Intellectual Property & Technology Law Journal

Edited by the Technology and Proprietary Rights Group of Weil, Gotshal & Manges LLP

ASPEN PUBLISHERS

VOLUME 22 • NUMBER 11 • NOVEMBER 2010

Encrypted communications, which are used to secure every Internet transaction and confidential exchange of information imaginable, have been considered relatively safe. However, as explained in this article by **Steven B. Roosa**, a partner in the law firm of Reed Smith LLP, and **Stephen Schultze**, the Associate Director of the Center for Information Technology Policy at Princeton University, recently revealed technical and operational vulnerabilities highlight significant defects in the underlying trust and authentication system that uses third parties called "Certificate Authorities" to vouch for the identity of Web sites to end-users.

Enforcement of IP rights against the government is not as easy as enforcement against a private party. But, according to **David S. Bloch**, a partner in the San Francisco office of Winston & Strawn LLP, and **James G. McEwen**, a partner in the Washington, DC, office of Stein McEwen LLP, it can be done, effectively and efficiently, if a litigant takes the time to identify the appropriate sovereign immunity waivers, selects the appropriate tools, and seeks only the damages that the government has authorized.

Assignor estoppel is a little-known tool that can have a big impact on inventor/assignors and patent owner/assignees. Here, **Peter E. Strand**, the senior partner in the Washington, DC, office of Shook, Hardy & Bacon L.L.P., explains that an awareness of the rule can be of critical importance.

W. Michael Ryan and **Christopher M. Loeffler** of Kelley Drye & Warren LLP examine the contracting issues relating to public cloud computing, as well as the privacy, data security, and e-discovery considerations for customers investigating public cloud computing alternatives.

Wolters Kluwer
Law & Business

The "Certificate Authority" Trust Model for SSL: A Defective Foundation for Encrypted Web Traffic and a Legal Quagmire

By Steven B. Roosa and Stephen Schultze

A great deal of attention is given to protecting networks and private data against a long list of security threats such as viruses, worms, rootkits, malware, spyware, and social engineering attacks. Nevertheless, encrypted communications, which are used to secure every Internet transaction and confidential exchange of information imaginable, have been considered relatively safe. Recently revealed technical and operational vulnerabilities, however, highlight significant defects in the underlying trust and authentication system that uses third parties called Certificate Authorities (CAs) to vouch for the identity of Web sites to end-users (referred to as the CA Trust Model). The vulnerabilities in the CA Trust Model enable exploits in which bad actors may receive and decrypt secure communications. Attackers can cause an end-user's Internet browser to treat imposter Web sites as genuine and can execute man-in-the middle attacks in which a bad actor is able to intercept and decrypt SSL communications between the end-user and a legitimate Web site without being detected. These attacks do not require access to any username, password, or data from the end-user and can be directed at any end-user on the Internet.[1]

To make matters worse, the standard legal documents associated with the CA Trust Model purport to divest end-users of a meaningful right to rely on the authentication process. This is a problem. First, as a general matter, the end-user has not had an opportunity to click his or her assent to the terms contained in the CA's legal documents. The typical end-user is unaware of the existence of CAs and the fact that CAs provide authentication services for the secure Web sites that the end-user is accessing. Furthermore, a significant number of Web sites actively urge end-users to rely heavily on a site's supposedly "secure" SSL connection but never mention either the flaws in the CA system or the existence of standard legal documentation that seeks to undermine the end-user's rights. Thus, many site operators may say one thing ("please rely on our extremely safe and secure SSL connection") but do another (fail to advise of potential weaknesses in the system and execute contracts with the CAs that undermine the ability of the end-user to rely on the authentication process). In the face of an exploit in which an end-user's encrypted communications are unlawfully intercepted and decrypted by a bad actor and the end-user asserts legal claims against the owner of the legitimate Web site, it is a fair question whether the owner of the Web site will be able to rely on its terms and conditions of use to limit its liability.

The CA Trust Model

The CA Trust Model is supposed to allow an end-user's Web browser to reliably authenticate the identity of Web sites offering "secure" communications over the Internet.[2] The purpose of this authentication is to serve as the foundation for communications using TLS/SSL (Transport Layer Security, formerly the Secure Sockets Layer or SSL, hereinafter simply SSL), a cryptographic protocol that creates an encrypted tunnel intended to render Internet traffic indecipherable to third parties that might intercept it. Unfortunately, the degree of security provided by SSL rises and falls with the authentication system upon which it rests.

