Trust Darknet:
Control and Compromise in the Internet's Certificate Authority Model

Steven Roosa[1] and Stephen Schultze[2]

[1] Co-Chair, Privacy and Data Security Team, Holland & Knight LLP
[2] Associate Director, Center for Information Technology Policy at Princeton University

Abstract. For more than a decade, Internet users have relied upon digital certificates issued by certificate authorities to encrypt and authenticate their most valuable communications. Computer security experts have lambasted weaknesses in the system since its inception. A series of recent exploits have brought several problems back into stark focus. This paper describes some of the proposed technology-based improvements, as well as the structural shortcomings of the trust model – legal, economic, and organizational. We explore some of these structural defects in the context of lessons learned over the lifetime of the certificate authority trust model, and propose first steps toward fixes and next steps for study.

In the past two years, we have witnessed remarkable failures in the certificate authority (CA) regime. Although the regime purports to protect the communications of Internet users from malicious man-in-the-middle attacks, the trust model is premised on unconstrained authentication authority that is granted to thousands of entities scattered across the globe. Recent events have highlighted how difficult it can be to maintain a trustworthy system that is based upon this premise.

The certificate authority system exists to authenticate one party to another in a Public Key Infrastructure (PKI). Although client software ultimately carries out the authentication, CAs are the entities that issue the digital certificates which make the authentication possible. Software vendors, at their discretion, build into their products a list of "root" CAs that are to be trusted to perform authentication on behalf of users. The most common business for root CAs is the sale of SSL/TLS certificates to web site operators. These "Domain Validation" (DV) certificates indicate that the CA has verified that the web site operator owns the domain name in question. Some CAs contract with other companies, called Registration Authorities (RAs), to perform the actual verification of a certificate applicant's domain name ownership. Some Root CAs do not issue SSL certificates directly but instead cryptographically delegate that authority to a third party via a "Subordinate CA" (SubCA) certificate chain.[1] If the browser successfully "chains" the certificate to a trusted root CA, it indicates to the user that it is communicating with the true owner of the domain name rather than a man-in-the-middle.

Security researchers have frequently lamented the known weaknesses of the CA trust model and perennially announced new vulnerabilities in the underlying technology. Although these revelations have met with some fanfare, the core system has remained largely unchanged.

Recent High-Profile Compromises

In March of 2011, one of the most popular CAs – **Comodo** – admitted that one of its third-party *Registration Authorities* (RAs) had been successfully hacked. The attacker made off with certificates for high-profile domains like google.com [1]. In August of 2011, the public learned of a different attack. This time, the hacker obtained approximately 500 certificates from Dutch certificate authority DigiNotar, beginning as early as June of that year. **DigiNotar** discovered the breach a month later, took incomplete steps to revoke the certificates, and did not alert the public or software vendors of the risk. When an Iranian Gmail user noticed that one of the certificates was being used to attempt a man-in-the-middle attack on his communications, the major vendors revoked DigiNotar's trusted status, and the Dutch government – which relied on DigiNotar for its own PKI – took over company operations [2].

Both before and after the DigiNotar incident, a series of successful attacks on SubCAs have also come to light. In these cases, attackers used software vulnerabilities or yet-unknown espionage techniques to obtain valid private keys. These SubCA certificates were trusted by client software (including Microsoft Windows) for code-signing. In each case, a Root CA had signed the SubCA's private keys – sometimes without the knowledge of the software vendors who had approved the Root CA for trusted status. In 2010, researchers discovered that the **Stuxnet** virus was signed with private keys that corresponded to two different SubCAs run by well-known hardware manufacturers [3]. In 2011, a Malaysian SubCA that inherited its chained trust from the company **Entrust** was revealed to have been issuing certificates with key lengths shorter than required of Root CAs. Attackers used a well-known vulnerability related to short key lengths in order to spoof their own code-signing certificates and to then sign malware [4]. In March 2012, it was discovered that the keys of a Swiss Sub CA that chained to **Verisign** had similarly been used to sign malware [5]. Although we focus primarily on domain validation certificates in this article, the risks to code-signing certificates are nearly identical. The certificates in both instances are part of the CA trust model, and the problems of lack of transparency, ineffective audits, and a flawed legal architecture apply with equal force in both instances. Indeed, many SubCAs are trusted to issue both code-signing and domain validation certificates.

