WIKIPEDIA

Certificate authority

In cryptography, a **certificate authority** or **certification authority** (**CA**) is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows others (relying parties) to rely upon signatures or on assertions made about the private key that corresponds to the certified public key. A CA acts as a trusted third party—trusted both by the subject (owner) of the certificate and by the party relying upon the certificate. The format of these certificates is specified by the X.509 standard.

One particularly common use for certificate authorities is to sign certificates used in HTTPS, the secure browsing protocol for the World Wide Web. Another common use is in issuing identity cards by national governments for use in electronically signing documents.

Contents

Overview

Trusted certificates can be used to create secure connections to a server via the Internet. A certificate is essential in order to circumvent a malicious party which happens to be on the route to a target server which acts as if it were the target. Such a scenario is commonly referred to as a man-in-the-middle attack. The client uses the CA certificate to authenticate the CA signature on the server certificate, as part of the authorizations before launching a secure connection. Usually, client software—for example, browsers—include a set of trusted CA certificates. This makes sense, as many users need to trust their client software. A malicious or compromised client can skip any security check and still fool its users into believing otherwise.

The clients of a CA are server supervisors who call for a certificate that their servers will bestow to users. Commercial CAs charge money to issue certificates, and their customers anticipate the CA's certificate to be contained within the majority of web browsers, so that safe connections to the certified servers work efficiently out-of-the-box. The quantity of internet browsers, other devices and applications which trust a particular certificate authority is referred to as ubiquity. Mozilla, which is a non-profit business, issues several commercial CA certificates with its products.[1] While Mozilla developed their own policy, the CA/Browser Forum developed similar guidelines for CA trust. A single CA certificate may be shared among multiple CAs or their resellers. A *root* CA certificate may be the base to issue multiple *intermediate* CA certificates with varying validation requirements.

In addition to commercial CAs, some non-profits issue digital certificates to the public without charge; notable examples are CAcert and Let's Encrypt.

Large organizations or government bodies may have their own PKIs (public key infrastructure), each containing their own CAs. Any site using self-signed certificates acts as its own CA.

Browsers and other clients of sorts characteristically allow users to add or do away with CA certificates at will. While server certificates regularly last for a relatively short period, CA certificates are further extended,[2] so, for repeatedly visited servers, it is less error-prone importing and trusting the CA issued, rather than confirm a security exemption each time the server's certificate is renewed.

Less often, trustworthy certificates are used for encrypting or signing messages. CAs dispense end-user certificates too, which can be used with S/MIME. However, encryption entails the receiver's public key and, since authors and receivers of encrypted messages, apparently, know one another, the usefulness of a trusted third party remains confined to the signature verification of messages sent to public mailing lists.

Providers

As of December 2019, the CA/Browser Forum includes the following Certificate Authority members:[3]

- Actalis S.p.A. (http://www.actalis.it)
- Amazon Trust Services LLC (https://www.amazontrust.com)
- ANF Autoridad de Certificación (https://anf.es)
- AS Sertifitseerimiskeskus (http://www.sk.ee)
- Buypass AS (http://www.buypass.no)
- Camerfirma (http://www.camerfirma.com)
- Certinomis (https://www.certinomis.fr)
- CERTIGNA (http://www.certigna.com)
- certSIGN (http://certsign.ro)
- Certum (http://www.certum.eu)
- China Financial Certification Authority (http://www.cfca.com.cn)
- Chunghwa Telecom Co., Ltd. (http://eca.hinet.net)
- China Internet Network Information Center (http://www1.cnnic.cn/IS/fwqzs)
- ComSign Ltd (https://www.comsign.co.il)
- D-TRUST GmbH (http://www.d-trust.net)
- Dark Matter (https://pki.darkmatter.ae)
- DigiCert, Inc. (https://www.digicert.com)
- Digidentity (http://www.digidentity.eu)
- Disig, a.s. (http://www.disig.sk)
- DocuSign (https://www.opentrustdtm.com) (formerly OpenTrust/KEYNECTIS)
- E-TUGRA Inc. (http://www.e-tugra.com.tr)
- eMudhra Technologies Limited (http://www.emsign.com)