The Authentication Process of SSL

Before any encrypted traffic is passed over an SSL connection, the client and server perform a handshake in which the server offers an SSL

Steven B. Roosa is a partner in the law firm of Reed Smith LLP. He can be reached at *sroosa@reedsmith.com*. **Stephen Schultze** is the Associate Director of the Center for Information Technology Policy at Princeton University. He can be reached at *sjs@princeton.edu*.

certificate indicating that it is the true server for a given domain name. It does not matter that this information is passed unencrypted, because the certificate uses public key cryptography in which this data can be disclosed publicly without allowing any snooping parties to actually use the data to impersonate the certificate owner. Critically, the SSL certificate is also digitally signed by a third party, typically a CA. The Web browser software verifies that this signature corresponds to a trusted CA in its local database (described later) and then uses the SSL certificate to establish an encrypted connection. If any part of this authentication process is compromised, the end-user could be establishing a seemingly trusted and encrypted connection with a fraudulent third party.

The Parties That Comprise the CA Trust Model

The CA Trust Model assumes four basic parties:

1. CAs;
2. Subscribers (generally consisting of Web site operators that have purchased SSL certificates from the CA);
3. Relying parties or end-users; and
4. A participating browser.[3]

In the CA Trust Model, the CA issues an SSL certificate to a Web site operator for a given domain name. Separately, the CA also provides its own CA certificate to the developers of major Internet browsers. When accessing an SSL-secured domain name, the end-user's browser automatically uses these two types of certificates to authenticate the identity of a Web site offering an SSL connection. It does so by cryptographically proving that the CA issued and signed the SSL certificate in question, thereby confirming that the CA has vouched that the certificate was issued only to the actual controller of this domain name.[4] An end-user, or "relying party," can be anyone from a consumer sitting at his or her laptop at home to a professional sitting behind a virtual private network at his or her place of employment.

With very limited exceptions, the CA Trust Model is not currently regulated by the US government or the states.[5] It is instead built upon various standards formulated by, among others, the American Bar Association (ABA), the American Institute of Certified Public Accountants (AICPA), the Canadian Institute of Chartered Accountants (CICA), the American National Standards Institute (ANSI), the International Telecommunications Union (ITU), and the Internet Engineering Task Force (IETF).[6] The CA Trust Model is also built, in part, on the independent policies and practices of the companies that sponsor the major Internet browsers.[7]

Who Are the CAs?

In most cases, would-be CAs can be physically located anywhere in the world. They may or may not be affiliated with a governmental entity. A would-be CA often begins by paying an accounting firm to evaluate it using the WebTrust Program for Certification Authorities or some similar private standard.[8] The purpose of the evaluation is to determine if the CA will meet industry standards regarding the CA Trust Model.[9]

The next step for the CA is to approach the sponsors of the major Internet browsers—for example, Microsoft, Apple, Mozilla—and ask each company to include its CA certificate in the browser's (or operating system's or online repository's) store of trusted CAs that identifies for that particular browser that the CA is a trusted root or a Root CA.[10]

Once a CA is able to convince one or more of the browser companies to include its CA certificate(s) in its default store, the CA will then offer SSL certificates for sale to Web sites/subscribers for the purpose of carrying out the third-party authentication process under the CA Trust Model.

The Authentication Process

If a Web site provides a valid SSL certificate from a CA that the end-user's browser recognizes as a Root CA, then the browser will automatically establish a secure connection with the Web site.[11] Currently, there are reportedly 264 CAs trusted by Microsoft; 166 CAs trusted by Apple, and 144 CAs trusted by Mozilla.[12] Microsoft, Apple, and Mozilla each have different standards for determining which CAs will be trusted by their respective browsers and which will not.[13]

Fundamental Weaknesses in the CA Trust Model

CAs Are Numerous, Unfamiliar, and Able to Authenticate Any Domain Name

There are multiple institutional and technical defects in the CA Trust Model that make it inherently vulnerable. Foremost among the institutional weaknesses is the inherent ability of any CA to issue an unauthorized (and presumably illegal), yet technically valid, SSL Certificate authenticating any domain name. In other words, *any* CA can issue a digital certificate vouching that the subscriber controls *any* domain name, accurate or not.[14] For example, a Root CA that is *not* the CA for ABC Bank (in other words, ABC Bank did not purchase its SSL certificates from that CA) can nevertheless issue an SSL certificate to a third-party bad actor that will incorrectly represent that the owner of the certificate is also the owner of ABC Bank's domain name. When an end-user's browser encounters this bogus SSL certificate, it will authenticate a secure connection as if nothing were wrong.