The DigiNotar removal is the first time that the major software vendors have penalized an active Root CA. Comodo was not penalized because its root key material was not compromised in the attacks on its RAs [6]. However, even if browsers become more stringent about root CA addition and removal, they will not have addressed the root of the problem. The root problem is not just a matter of better managing the list of root CAs – it is embedded in the structure of the system itself.

References

1. Comodo Report of Incident, 26 Mar. 2011; https://www.comodo.com/Comodo Fraud-Incident-2011-03-23.html.

2. S. Schultze, "DigiNotar Hack Highlights Critical Failures of our SSL Web Security Model," blog, 6 Sept. 2011; https://freedom-to-tinker.com/blog/sjs/diginotar-hack-highlights-critical-failures-our-ssl-web-security-model.

3. P. Bureau, "Win32/Stuxnet Signed Binaries," blog, 9 Aug. 2010; http://blog.eset.com/2010/07/19/win32stuxnet-signed-binaries.

4. J. Nightingale, "Revoking Trust in DigiCert Sdn. Bhd Intermediate Certificate Authority," blog, 23 Nov. 2011; https://blog.mozilla.org/security/2011/11/03/revoking-trust-in-digicert-sdn-bhd-intermediate-certificate-authority/.

5. V. Zakorzhevsky, "Mediyes–the dropper with a valid signature," blog, 15 Mar. 2012; https://www.securelist.com/en/blog/682/Mediyes_the_dropper_with_a_valid_ signature.

6. J. Nightingale, blog, Comodo Certificate Issue – Follow Up, 25 Mar. 2011; http://blog.mozilla.org/security/2011/03/25/comodo-certificate-issue-follow-up/.

Systemic Technical Weaknesses of Today's PKI

The computer security community has long focused on technical shortcomings of the CA trust model, and the recent breaches [see side bar] have amplified efforts to strengthen the system. The race to discover core cryptographic vulnerabilities and to design better algorithms will no doubt continue, but that dynamic is fairly well known. Instead, we briefly outline some of the more systemic technical weaknesses of the CA trust model as it stands.

Surface Area. The recent compromises have helped to highlight the diverse set of entities that hold broad-brush authority to issue certificates. The universe of Root CAs includes companies from around the world, governments, and defunct CAs that have re-sold their keys.[2] The Comodo incident in 2011, in which a hacker caused the issuance of unauthorized certificates for a number of high value domains, heightened awareness of the much larger number of RAs to which CAs outsource critical operations. Researchers have also begun to reveal the extent to which CAs have turned over the cryptographic keys to the kingdom by delegating chains of trust to others.[3]

Constrainability. As it stands, nearly *every* user of a given software package trusts the same list of root CAs, and they trust each of them with the ability to authenticate *any* web site. For instance, there is no practical means for users to restrict the CA of a national government to issue certificates only for entities within its borders. RFC 5280 includes optional "Name Constraints" that would limit the domains for which a given CA can issue certificates. However, this feature remains largely unsupported.

Trust Agility. Over time, new facts emerge that change the assessment of CA trustworthiness. In current software however, the list of root CAs resembles a write-only data structure in which incumbents retain their spot, excepting DigiNotar. In order to effectively remove DigiNotar in the wake of that CA's compromise, browsers and operating system vendors were forced to ship security updates or completely new binaries. This combination of technical, operational, and political stasis stands in opposition to what Moxie Marlinspike has termed "trust agility."[4] Empowering users with greater agility in their trust decisions can however present usability challenges.

Usability. Studies have repeatedly demonstrated that users do not understand the concept of trusted CAs, or even heed strongly worded security warnings that appear when authentication fails. Some researchers have concluded that it may be better to completely prevent users from engaging in dangerous behavior than to try to design for choice. Usability concerns can conflict with attempts to give users more control over the surface area, constraints, and trust agility of their root CA list.

