Trust Darknet

Control and Compromise in the Internet's Certificate Authority Model

For more than a decade, Internet users have relied on 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. Recent exploits have brought several problems back into stark focus. The authors describe some proposed technology-based improvements, as well as some legal, economic, and organizational shortcomings of the trust model. They also propose first steps toward fixes and next steps for study.

Steven B. Roosa
Holland & Knight

Stephen Schultze
Princeton University

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

The CA system exists to authenticate one party to another in a public-key infrastructure (PKI). Although client software ultimately carries out the authentication, CAs issue the digital certificates that make the authentication possible. Software vendors, at their discretion, build into their products a list of "root" CAs that are trusted to perform authentication on behalf of users. The most common business for root CAs is the sale of SSL/TLS certificates to website operators. These *domain validation* (DV) certificates indicate that the CA has verified that the website 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 don't 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's communicating with the domain name's true owner rather than a man in the middle.

Security researchers have frequently lamented the CA trust model's known

1089-7801/13/$31.00 © 2013 IEEE

Recent High-Profile Compromises

In March 2011, one of the most popular certificate authorities (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 such as google.com (see www.comodo.com/Comodo-Fraud-Incident-2011-03-23.html). In August 2011, the public learned of a different attack. This time, the hacker obtained approximately 500 certificates from Dutch CA DigiNotar, beginning as early as June of that year. DigiNotar discovered the breach a month later, took incomplete steps to revoke the certificates, and didn't 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 public-key infrastructure (PKI) – took over company operations.[1]

Both before and after the DigiNotar incident, a series of successful attacks on SubCAs came 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.[2] 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 have also used a well-known vulnerability related to short key lengths to spoof their own code-signing certificates and then sign malware.[3] In March 2012, it was discovered that the keys of a Swiss SubCA that chained to Verisign had similarly been used to sign malware.[4] Although we focus primarily on domain validation certificates in the main text, 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 wasn't penalized because its root key material wasn't compromised in the attacks on its RAs.[5] However, even if browsers become more stringent about root CA addition and removal, they will not have addressed the root of the problem, which isn't just a matter of better managing the list of root CAs – but is embedded in the system structure itself.

References

1. 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-websecurity-model.
2. P. Bureau, "Win32/Stuxnet Signed Binaries," blog, 9 Aug. 2010; http://blog.eset.com/201%7/19/win32stuxnet-signed-binaries.
3. 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/.
4. V. Zakorzhevsky, "Mediyes – The Dropper With a Valid Signature," blog, 15 Mar. 2012; www.securelist.com/en/blog/682/Mediyes_the_dropper_with_a_valid_ signature.
5. J. Nightingale, "Comodo Certificate Issue – Follow Up, blog, 25 Mar. 2011; http://blog.mozilla.org/security/2011/03/25/comodo-certificate-issue-follow-up/.

weaknesses and perennially announce new vulnerabilities in the underlying technology. Although these revelations have met with some fanfare, the core system has remained largely unchanged.

Systemic Technical Weaknesses

The computer security community has long focused on the CA trust model's technical shortcomings, and recent breaches (see the "Recent High-Profile Compromises" sidebar) have amplified efforts to strengthen the system. The race to discover core cryptographic vulnerabilities and 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

Recent compromises have helped 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 (see https://bugzilla.mozilla.org/show_bug.cgi?id=242610#c7). The Comodo incident in 2011, in which a hacker caused an RA to trigger the CA Comodo to issue unauthorized certificates for several 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.[2]

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 website. For instance, no practical means exists 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.[1] 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 spots, excepting DigiNotar. To effectively remove DigiNotar in the wake of that CA's compromise, browsers and operating system vendors had 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."[3] However, empowering users with greater agility in their trust decisions can present usability challenges.

In current software, the list of root CAs resembles a write-only data structure in which incumbents retain their spots, excepting DigiNotar.

Usability

Studies have repeatedly demonstrated that users don't understand the concept of trusted CAs, or even heed strongly worded security warnings that appear when authentication fails. Some researchers have concluded that it might 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 their root CA list's surface area, constraints, and trust agility.

Legal, Economic, and Organizational Flaws

An implementation of the CA trust model that conforms perfectly to the technical specifications can nevertheless manifest deep flaws. Augmenting or replacing the technical infrastructure might similarly fail if it doesn't 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. As 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."[4]

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 don't seem to have much faith in the product that they provide.