A bad actor using such a certificate can insert itself as a man-in-the-middle in the SSL communications between an end-user and a legitimate Web site. This allows the attacker to decrypt all communications and to alter communications on the fly. As of 2009, at least one commercial vendor was allegedly producing a turnkey intercept solution offering the ability to engage in active man-in-the-middle attacks using wrongfully issued SSL certificates. The device was reportedly marketed as "an attack against the underlying . . . cryptographic key agreement protocol. . . ."[15] Although purportedly only marketed by the vendor to law enforcement, the technology behind the device is not considered to be cutting edge, and such technical capabilities have long been within the reach of criminals, untrustworthy network intermediaries, or ill-intentioned governmental regimes.

The recently articulated compelled-certificate-creation-attack leverages the power of CAs under the CA Trust Model to authenticate any domain name. This vulnerability is based on the fact that a large number of CAs are either private entities existing under the laws of jurisdictions *other than* the United States or the European Union or are governmentally owned or affiliated with countries that one might not trust even for trivial matters, let alone for security-imperative matters such as authenticating the Web site of a bank.[16] It would seem reasonable that some of these far-flung Root CAs could be compelled by state actors to issue a technically valid certificate for a given domain in order to facilitate surveillance or industrial espionage.[17]

Although these particular weaknesses are new, or at least newly discussed, this is not the first time that the CA Trust Model has shown itself to be vulnerable on a systemic basis. The fact that any Root CA can successfully vouch for the identity of any imposter Web site on the Internet has been a problem before.[18] Indeed, prior bugs and defects were troubling for that very reason: Flaws could be broadly exploited against a wide array of end-users and Web sites.[19] For example, in 2002, one browser company used a faulty library of digital certificates that did not distinguish between trusted CA certificates and mere SSL certificates issued by a trusted CA. As a result, an attacker could purchase an SSL certificate for *nastyattacker.com* and use it to sign a certificate for *amazon.com*.[20]

On another occasion in 2008, a Web researcher unaffiliated with Mozilla (the sponsor of the Firefox browser) suspected that a particular CA was performing little or no confirmation of a subscriber's identity or the actual ownership of a given domain name prior to issuing SSL certificates. When the researcher requested that the CA issue him an SSL certificate for *mozilla.com*, the CA reportedly issued the certificate with "no questions asked, no verification checks done, no control validation, no subscriber agreement presented, nothing."[21]

On yet another occasion in 2008, other researchers were successful in creating a rogue CA certificate trusted by all common Web browsers. Because the certificate appeared to be signed by a commonly trusted CA, it allowed the researchers to "impersonate any Web site on the Internet, including banking and e-commerce sites secured with [SSL]."[22]

In each case, an isolated flaw had a potentially broad impact across the Internet because of the inherent ability of the CA to vouch for ownership of any domain. This weakness cannot be removed without changing the CA Trust Model. There is no way under the model for a Web site to limit which CAs have the power to authenticate its identity to end-users. Rather, browsers will "automatically accept any identity certificate issued by any of the trusted CAs."[23] As one security expert at Princeton University recently noted, "[i]t should be abundantly clear . . . that the current model for certifying the identity of Web sites is deeply flawed."[24]

Undisclosed Delegation by Root CAs of Their Certification Authority

Another problem with the CA Trust Model is that some Root CAs delegate their SSL certificate issuing authority to certain other, unrelated CAs. These unrelated CAs are often not designated as a trusted root by the browser but will nevertheless be trusted. They are typically not designated as explicitly trusted either because the browser has not approved the unrelated CA for inclusion in the default store or the end-user is particularly tech savvy and has elected to un-trust the CA in their browser. In either case, the ultimate result of this particular cross-certification by the Root CA is that the end-user's browser will treat these certificates as if they had been issued by the Root CA even though they were not.