Related Research on Technical Failures and Solutions in CA Trust Model

Patches. As problems with core cryptography or protocols are discovered, developers create patches. Historically, this has taken the form of fixes to specific vulnerabilities (eg. RFC 5746), improved cryptographic algorithms (eg. RFC 4270), and added features (eg. RFC 2459). These patches ultimately operate within the constraints of the same basic trust model.

Consistency Checks. Browser extensions like Certificate Patrol, http://patrol.psyced.org, are designed to alert users when certificates change or seem suspiciously inconsistent. Such extensions have enjoyed limited adoption because they require savvy users who understand the nature of digital certificates. More recent proposals, such as the Internet-Draft "Public Key Pinning Extension for HTTP", appear poised for greater adoption. These approaches take a Trust on First Use (ToFU) approach and simply terminate connections if the keys are inconsistent with those that were indicated in the first connection [1].

Consensus Tools. Researchers have created systems to help users determine whether other people are seeing the same key-domain pairs that they are seeing. The first such system, called "Perspectives," established a set of public key notaries run by trusted operators [2]. Follow-on work by Moxie Marlinspike, improved on this model by making the system distributed and more anonymous [3]. Other researchers proposed variations on yet another approach in which certificate-key pairs could be posted to a shareable write-only data structures on a first-come-first-serve basis. The Sovereign Keys project is one such example, https://www.eff.org/sovereign-keys. Nevertheless, the consensus approach has yet to be adopted natively by browsers or other clients, and it is unclear whether or not it will ultimately catch on.

Existing Trust Systems. Others have suggested that existing Internet trust systems could be leveraged into the traditional PKI system. Most notably, the DNS-Based Authentication of Named Entities (DANE) Proposed Standard aims to enable domain operators to place certificate information directly into their DNSSEC-signed DNS records [4]. For any given domain name, there is a single trust path dictated by the DNS hierarchy that chains up to the custodian of the top-level-domain (TLD) and ultimately to ICANN. It is unclear whether users will behave in a way that reflects changed trust. It may be reasonable to expect an Iranian user to recognize that an ".ir" domain is subject to eavesdropping by the regime, but it is unclear whether an ".ly" domain would signal to the average user that Libya holds the keys to their communications.

[1] C. Evans, C. Palmer, and R. Sleevi, *Public Key Pinning Extension for HTTP (Draft 4)*, IETF Internet Draft, work in progress, December 7 2012; https://tools.ietf.org/html/draft-ietf-websec-key-pinning.

[2] D. Wendlandt, D.G. Andersen, and A. Perrig, *Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing*, In Proc. ATC'08: USENIX 2008; http://www.cs.cmu.edu/~dga/papers/perspectives-usenix2008.pdf.

[3] M. Marlinspike, *SSL and the Future of Authenticity: Moving Beyond Certificate Authorities*, BlackHat USA 2011, July 2011; http://www.securitytube.net/video/2203.

[4] P. Hoffman, J. Schlyter, *The DNS Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol (TLSA)*, IETF RFC 6698, Aug. 2012, http://www.rfc-editor.org/rfc/rfc6698.txt.

Legal, Economic, and Organizational Flaws

An implementation of the certificate authority trust model that conforms perfectly to the technical specifications can nevertheless manifest deep flaws. Augmenting or replacing the technical infrastructure may similarly fail if it does not also address some of the more fundamental problems and assumptions that underlie today's model.

CA Liability and Economic Incentives

Third-party trust problems are nothing new. Steve Bellovin has noted that in the early days of electric communication, the telegraph company's liability and economic incentives were unsettling. One author at the time noted,

> *"On the Continent it is frequently the case that the signatures of messages involving, for instance, money payments or delivery of valuable documents, purport to be certified by the telegraph operator…" but the telegraph company will not "back up [a guarantee] with an admission of their own liability in the event of a fraud occurring."*[5]

Unfortunately, the documents that serve as the legal architecture of the CA trust model today—the Certification Practice Statement (CPS), Certificate Policy, Subscriber Agreement, and Relying Party Agreement—reflect a strikingly parallel situation. The CAs do not seem to have much faith in the product that they provide.