For instance, a CPS customarily includes 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."[5] This means that if a bad actor obtains a certificate by either tricking or hacking the CA, an RA, or a SubCA, and then uses that 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 that the CPS leaves room for any liability, it often includes substantial caps on aggregate liability, typically on a "per certificate" basis apportioned among those claims that are filed first.[6] In fact, it's unclear whether anyone has ever successfully brought any such claim.

These types of disclaimers are unsurprising, given the "baseline guidelines" supplied by the leading CA industry trade group, the CA Browser Forum, which state the following:

> 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.[7]

These provisions let the CA sell certificates while seemingly offloading 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 user's client software relies on the CA's certificates. Due to the obvious absence of notice, assent, and meeting of the minds, it seems a relatively sure bet that both the CPS and the relying party agreement are unenforceable as contracts against relying parties. So, why does this purported 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 (the WebTrust Framework) actively encourages CAs to post their CPS documents, but doesn't require actual notice to, or assent of, the relying party.[8,9] RFC 3647 takes the same approach and states that CAs can have disclaimers of warranties, disclaimers of liability, and other legal provisions appear in their 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 this approach. They routinely copy WebTrust's "illustrative disclosures" into their CPS and relying party agreements. 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, no court decision in the US holds that any of the CA documents are enforceable against relying parties based on the mere posting of online documents or that CAs are excused from the standard precepts governing contract law.

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 might be incentivized to emphasize higher sales volume over quality business practices. These CAs could perceive that the CPS has minimized or eliminated the downside risk associated with aggressive reselling via RAs or SubCAs. This tendency is reinforced by the highly priced competitive market for certificates in which volume is paramount for survival, and penalties for untrustworthy behavior have been virtually nonexistent. Furthermore, CA customers — website 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.

CA customers gain no benefit from purchasing certificates from a more trustworthy CA; any standard certificate works the same in all client software.

Audits and Transparency

The WebTrust Framework and the CA Browser Forum baseline requirements for issuing and managing publicly trusted certificates, together with individual software vendors' requirements, 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 these entities' identities be disclosed to the public. Consequently, CAs structure their businesses in a way that creates significant zones of unaudited and undisclosed certificate-granting authority.

One area of concern involves companies, RAs, that are external to the CA but that have partial or complete ability to conduct identity verification. Although these RAs don't typically hold private-key material, they verify identity and then submit a request to the CA, which results in the CA issuing a certificate, often in an automated fashion. WebTrust decided to "carve out" RA operations from the scope of CA audits. It admitted that "some end users" might not find this satisfactory but claimed that it 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."[8]

The WebTrust Framework went unmodified for more than a decade, until version 2.0 was abruptly published without fanfare in mid 2011.[9] This new version continued to leave the vast majority of RAs and RA functions beyond the reach of any external audit. Although an auditor isn't 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." In this statement, WebTrust 2.0 has in fact laid bare the severity of the RA problem by implying that it's "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're often the first and last line of defense against fraudulently obtained certificates.

Because RAs perform identity verification, they're often the first and last line of defense against fraudulently obtained certificates.

The US National Institute of Standards and Technology's (NIST's) *Information Technology Laboratory Bulletin* for July 2012 identified the four overarching categories of CA compromise, two of which focused almost entirely on the RA: *impersonation*, or those circumstances in which a certificate applicant fools the RA into causing the CA to issue a fraudulent certificate, and *RA compromise*, or circumstances in which the RA's certificate request process to the CA is compromised, and the hacker can make certificate requests to the CA, as if the hacker were the RA.[11]

Moreover, it appears that even in those "rare" situations when audit activity might occur with respect to the RA, the auditor doesn't appear to be able to unilaterally require an RA audit. The WebTrust 2.0 guidelines state that "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 let the CA 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 don't have to disclose their RAs' identity or track record. A relying party or user has no choice but to trust 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 "remove 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 2.0 doesn't require that these so-called SubCAs be audited or disclosed to the public. Several CAs sell costly SubCA certificates, even though they have no technical means for monitoring the certificates' use. These SubCAs are typically intended for an enterprise user who wishes to generate a large number of SSL certificates or email (S/MIME) certificates for its domains. Many CAs will also "cross sign" other CAs' certificates such that a user who doesn't trust the cross-signed CA directly will nevertheless trust it via the signer's authority. These relationships likewise often aren't disclosed when software vendors approve or consider removing the signing CA from the root CA.

In February 2012, CA Trustwave admitted to issuing a SubCA certificate to a company so that the latter could perform a man-in-the-middle attack on all it employees' HTTPS browsing activity. Trustwave revoked the certificate, and pledged that it would issue no similar certificates in the future.[12] 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 2013, a different CA, Turktrust, was found to have 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 — it had intended to issue an SSL certificate — and that the proxy had affected only that office's employees (see http://turktrust.com.tr/en/kamuoyu-aciklamasi-en.html).