- Entrust (http://www.entrust.com)
- ESG de Electronische Signatuur B.V. (https://www.de-electronische-signatuur.nl)
- Firmaprofesional (http://www.firmaprofesional.com)
- Global Digital Cybersecurity Authority Co., Ltd (https://www.gdca.com.cn)
- GlobalSign (http://www.globalsign.com)
- GoDaddy Inc (http://www.godaddy.com)
- Hellenic Academic and Research Institutions Certification Authority (HARICA) (http://www.harica.gr)
- Izenpe S.A. (http://www.izenpe.com)
- Kamu Sertifikasyon Merkezi (http://www.kamusm.gov.tr)
- KPN Corporate Market BV (http://www.kpn.com)
- kvcc root certificate authority (https://www.kvcc.me.edu/)
- Let's Encrypt (https://letsencrypt.org)
- Logius PKIoverheid (http://www.logius.nl/english)
- National Center for Digital Certification (http://www.ncdc.gov.sa)
- Network Solutions, LLC (http://www.networksolutions.com/SSL-certificates/index.jsp)
- Open Access Technology International (http://www.oati.com)
- Prvni certifikacni autorita, a.s. (http://www.ica.cz)
- QuoVadis Ltd. (http://www.quovadisglobal.com)
- Secom Trust Systems (http://www.secomtrust.net)
- SecureTrust (https://www.securetrust.com)
- Sectigo
- Shanghai Electronic Certification Authority Center Co. Ltd (http://www.sheca.com)
- Skaitmeninio sertifikavimo centras (SSC) (http://www.ssc.lt)
- SSL.com (https://www.ssl.com)
- Swisscom (Switzerland) Ltd (http://www.swisscom.ch)
- SwissSign AG (http://www.swisssign.com)
- TAIWAN-CA Inc. (https://www.twca.com.tw/Portal/Portal.aspx)
- TrustCor Systems, S. de R.L. (https://www.trustcorsystems.com)
- TURKTRUST (http://www.turktrust.com.tr)
- Visa (http://www.visa.com)
- Wells Fargo (http://www.wellsfargo.com)

Worldwide, the certificate authority business is fragmented, with national or regional providers dominating their home market. This is because many uses of digital certificates, such as for legally binding digital signatures, are linked to local law, regulations, and accreditation schemes for certificate authorities.

However, the market for globally trusted TLS/SSL server certificates is largely held by a small number of multinational companies. This market has significant barriers to entry due to the technical requirements.[4] While not legally required, new providers may choose to undergo annual security audits (such as WebTrust[5] for certificate authorities in North America and ETSI in Europe[6]) to be included as a trusted root by a web browser or operating system. More than 180 root certificates are trusted in the Mozilla Firefox web browser, representing approximately eighty organizations.[7] Over 200 root certificates are trusted by macOS. As of Android 4.2 (Jelly Bean), Android currently contains over 100 CAs that are updated with each release.[8]

On November 18, 2014, a group of companies and nonprofit organizations, including the Electronic Frontier Foundation, Mozilla, Cisco, and Akamai, announced Let's Encrypt, a nonprofit certificate authority that provides free domain validated X.509 certificates as well as software to enable installation and maintenance of certificates.[9] Let's Encrypt is operated by the newly formed Internet Security Research Group, a California nonprofit recognized as tax-exempt under Section 501(c)(3).[10]

According to NetCraft in May 2015, the industry standard for monitoring active TLS certificates, states that "Although the global [TLS] ecosystem is competitive, it is dominated by a handful of major CAs — three certificate authorities (Symantec, Comodo, GoDaddy) account for three-quarters of all issued [TLS] certificates on public-facing web servers. The top spot has been held by Symantec (or VeriSign before it was purchased by Symantec) ever since [our] survey began, with it currently accounting for just under a third of all certificates. To illustrate the effect of differing methodologies, amongst the million busiest sites Symantec issued 44% of the valid, trusted certificates in use — significantly more than its overall market share."[11]