Cross-certification is not rogue activity by Root CAs; rather, it is a specific arrangement mentioned in the WebTrust literature as the "hybrid" model for "sharing trust."[25] However, it dilutes the control that the major browser companies have over their own default stores of trusted CAs and that end-users have over their customizations to that store. It also injects confusion as to who is being trusted and for what, especially because such arrangements are often not required to be publicly disclosed.[26]

Problems with the Legal Documentation Associated with the CA Trust Model

The institutional and technical weaknesses of the CA Trust Model are compounded by problems with the legal documentation typically associated with the Model. CAs often employ at least four standard documents:

1. The certification practice statement (CPS);

2. The certificate policy (CP);

3. The subscriber agreement; and

4. The relying party agreement.

The subscriber agreement is a contract between the CA and the owner/operator of the domain name. It sets forth the terms and conditions governing the CA's issuance of SSL certificates to the Web site operator and the operator's subsequent permitted use of those certificates.

The CPS is a separate document that describes the business practices regarding the CA's issuance of digital certificates. The CPS purports to limit the CA's monetary liability and limit the extent to which a subscriber or relying party (in the latter case, an end-user with a browser) may rely on authentication or encryption methods that use the CA's certificates. The terms of the CPS are typically incorporated by reference in the subscriber agreement, but there appears to be no mechanism by which they are presented to the relying party for his or her approval or assent. According to typical language included in the CPS, one might readily conclude that CAs do not stand behind the digital certificates that they issue:

> **Warranties and Limitations on Warranties**
>
> * * *
>
> In no event does [the CA] . . . make any representations, or provide any warranties . . . to any . . . Subscribers, Relying Parties, or any other persons, entities, or organizations with respect to . . . the reliability of any cryptographic techniques or methods used in conducting *any act*, transaction, or process involving or utilizing [a] Certificate . . . .[27]

The third document, the relying party agreement, purports to be an agreement between the CA and the relying party/end-user. This document often purports to place onerous technical obligations on the end-user, such as being familiar with the underlying cryptographic protocols and making independent judgments about the trustworthiness of any given digital certificate. A typical relying party agreement also contains a significant liability disclaimer (by the CAs) to end-users for defects in authentication.[28]

The end-user's assent to these standard documents is generally neither obtained nor sought. There appears to be no occasion when an end-user clicks his or her assent to the relying party agreement, the CPS, or the subscriber agreement.[29] As far as the end-user is concerned, these documents do not exist.

The absence of assent by the end-user places the Web site operator that is a "subscriber" to the CA's SSL certificates in a difficult position, as Web site operators are actively encouraging end-users to rely heavily on SSL encrypted communications, while entering into contracts with CAs that seek to minimize, if not eliminate, the end-user's right to rely on the authentication processes on which SSL communications depend. A review of the published decisional law fails to reveal any court decision that speaks directly to the issue of end-user rights relative to the legal documentation associated with the CA Trust Model. As a result, the legal architecture on which the model rests is untested.

Potential Exposure

Companies that do business with consumers or clients over their Web sites, and are caught in this conflicted position, may find that their own terms and conditions of use and privacy policy are either outright misleading or materially deficient because they fail to adequately apprize end-users of the potential technical and legal issues associated with the CA Trust Model. It is therefore important for any company that communicates with its customers or clients using SSL to review closely the legal documents associated with the purchase of SSL certificates from a CA (including the subscriber agreement, the CPS, and the relying party agreement) to determine if that documentation purports to limit the rights of end-users. If this proves to be the case, then the company should consider either demanding alternative contract documents from the CA or simply amending its Web site terms and conditions of use to provide end-users some degree of notice regarding the potential technical and institutional weaknesses inherent in the CA Trust Model, as well as notifying end-users of the existence of CA-related legal documentation that may seek to limit the end-user's rights.

Conclusion

Institutional and technical weaknesses are inherent in the CA Trust Model. Compounding these shortcomings is a flawed and untested legal architecture that may not afford CAs or Web sites the legal protections they anticipate.