For instance, it is customary for a CPS to include a total disclaimer of all liability for any claim or loss arising out of a certificate "that was issued as a result of errors, misrepresentations, or other acts or omissions of a Subscriber or any other person, entity, or organization".[6] This means that if a bad actor obtains a certificate by either tricking or hacking the CA, an RA, or a SubCA, and the bad actor then uses the certificate for a successful man-in-the-middle attack against an end-user, the CPS says that the CA, RA, and SubCA have no liability. To the extent the CPS leaves room for any liability, it often includes substantial caps on aggregate liability, often on a "per certificate" basis apportioned among those claims that are filed first.[7] In fact, it is unclear whether any such claim has ever successfully been brought.

These types of disclaimers are unsurprising, given the "Baseline Guidelines" supplied by the leading CA industry trade group, the CA Browser Forum, which state: "If the CA has not issued or managed the Certificate in compliance with [the CA Browser Forum's Requirements] and its Certificate Policy and/or Certification Practice Statement, the CA MAY seek to limit its liability to the

Subscriber and to Relying Parties, regardless of the cause of action or legal theory involved, for any and all claims, losses or damages suffered as a result of the use or reliance on such Certificate by any appropriate means that the CA desires."[8] These provisions allow the CA to sell certificates while seemingly off-loading all of the significant downside legal risk associated with the sale.

The CA legal documents often purport to legally bind end-users (also referred to as "relying parties" in the model) merely because the end-user's client software relies on the CA's certificates. Due to the obvious absence of notice, assent, and meeting of the minds, it appears a relatively sure bet that both the CPS and the Relying Party Agreement are unenforceable as contracts against relying parties. So why does this legal architecture persist? Perhaps because the CA audit framework published by the American Institute of Certified Public Accountants and the Canadian Institute of Chartered Accountants (WebTrust Framework) actively encourages CAs to post their CPS documents, but makes no mention of actual notice or assent of the relying party.[9] RFC 3647 takes the same approach and states that it is permissible for CAs to have disclaimers of warranties, disclaimers of liability, and other legal provisions appear in a CA's legal documents and that mere "publication and posting to a repository" is sufficient "for the purpose of communicating to a wide audience of recipients, such as all relying parties."[10]

The CAs have embraced the approach. CAs routinely copy WebTrust's "illustrative disclosures" into their CPS and Relying Party Agreement. These model provisions address indemnity, disclaimer of fiduciary duties, governing law, mandatory dispute resolution, and supposed relying party obligations. Many CAs no doubt believe their CPS is actually enforceable as a result of the CA Browser Forum, WebTrust, and RFC guidance. Unfortunately for the model, there is no court decision in the United States that holds that any of the CA documents are enforceable against relying parties based on the mere posting of on-line documents or that CAs are excused from the standard precepts governing the law of contracts.

The problems with the model's legal architecture create economic incentives for CAs that are at best uncertain, and at worse perverse. Those CAs that believe their CPS is enforceable may be incentivized to emphasize higher sales volume over quality business practices. These CAs could perceive that the downside risk associated with aggressive reselling via RAs or SubCAs has been minimized or eliminated by the CPS. This tendency is reinforced by the highly price competitive market for certificates in which volume is paramount for survival and penalties for untrustworthy behavior have been virtually non-existent. Furthermore, customers of CAs—web site operators—gain no benefit from purchasing certificates from a more trustworthy CA, because any standard certificate looks and works the same in all client software. Certificates have become unregulated commodities. These factors conspire to create an unfortunate "race to the bottom" in CA security practices.

Audits and Transparency

The WebTrust Framework, and the CA Browser Forum Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates, together with the requirements of individual software vendors, form the de facto compliance regime for CAs. Many of the requirements are sound and uncontroversial. However, the current regime falls far short of covering certain entities that carry out critical CA functions. The regime also fails to require that their identity be disclosed to the public. As a direct result, CAs structure their businesses in a way that creates significant zones of un-audited and undisclosed certificate granting authority.