A W3Techs survey from May 2015 shows:[12][13]

Rank	Issuer	Usage	Market share
1	Comodo	6.1%	41.0%
2	Symantec	5%	30.2%
3	GoDaddy	2.2%	13.3%
4	GlobalSign	1.7%	10.4%
5	DigiCert	0.5%	3.1%
6	StartCom	0.4%	2.2%
7	Entrust	0.1%	0.8%
8	Verizon	0.1%	0.7%
9	Trustwave	0.1%	0.6%
10	Secom	0.1%	0.6%
11	Unizeto	0.1%	0.4%
12	Buypass	0.1%	0.1%
13	QuoVadis	< 0.1%	0.1%
14	Deutsche Telekom	< 0.1%	0.1%
15	Network Solutions	< 0.1%	0.1%
16	SwissSign	< 0.1%	0.1%

A W3Techs survey from November 2017 shows:[14]

Rank	Issuer	Usage	Market share
1	Comodo	16.7%	38.4%
2	IdenTrust	13.9%	32.0%
3	Symantec	5.6%	12.9%
4	GoDaddy	3.3%	7.5%
5	GlobalSign	1.9%	4.5%
6	DigiCert	1.0%	2.2%
7	Certum	0.3%	0.7%
8	Entrust	0.2%	0.4%
9	Secom	0.1%	0.3%
10	Actalis	0.1%	0.3%
11	Trustwave	0.1%	0.2%
12	Let's Encrypt	0.1%	0.2%
13	StartCom	0.1%	0.2%
14	WISeKey Group	< 0.1%	0.1%

A W3Techs survey from May 2018 shows that IdenTrust, a cross-signer of Let's Encrypt intermediates,[15] has risen to be the most popular SSL certificate authority, while Symantec has dropped out of the chart, due to its security services being acquired by DigiCert:[16][17]

Rank	Issuer	Usage	Market share
1	IdenTrust	20.4%	39.7%
2	Comodo	17.9%	34.9%
3	DigiCert	6.3%	12.3%
4	GoDaddy	3.7%	7.2%
5	GlobalSign	1.8%	3.5%
6	Certum	0.4%	0.7%
7	Actalis	0.2%	0.3%
8	Entrust	0.2%	0.3%
9	Secom	0.1%	0.3%
10	Let's Encrypt	0.1%	0.2%
11	Trustwave	0.1%	0.1%
12	WISeKey Group	< 0.1%	0.1%
13	StartCom	< 0.1%	0.1%
14	Network Solutions	< 0.1%	0.1%

Validation standards

The commercial CAs that issue the bulk of certificates for HTTPS servers typically use a technique called "domain validation" to authenticate the recipient of the certificate. The techniques used for domain validation vary between CAs, but in general domain validation techniques are meant to prove that the certificate applicant controls a given domain name, not any information about the applicant's identity.

Many Certificate Authorities also offer Extended Validation (EV) certificates as a more rigorous alternative to domain validated certificates. Extended validation is intended to verify not only control of a domain name, but additional identity information to be included in the certificate. Some browsers display this additional identity information in a green box in the URL bar. One limitation of EV as a solution to the weaknesses of domain validation is that attackers could still obtain a domain validated certificate for the victim domain, and deploy it during an attack; if that occurred, the difference observable to the victim user would be the absence of a green bar with the company name. There is some question as to whether users would be likely to recognise this absence as indicative of an attack being in progress: a test using Internet Explorer 7 in 2009 showed that the absence of IE7's EV warnings were not noticed by users, however Microsoft's current browser, Edge, shows a significantly greater difference between EV and domain validated certificates, with domain validated certificates having a hollow, grey lock.

Validation weaknesses

Domain validation suffers from certain structural security limitations. In particular, it is always vulnerable to attacks that allow an adversary to observe the domain validation probes that CAs send. These can include attacks against the DNS, TCP, or BGP protocols (which lack the cryptographic protections of TLS/SSL), or the compromise of routers. Such attacks are possible either on the network near a CA, or near the victim domain itself.

