Trevor Cole (@djphileep) for proof-reading the document and catching typos. Trevor does not claim to be a security professional (to my knowledge).
- Readers of [therightkey] mailing list and reddit.com/r/netsec for their comments.

Document History & References follow next

9 Document History

- 1.1.1 - April 26, 2014 - Updated all instances of 'DNSNMC' to 'DNSChain'. Updated a few sentences near the end of Section 5.
- 1.1 - December 17th, 2013 - Updated Acknowledgments section and corrected typos (thanks Trevor Cole!). Added footnote on the EFF's 600+ figure. Added Document History section.
- 1.0 - December 12th, 2013 - IPO.

10 References

[1] A. Swartz, "Squaring the Triangle: Secure, Decentralized, Human-Readable Names", <http://www.aaronsw.com/weblog/squarezooko> (6 December 2013).

[2] M. Rumold, "EFF Victory Results in Release of Secret Court Opinion Finding NSA Surveillance Unconstitutional", <https://www.eff.org/deeplinks/2013/08/eff-victory-results-expected-release-secret-court-opinion-finding-nsa-surveillance> (6 December 2013).

[3] J. Bamford, "The NSA Is Building the Country's Biggest Spy Center (Watch What You Say)", <http://www.wired.com/threatlevel/2012/03/ff_nsadatacenter/all/> (6 December 2013).

[4] J. Stray, "FAQ: What You Need to Know About the NSA's Surveillance Programs", ProPublica, <http://www.propublica.org/article/nsa-data-collection-faq> (6 December 2013).

[5] B. Schneier, "Evidence that the NSA Is Storing Voice Content, Not Just Metadata", Schneier on Security, <https://www.schneier.com/blog/archives/2013/06/evidence_that_t.html> (6 December 2013).

[6] J. Howerton, "NSA Can Spy on 75 Percent of ALL U.S. Internet Traffic, Says Stunning New Report", TheBlaze.com, <http://www.theblaze.com/stories/2013/08/20/report-nsas-broad-reach-covers-75-percent-of-all-u-s-internet-traffic/> (6 December 2013).

[7] M. Kassner, "Does using encryption make you a bigger target for the NSA?", TechRepublic, <http://www.techrepublic.com/blog/it-security/does-using-encryption-make-you-a-bigger-target-for-the-nsa/> (6 December 2013).

[8] "Forward secrecy", Wikipedia, <https://en.wikipedia.org/wiki/Perfect_forward_secrecy> (6 December 2013).

[9] "Plausible deniability", Wikipedia <https://en.wikipedia.org/wiki/Plausible_deniability#Use_in_cryptography> (6 December 2013).

[10] "Transport Layer Security", Wikipedia, <https://en.wikipedia.org/wiki/Transport_Layer_Security> (6 December 2013).

[11] T. Dierks, "The transport layer security (TLS) protocol version 1.2" (2008).

[12] "X.509", Wikipedia, <https://en.wikipedia.org/wiki/X.509> (6 December 2013).

[13] "Man-in-the-middle attack", Wikipedia, <https://en.wikipedia.org/wiki/Man-in-the-middle_attack> (6 December 2013).

[14] P. Eckersley, "How secure is HTTPS today? How often is it attacked?", Electronic Frontier Foundation, <https://www.eff.org/deeplinks/2011/10/how-secure-https-today> (6 December 2013).

[15] "Spooks break most Internet crypto, but how?", Ars Technica, <http://arstechnica.com/security/2013/09/spooks-break-most-internet-crypto-but-how/> (6 December 2013).

[16] "Namecoin", Namecoin, <http://namecoin.info/> (6 December 2013).

[17] "Javascript Cryptography Considered Harmful", Matasano Web Security Assessments for Enterprises, <http://www.matasano.com/articles/javascript-cryptography/> (6 December 2013).

[18] "Defensive JavaScript home", Defensive JavaScript home, <http://www.defensivejs.com/> (6 December 2013).

[19] "Welcome to Oasis", Welcome to Oasis, <http://oasisjs.com/> (6 December 2013).

[20] "Iranian Man-in-the-Middle Attack Against Google Demonstrates Dangerous Weakness of Certificate Authorities", Electronic Frontier Foundation, <https://www.eff.org/deeplinks/2011/08/iranian-man-middle-attack-against-google> (6 December 2013).