One area of concern involves companies, RAs, that are external to the CA but that have been granted the partial or complete ability to conduct identity verification. Although these RAs do not typically hold private key material, they have the operational ability to cause the issuance of certificates. WebTrust decided to “carve out” RA operations from the scope of CA audits. They admitted that “some end users” might not find this satisfactory but claimed that they had “concluded that the issuance and use of [the Webtrust Framework] was desirable and that the impact of a third-party registration function was beyond the scope of this document.”[11]

The WebTrust Framework went un-modified for over a decade, until Version 2.0 of the document was abruptly published without fanfare in mid-2011. This new version continued to leave the vast majority of RAs and RA functions beyond the reach of any external audit. Although an auditor is not technically forbidden from auditing RA operations, WebTrust 2.0 considers such audits to be "rare situations" warranted only in circumstances in which "the CA exercises extensive monitoring controls (including onsite audit) over all aspects of the RA operations and the CA is willing to assert to the effectiveness of the controls performed by the external RAs . . . ."[12] In this statement, WebTrust has in fact laid bare the severity of the RA problem by implying that it is "rare" that a CA would exercise "extensive monitoring controls . . . over RA operations" or "be willing to assert to the effectiveness of the controls performed by the external RAs." However, because RAs perform identity verification, they are often the first and last line of defense against the issuance of fraudulently obtained certificates. NIST's Information Technology Laboratory Bulletin for July of 2012 identified the four overarching categories of CA compromise, two of which focused almost entirely on the RA: "impersonation," those circumstances where a certificate applicant fools the RA into causing a fraudulent certificate to be issued, and "RA compromise," those circumstances where the RA's certificate request process is compromised and the hacker is able to make certificate requests to the CA.[13]

Moreover, it appears that even in those "rare" situations when there may be audit activity with respect to the RA, the auditor does not appear to have the ability to unilaterally require an RA audit. The WebTrust 2.0 guidelines state "the CA and the auditor need to agree in advance with this approach, including the extent and sufficiency of controls being exercised." Thus, the WebTrust 2.0 criteria appear to allow the CA to set the terms of RA "audits," if any, and to shop for an auditor that agrees to take their preferred approach. Compounding the problem with the audit regime is perhaps a more fundamental issue: CAs are not required to disclose the identity or track record of their RAs. A relying party or end-user trusts the RA as much as the CA, yet the RAs are unknown. This makes managing trust almost impossible. NIST's bulletin exhorts companies and other organizations to "[r]emove any trust anchors that should not be trusted," but how can an organization as a relying party even begin that exercise without knowing the identity of all of the RAs used by any particular CA?

Another problematic practice is the cryptographic delegation of complete certificate-granting powers by CAs to third parties via a certificate chain. WebTrust does not require that these so-called SubCAs be audited or disclosed to the public. Several CAs sell costly SubCA certificates, despite the fact that they have no technical means for monitoring their use. These SubCAs are typically intended for use by an enterprise user that wishes to generate a large number of SSL certificates or email (S/MIME) certificates for its domains. Many CAs will "cross sign" other CAs' certificates such that a user that does not trust the cross-signed CA directly will nevertheless trust it via the signer's authority. These relationships are likewise often not disclosed when software vendors approve or consider removing the signing CA from the root CA.

In February of 2012, CA Trustwave admitted to having issued a SubCA certificate to a company for the purpose of performing a man-in-the-middle attack on all HTTPS browsing activity of its employees. Trustwave revoked the certificate, and pledged that it would issue no similar certificates in the future.[14] At the same time, it claimed that, "It has been common practice for Trusted CAs to issue subordinate roots for enterprises for the purpose of transparently managing encrypted traffic." In January of 2013, it was discovered that a different CA, Turktrust, had issued a SubCA certificate to a Turkish government office, which subsequently installed it on a man-in-the-middle proxy. Turktrust claimed the issuance was an error—they had intended to issue an SSL certificate—and that the proxy had affected only employees of that office.[15]

These practices essentially create a "trust darknet" with a risk surface area that far exceeds the size of the audited CA universe. It is also worth stating that audits themselves are far from perfectly suited silver bullets that ensure trustworthy practices. To begin with, the audit simply confirms that the processes stated in the CPS are in place. The public output of the audit process is typically a pro forma one or two-page attestation to this effect. DigiNotar, audited by PriceWaterhouseCoopers under the ETSI 101.456 standard *and* the WebTrust Extended Validation Audit Criteria, now serves as a reminder that simply obtaining an audit attestation does not guarantee trustworthy operations.