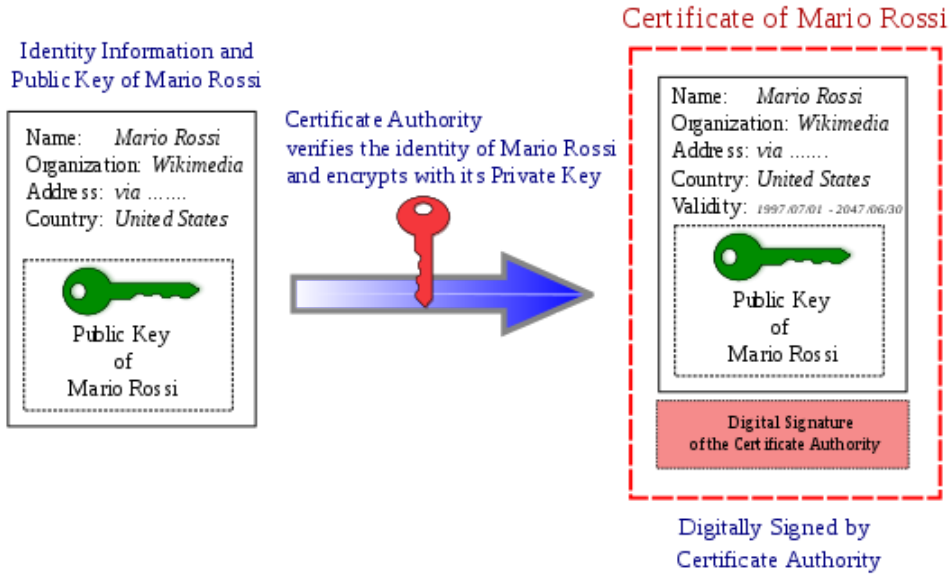
One of the most common domain validation techniques involves sending an email containing an authentication token or link to an email address that is likely to be administratively responsible for the domain. This could be the technical contact email address listed in the domain's WHOIS entry, or an administrative email like `admin@`, `administrator@`, `webmaster@`, `hostmaster@` or `postmaster@` the domain.[18][19] Some Certificate Authorities may accept confirmation using `root@`, `info@`, or `support@` in the domain.[20] The theory behind domain validation is that only the legitimate owner of a domain would be able to read emails sent to these administrative addresses.

Domain validation implementations have sometimes been a source of security vulnerabilities. In one instance, security researchers showed that attackers could obtain certificates for webmail sites because a CA was willing to use an email address like `ssladmin@domain.com` for domain.com, but not all webmail systems had reserved the "ssladmin" username to prevent attackers from registering it.[21]

Prior to 2011, there was no standard list of email addresses that could be used for domain validation, so it was not clear to email administrators which addresses needed to be reserved. The first version of the CA/Browser Forum Baseline Requirements, adopted November 2011, specified a list of such addresses. This allowed mail hosts to reserve those addresses for administrative use, though such precautions are still not universal. In January 2015, a Finnish man registered the username "hostmaster" at the Finnish version of Microsoft Live and was able to obtain a domain-validated certificate for live.fi, despite not being the owner of the domain name.[22]

Issuing a certificate

A CA issues digital certificates that contain a public key and the identity of the owner. The matching private key is not made available publicly, but kept secret by the end user who generated the key pair. The certificate is also a confirmation or validation by the CA that the public key contained in the certificate belongs to the person, organization, server or other entity noted in the certificate. A CA's obligation in such schemes is to verify an applicant's credentials, so that users and relying parties can trust the information in the CA's certificates. CAs use a variety of standards and tests to do so. In essence, the certificate authority is responsible for saying "yes, this person is who they say they are, and we, the CA, certify that".[23]



The procedure of obtaining a Public key certificate

If the user trusts the CA and can verify the CA's signature, then they can also assume that a certain public key does indeed belong to whoever is identified in the certificate.