[21] "New NSA Leak Shows MITM Attacks Against Major Internet Services", Schneier on Security, <https://www.schneier.com/blog/archives/2013/09/new_nsa_leak_sh.html> (6 December 2013).

[22] Laurie, "Certificate Authority Transparency and Auditability", <http://www.certificate-transparency.org/original-proposal> (6 December 2013).

[23] "DNSSEC – What Is It and Why Is It Important?", ICANN, <http://www.icann.org/en/about/learning/factsheets/dnssec-qaa-09oct08-en.htm> (6 December 2013).

[24] M. Marlinspike, "Convergence | Beta", <http://convergence.io/details.html> (7 December 2013).

[25] Various, "[Cryptography] DNSSEC = completely unnecessary?", <http://comments.gmane.org/gmane.comp.encryption.general/16353> (7 December 2013).

[26] "DNSSEC Downtime: Outages & Validation Failures", <http://ianix.com/pub/dnssec-outages.html> (7 December 2013).

[27] "How Egyptian police quickly cracked journalist's computer password", Hacker 10 – Security Hacker, <http://www.hacker10.com/other-computing/how-egyptian-police-quickly-cracked-journalists-computer-password/> (8 December 2013).

[28] "Apple deluged by police demands to decrypt iPhones | Politics and Law", CNET News, <http://news.cnet.com/8301-13578_3-57583843-38/apple-deluged-by-police-demands-to-decrypt-iphones/> (8 December 2013).

[29] "UK Police Can Now Demand Encryption Keys", Schneier on Security, <https://www.schneier.com/blog/archives/2007/10/uk_police_can_n.html> (8 December 2013).

[30] G. Greenwald, "Glenn Greenwald: detaining my partner was a failed attempt at intimidation", The Guardian, <http://www.theguardian.com/commentisfree/2013/aug/18/david-miranda-detained-uk-nsa> (8 December 2013).

[31] "Victimless Crime Constitutes 86% of The Federal Prison Population", Libertarian News, <http://www.libertariannews.org/2011/09/29/victimless-crime-constitutes-86-of-the-american-prison-population/> (8 December 2013).

[32] "NSA director admits to misleading public on terror plots", salon.com, <http://www.salon.com/2013/10/02/nsa_director_admits_to_misleading_public_on_terror_plots/> (8 December 2013).

[33] "NSA illegally collected thousands of emails before Fisa court halted program", theguardian.com, <http://www.theguardian.com/world/2013/aug/21/nsa-illegally-collected-thousands-emails-court> (8 December 2013).

[34] "NSA broke privacy rules thousands of times per year, audit finds", Washington Post, <http://articles.washingtonpost.com/2013-08-15/world/41431831_1_washington-post-national-security-agency-documents> (8 December 2013).

[35] M. Zenko, "Americans Are as Likely to Be Killed by Their Own Furniture as by Terrorism", The Atlantic, <http://www.theatlantic.com/international/archive/2012/06/americans-are-as-likely-to-be-killed-by-their-own-furniture-as-by-terrorism/258156/> (8 December 2013).

[36] "You're Eight Times More Likely to be Killed by a Police Officer than a Terrorist", Cato @ Liberty, <http://www.cato.org/blog/youre-eight-times-more-likely-be-killed-police-officer-terrorist> (8 December 2013).

[37] "These 32 People Are Spending Their Lives In Prison For Nonviolent Crimes", Huffington Post, <http://www.huffingtonpost.com/2013/11/13/life-without-parole_n_4256789.html> (8 December 2013).

[38] M. Marlinspike, "Simplifying OTR deniability", Open WhisperSystems Blog, <https://www.thoughtcrime.org/blog/simplifying-otr-deniability/> (8 December 2013).

[39] M. Holtkamp, "Analysis of the OpenPGP and OTR protocols", <http://www.ai.rug.nl/mas/finishedprojects/2007/PieterMichiel/otr_gny.html> (8 December 2013).

[40] "Secure Function Evaluation vs. Deniability in OTR and similar protocols", .:: Phrack Magazine ::, <http://www.phrack.org/issues.html?issue=68&id=14> (8 December 2013).

[41] "Governments Admit They Carry Out False Flag Terror", The Big Picture, <http://www.ritholtz.com/blog/2013/12/governments-admit-they-carry-out-false-flag-terror/> (10 December 2013).

[42] "False flag", Wikipedia, <https://en.wikipedia.org/w/index.php?title=False_flag&oldid=585413085#As_pretexts_for_war> (10 December 2013).