Jurisdiction and Communities of Trust

The jurisdiction in which a CA is located, and in which its affiliates and delegates operate, affects whether an individual should trust a particular CA. For instance, because governments have the power to compel CAs within their jurisdiction to issue unauthorized SubCA certificates for the purpose of spying on encrypted traffic such as email, citizens of autocratic or untrustworthy political regimes may wish to trust only CAs located beyond the reach of their government.[16] Similarly, companies may wish to avoid trusting CAs that are either affiliated with, or potentially controlled by, governments that they believe would facilitate industrial espionage on behalf of state or private competitors in that jurisdiction. However, CAs do not currently disclose enough information for even vigilant users to know which jurisdictions have influence over the certificates that they rely upon—especially certificates emanating from RAs, SubCAs, and cross-signed CAs. Currently, the CA Browser Forum guidelines only require the country of the CA to be disclosed. The identities of RAs, together with the jurisdictions in which they reside, are completely invisible in the certificate authority trust model. If a relying party wishes to avoid trust being anchored in an entity located in jurisdiction X, there is no way under the current model to enforce that choice. CAs which purport to be located in jurisdiction Y may also have RAs in jurisdiction X.

Location—i.e. the location of the CA, RAs, SubCAs, cross-signed CAs, and of the relying party—is only one of many possibly relevant trust factors. Other factors include track record, parent/subsidiary affiliation, number of outstanding certificates, and global reach. Researchers suggest that one technical-structural approach to consider might be to enable like-minded relying parties to curate their own root CA lists. Inspired by the success of customized "ad block" lists, a few dedicated users might create and maintain tailored root CA lists for the benefit of a much larger community. Greater CA transparency might go a long way to enabling such tools. More research should be done on how to enable trust agility for users that have different trust profiles while also facilitating a low barrier "set it and forget it" user experience.

Strategies for Improvement

The problems with the CA trust model have not placed it beyond redemption. There are three categories of discrete improvements that could make the model significantly better. First transparency could enable meaningful choice by relying parties. The current lack of transparency impairs the ability of relying parties to know the identity of RAs, the identity of all SubCAs and cross-signed CAs, and the jurisdiction in which the RAs, SubCAs, and cross-signed CAs reside and carry out operations. The lack of transparency prevents software developers from having sufficient data sources to provide solutions that would allow end-users to trust or un-trust CAs based on this information. To improve transparency and choice:

- CAs should be required to make complete on-line disclosure of the identity and legal jurisdiction of all of their RAs, SubCAs, and cross-signed CAs,
- CAs should be required to disclose governmental affiliation, ownership, and control of themselves, their RAs, SubCAs, and cross-signed CAs, and
- CAs must be advised by self-regulatory bodies that blanket liability disclaimers in CPs, CPSs, and RPAs should be accompanied by some degree of at least one-time actual notice to relying parties.

The second problem area is audits. The CA audit regime could be improved in the following ways:

- Any party that performs identity verification or can cause the issuance of certificates should be audited at the same level as a root CA and
- Self-regulatory bodies such as the CA Browser Forum should require more detailed information regarding audit results to be made public (i.e. something beyond a pro forma 2-page attestation).

The third area relates to the self-regulatory process itself. Although the CA Browser Forum has made some significant improvements in its requirements for certificate issuance, its internal processes are burdened by opacity and limited participation. Accordingly, self-regulatory bodies should:

- Conduct their work in a manner more consistent with disclosure security, and
- Continue to broaden participatory scope, especially by representatives of the relying party community.