These practices essentially create a "trust darknet" with a risk surface area that far exceeds the size of the audited CA universe. Note also that audits themselves are far from perfectly suited silver bullets that ensure trustworthy practices. Initially, the audit simply confirms that the processes stated in the CPS are in place. The audit process's public output 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, reminds us that simply obtaining an audit attestation doesn't guarantee trustworthy operations.

Jurisdiction and Communities of Trust

The jurisdiction in which a CA is located and where its affiliates and delegates operate affects whether an individual should trust it. For instance, because governments can compel CAs within their jurisdiction to issue unauthorized SubCA certificates to spy on encrypted traffic such as email, citizens of autocratic or untrustworthy political regimes might wish to trust only CAs located beyond their governments' reach.[13] Similarly, companies might 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 don't currently disclose enough information for even vigilant users to know which jurisdictions have influence over the certificates that users rely on — especially certificates emanating from RAs, SubCAs, and cross-signed CAs. Currently, the CA Browser Forum guidelines require that only the CA's country be disclosed. RAs' identities, together with the jurisdictions in which they reside, are completely invisible in the CA trust model. If a relying party wishes to avoid trust being anchored in an entity located in jurisdiction X, the current model offers no way to enforce that choice. CAs that purport to be located in jurisdiction Y might also have RAs in jurisdiction X.

Location — the location of the CA, RAs, SubCAs, cross-signed CAs, and the relying party — is only one of many possibly relevant trust factors. Others include track record, parent/subsidiary affiliation, number of outstanding certificates, and global reach. One technical-structural approach to consider might be enabling 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 larger community's benefit. 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.

Inspired by the success of customized "ad block" lists, a few dedicated users might create and maintain tailored root CA lists for the larger community's benefit.

Strategies for Improvement

The problems with the CA trust model haven't placed it beyond redemption. Three categories of discrete improvements could make the model significantly better. First, transparency could enable meaningful choice by relying parties. The current lack of transparency impairs relying parties' ability 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. This 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 online disclosure of the identity and legal jurisdiction of all of their RAs, SubCAs, and cross-signed CAs;
- be required to disclose governmental affiliation, ownership, and control of themselves, their RAs, SubCAs, and cross-signed CAs; and
- 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 CA to issue certificates should be audited at the same level as a root CA.
- Self-regulatory bodies such as the CA Browser Forum should require more detailed information regarding audit results to be made public (that is, something beyond a pro forma two-page attestation).

The third area relates to the self-regulatory process. 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.

The CA trust model has global reach and pervasive deployment. Although systems have been proposed to help enhance this model's reliability, no comprehensive replacements are 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 improved in the ways noted, it might be structurally sound enough to survive as the foundation of trust.

References

1. D. Cooper et al., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, IETF RFC 5280, May 2008; www.rfc-editor.org/rfc/rfc5280.txt.
2. P. Eckersley and J. Burns, "An Observatory for the SSLiverse," *SSL Observatory*, Electronic Frontier Foundation, 18 July 2010; www.eff.org/files/DefconSSLiverse.pdf.
3. M. Marlinspike, "SSL and the Future of Authenticity: Moving beyond Certificate Authorities," *Proc. BlackHat USA Conf.,* UBM Tech, 2011; www.securitytube.net/video/2203.
4. S. Bellovin, "SSL Failings," *Workshop on the Future of User Authentication and Authorization on the Web*, featured presentation, 2011.
5. "Entrust Limited: Certification Practice Statement, v.2.6," Entrust Certificate Services, 28 Feb. 2011; www.entrust.net/CPS/pdf/ssl-cps-english-28-02-11-v2-6.pdf.
6. "Digicert Inc.: Certification Practice Statement, v. 4.04," Digicert, 10 May 2012; www.digicert.com/docs/cps/DigiCert_CPS_v404-may-10.pdf.
7. "Baseline Requirements for the Issuance and Management of Publicly Trusted Certificates, v. 1.0," CA/Browser Forum, 22 Nov. 2011; www.cabforum.org/Baseline_Requirements_V1.pdf.
8. "WebTrust Program for Certification Authorities, v. 1.0," American Inst. of Certified Public Accountants and Canadian Institute of Chartered Accountants, 25 Aug. 2000; www.webtrust.org/homepage-documents/item27839.aspx.
9. "Trust Service Principles and Criteria for Certification Authorities, v. 2.0," Canadian Inst. of Chartered Accountants, Mar. 2011; www.webtrust.org/homepage-documents/item54279.pdf.
10. S. Chokhani et al., *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, IETF RFC 3647, Nov. 2003; www.rfc-editor.org/rfc/rfc3647.txt.
11. P. Turner, W. Polk, and E. Barker, "Preparing for and Responding to Certification Authority Compromise and Fraudulent Certificate Issuance," *ITL Bulletin*, Nat'l Inst. of Standards and Technology (NIST), July 2012; http://csrc.nist.gov/publications/nistbul/july-2012_itl-bulletin.pdf.