Notes

1. Such attacks do, however, require the attacker to first have the ability to intercept and manipulate the victim's network traffic. This is a risk scenario that SSL was designed to mitigate. Entities such as network providers, DNS servers, and governments all typically have this type of access. Other entities may also gain such access via technical exploits.
2. *See* Soghoian, Christopher, and Sid Stamm. "Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL" (2010) *http://files.cloudprivacy.net/ssl-mitm.pdf*.
   Gibson, Steve. "Subverted SSL," *Security Now*, Apr. 8, 2010, *http://twit.tv/sn243*.
   Prof. Ed Felten. "Mozilla Debates Whether to Trust Chinese CA" (Princeton University, Center For Information Technology Policy, Feb. 16, 2010), *http://www.freedom-to-tinker.com/blog/felten?page=1*. Schultze, Stephen. "Web Security Trust Models" (CITP Feb. 22, 2010), *http://www.freedom-to-tinker.com/blog/sjs*.
3. WebTrust Program for Certification Authorities, American Institute of Certified Public Accountants (AICPA) and Canadian Institute of Chartered Accountants (CICA) (2000).
4. *Id.* at p.15.
5. One exception would be the FDA regulations governing electronic records and digital signatures. *See* 21 C.F.R. part 11. The FDA regulations, however, are narrow and do not apply to the vast majority of SSL connections encountered by the end-user. Furthermore, compliance with the FDA regulations is not believed to protect against the discussed defects. As for the states, there was a movement, more than a decade ago, to urge states to enact legislation governing the conduct of CAs. That movement was unsuccessful. In 2006, the lead state in the effort, Utah, repealed its CA law, known as the Utah Digital Signature Act of 1995. *See* Utah Code § 46-3-101.
6. *See* Draft PKI Assessment Guidelines (American Bar Association 2001); WebTrust Program for Certification Authorities (AICPA, CICA 2000) at p.13 n.2; ANSI X9.79:2001, Financial Services Public Key Infrastructure (PKI) Policy and Practices Framework. The current proposed standards by the IETF parallel the ITU proposed standard known as X.509. The IETF proposed standards are set forth in various requests for comment (RFCs) posted on the IETF's Web site. *See* RFC 5280 (IETF 2008).
7. Soghoian and Stamm, *supra* n.2.

8. *Id.*

9. Examples of private CA's are private companies such as Verisign, Entrust, and DigiCert. Other CA's are either run by governmental authorities or are closely affiliated with them, such as, for example, the China Internet Network Information Center and the Hongkong Post e-cert.

10. There are other CAs that are non-root, such as "subordinate" or "intermediate" CAs.

11. Soghoian and Stamm, *supra* n.2.

12. Sogohian and Stamm, *supra* n.2, at 3. Compliance with WebTrust is not mandatory but may be, as a practical matter, a condition precedent for private CA's to be trusted by the major browsers.

13. Sogohian and Stamm, *supra* n.2; Gibson, *supra* n.2.

14. *See* Felten, *supra* n.2.

15. *See* Soghoian and Stamm, *supra* n.2, at 7.

16. *Id.* at 1, 4-6.

17. *Id.*

18. *See* Felten, *supra* n.2.

19. *See* Ferguson, Niels and Bruce Schneier, Cryptography Engineering: Design Principles and Practical Applications (2010) at 278.

20. *Id.*

21. *https://blog.startcom.org.*

22. Sotirov, Alexander and Stevens, Marc. "Creating a Rogue CA Certificate" (2009) *http://www.win.tue.nl/hashclash/rogue-ca/.* Also published in the Proceedings of the 29th Annual International Cryptology Conference on Advances in Cryptology, Berlin, Heidelberg (Springer-Verlag 2009).

23. *See* Felten, *supra* n.2.

24. *See id.*

25. WebTrust Program for Certification Authorities (AICPA, CICA 2000) at 20-21.

26. Ferguson and Schneier, *supra* n.19, at 278.

27. *See* Entrust "Certification Practice Statement," *http://www.entrust.net/CPS/pdf/SSL-CPS-English-160810-v2-4.pdf* at § 2.2.1.1 (emphasis added) (authentication would appear to fall into the category of "any act").

28. *See, e.g.,* Entrust "SSL Web Server Certificate Relying Party Agreement," *http://www.entrust.net/relying/pdf/webrelying010103.pdf and Entrust "Certification Practice Statement," http://www.entrust.net/CPS/pdf/SSL-CPS-English-160810-v2-4.pdf.*

29. The Entrust SSL Web Server Certificate Relying Party Agreement, by its terms, reflects the fact that clicked assent is not required. It states that a party can become bound by either "clicking the 'accept' icon"—and no such icon appears on the exemplar Agreement posted on the Web site—or merely "by using an Entrust SSL Web Server Certificate, any information in the Entrust Repository, or any other services provided in respect to Entrust SSL Web Server Certificates *or the validation of digital signatures.*" *http://www.entrust.net/relying/pdf/webrelying010103.pdf* at 1 (emphasis added).