Conclusion

The CA trust model has global reach and pervasive deployment. Although there are systems that have been proposed to help enhance the reliability of the CA trust model, there are no comprehensive replacements on the horizon. Moreover, the model has much to recommend it in terms of scalability, elegance, capacity for evolution, and collaborative solutions. It also has substantial institutional commitments from the software and vendor industries. If its transparency, audits, and self-regulation were improved in the ways noted, it may be structurally sound enough to survive as the foundation of trust.

[1] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, IETF RFC 5280, May 2008; http://www.rfc-editor.org/rfc/rfc5280.txt.

[2] See, eg: Bug 242610 - Add USERTRUST Root certificates; https://bugzilla.mozilla.org/show_bug.cgi?id=242610#c7

[3] P. Eckersley and J. Burns, *An Observatory for the SSLiverse, SSL Observatory,* Electronic Frontier Foundation. SSL Observatory, DefCon 18, July 2010; https://www.eff.org/files/DefconSSLiverse.pdf.

[4] M. Marlinspike, *SSL and the Future of Authenticity: Moving Beyond Certificate Authorities*, BlackHat USA 2011, July 2011; http://www.securitytube.net/video/2203.

[5] Bellovin, "SSL Failings" presentation at the Workshop on The Future of User Authentication and Authorization on the Web, Financial Cryptography 2011. Quoting Robert Slater, Banking Telegraphy: Combining Authenticity, Economy, and Secrecy. London, 1876.

[6] Entrust Certificate Services, Entrust Limited: Certification Practice Statement, v.2.6, sec. 2.1.2 and 2.2.1.4, February 28, 2011; http://www.entrust.net/CPS/pdf/ssl-cps-english-28-02-11-v2-6.pdf.

[7] Digicert Inc.: Certification Practice Statement, v. 4.04, sec. 9.8, May 10, 2012; http://www.digicert.com/docs/cps/DigiCert_CPS_v404-may-10.pdf.

[8] CA/Browser Forum: Baseline Requirements for the Issuance and Management of Publicly Trusted Certificates , v. 1.0, November 22, 1011; https://www.cabforum.org/Baseline_Requirements_V1.pdf.

[9] Canadian Institute of Chartered Accountants: Trust Service Principles and Criteria for Certification Authorities, v. 2.0, March 2011; WebTrust Program for Certification Authorities, v. 1.0, August 25, 2000; http://www.webtrust.org/homepage-documents/item54279.pdf .

[10] S. Chokhani, W. Ford, R. Sabett, C. Merrill, S. Wu, *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, IETF RFC 3647, November 2003; http://www.rfc-editor.org/rfc/rfc3647.txt.

[11] American Institute of Certified Public Accountants and Canadian Institute of Chartered Accountants: WebTrust Program for Certification Authorities, v. 1.0, footnote 4, August 25, 2000; http://www.webtrust.org/homepage-documents/item65306.pdf.

[12] Canadian Institute of Chartered Accountants: Trust Service Principles and Criteria for Certification Authorities, v. 2.0, p. 13, March 2011; WebTrust Program for Certification Authorities, v. 1.0, August 25, 2000; http://www.webtrust.org/homepage-documents/item54279.pdf .

[13] P. Turner, W. Polk, E. Barker, "Preparing for and Responding to Certification Authority Compromise and Fraudulent Certificate Issuance," ITL Bulletin for July 2012, National Institute of Standards and Technology (NIST), U.S. Dept. of

Commerce, July 2012; http://csrc.nist.gov/publications/nistbul/july-2012_itl-bulletin.pdf.

[14] Trustwave Spider Labs' Official Blog, "Clarifying the Trustwave CA Policy Update," February 4, 2012; http://blog.spiderlabs.com/2012/02/clarifying-the-trustwave-ca-policy-update.html

[15] Turktrust Public Announcements; http://turktrust.com.tr/en/kamuoyu-aciklamasi-en.html

[16] Soghoian, C. and Stamm, S.: "Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL," Financial Cryptography and Data Security 2011 (2011)