Related Research on Failures and Solutions in CA Trust Model

The CA trust model has been the subject of significant innovation in an effort to increase its reliability.

Patches

As problems with core cryptography or protocols are discovered, developers create patches. Historically, this has taken the form of fixes to specific vulnerabilities,[1] improved cryptographic algorithms,[2] and added features.[3] These patches ultimately operate within the constraints of the same basic trust model.

Consistency Checks

Browser extensions such as 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.[4]

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.[5] Follow-on work by Moxie Marlinspike improved on this model by making the system distributed and more anonymous.[6] Other researchers proposed variations on yet another approach in which certificate-key pairs could be posted to shareable, write-only data structures on a first-come-first-serve basis. The Sovereign Keys project is one such example (www.eff.org/sovereign-keys). Nevertheless, the consensus approach has yet to be adopted natively by browsers or other clients, and it's unclear whether 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.[7] For any given domain name, a single trust path is dictated by the DNS hierarchy that chains up to the custodian of the top-level-domain (TLD) and ultimately to ICANN. It's unclear whether users will behave in a way that reflects changed trust. It might be reasonable to expect an Iranian user to recognize that an ".ir" domain is subject to eavesdropping by the regime, but it's unclear whether an ".ly" domain would signal to the average user that Libya holds the keys to their communications.

References

1. E. Rescorla et al., *Transport Layer Security (TLS) Renegotiation Indication Extension*, IETF RFC 5746, Feb. 2010; https://tools.ietf.org/html/rfc5746.
2. P. Hoffman and B. Schneier, *Attacks on Cryptographic Hashes in Internet Protocols*, IETF RFC 4270, Nov. 2005; www.ietf.org/rfc/rfc4270.txt.
3. R. Housley et al., *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, IETF RFC 2459, Jan. 1999; www.ietf.org/rfc/rfc2459.txt.
4. C. Evans, C. Palmer, and R. Sleevi, "Public Key Pinning Extension for HTTP (Draft 4)," IETF Internet draft, work in progress, Dec. 2012.
5. D. Wendlandt, D.G. Andersen, and A. Perrig, "Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing," *Proc. Usenix 2008 Ann. Technical Conf.* (ATC 08); Usenix Assoc., 2008, pp. 321–334; www.cs.cmu.edu/~dga/papers/perspectives-usenix2008.pdf.
6. M. Marlinspike, "SSL and the Future of Authenticity: Moving beyond Certificate Authorities," *Proc. BlackHat USA Conf.,* UBM Tech, 2011; www.securitytube.net/video/2203.
7. P. Hoffman and J. Schlyter, *The DNS Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol (TLSA)*, IETF RFC 6698, Aug. 2012; www.rfc-editor.org/rfc/rfc6698.txt.

12. "Clarifying the Trustwave CA Policy Update," Trustwave Spider Labs' blog, 4 Feb. 2012; http://blog.spiderlabs.com/2012/02/clarifying-the-trustwave-ca-policy-update.html.
13. C. Soghoian and S. Stamm, "Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL," *Financial Cryptography and Data Security,* LNCS 7035, Springer, 2012, pp. 250–259.

Steven B. Roosa is a partner in Holland & Knight's New York office and cochair of its Data Privacy and Security team. His practice focuses on advising companies on mobile app privacy compliance, Internet tracking, Web security, geo-fencing, certification authority matters pertaining to online trust, and Web-based reputation issues. Roosa holds a law degree from Rutgers School of Law and is a fellow at the Center for Information Technology Policy at Princeton University. Contact him at steven.roosa@hklaw.com.

Stephen Schultze is associate director at the Center for Information Technology Policy at Princeton University. His work includes Internet privacy, computer security, government transparency, and telecommunications policy. Schultze has a BA in computer science from Calvin College and a masters in comparative media studies from the Massachusetts Institute of Technology. He served as a fellow at the Berkman Center for Internet and Society at Harvard University. Contact him at sjs@princeton.edu.