Example

Public-key cryptography can be used to encrypt data communicated between two parties. This can typically happen when a user logs on to any site that implements the HTTP Secure protocol. In this example let us suppose that the user logs on to their bank's homepage www.bank.example to do online banking. When the user opens www.bank.example homepage, they receive a public key along with all the data that their web-browser displays. The public key could be used to encrypt data from the client to the server but the safe procedure is to use it in a protocol that determines a temporary shared symmetric encryption key; messages in such a key exchange protocol can be enciphered with the bank's public key in such a way that only the bank server has the private key to read them.

The rest of the communication then proceeds using the new (disposable) symmetric key, so when the user enters some information to the bank's page and submits the page (sends the information back to the bank) then the data the user has entered to the page will be encrypted by their web browser. Therefore, even if someone can access the (encrypted) data that was communicated from the user to www.bank.example, such eavesdropper cannot read or decipher it.

This mechanism is only safe if the user can be sure that it is the bank that they see in their web browser. If the user types in www.bank.example, but their communication is hijacked and a fake website (that pretends to be the bank website) sends the page information back to the user's browser, the fake web-page can send a fake public key to the user (for which the fake site owns a matching private key). The user will fill the form with their personal data and will submit the page. The fake web-page will then get access to the user's data.

This is what the certificate authority mechanism is intended to prevent. A certificate authority (CA) is an organization that stores public keys and their owners, and every party in a communication trusts this organization (and knows its public key). When the user's web browser receives the public key from www.bank.example it also receives a digital signature of the key (with some more information, in a so-called X.509 certificate). The browser already possesses the public key of the CA and consequently can verify the signature, trust the certificate and the public key in it: since www.bank.example uses a public key that the certification authority certifies, a fake www.bank.example can only use the same public key. Since the fake www.bank.example does not know the corresponding private key, it cannot create the signature needed to verify its authenticity.

Security

It is difficult to assure correctness of match between data and entity when the data are presented to the CA (perhaps over an electronic network), and when the credentials of the person/company/program asking for a certificate are likewise presented. This is why commercial CAs often use a combination of authentication techniques including leveraging government bureaus, the payment infrastructure, third parties' databases and services, and custom heuristics. In some enterprise systems, local forms of authentication such as Kerberos can be used to obtain a certificate which can in turn be used by external relying parties. Notaries are required in some cases to personally know the party whose signature is being notarized; this is a higher standard than is reached by many CAs. According to the American Bar Association outline on Online Transaction Management the primary points of US Federal and State statutes enacted regarding digital signatures has been to "prevent conflicting and overly burdensome local regulation and to establish that electronic writings satisfy the traditional requirements associated with paper documents." Further the US E-Sign statute and the suggested UETA code[24] help ensure that:

1. a signature, contract or other record relating to such transaction may not be denied legal effect, validity, or enforceability solely because it is in electronic form; and
2. a contract relating to such transaction may not be denied legal effect, validity or enforceability solely because an electronic signature or electronic record was used in its formation.

Despite the security measures undertaken to correctly verify the identities of people and companies, there is a risk of a single CA issuing a bogus certificate to an imposter. It is also possible to register individuals and companies with the same or very similar names, which may lead to confusion. To minimize this hazard, the *certificate transparency* initiative proposes auditing all certificates in a public unforgeable log, which could help in the prevention of phishing.[25][26]

In large-scale deployments, Alice may not be familiar with Bob's certificate authority (perhaps they each have a different CA server), so Bob's certificate may also include his CA's public key signed by a different CA_2, which is presumably recognizable by Alice. This process typically leads to a hierarchy or mesh of CAs and CA certificates.

Authority revocation lists

An *authority revocation list* (ARL) is a form of certificate revocation list (CRL) containing certificates issued to certificate authorities, contrary to CRLs which contain revoked end-entity certificates.

Industry organizations

- Certificate Authority Security Council (CASC) – In February 2013, the CASC was founded as an industry advocacy organization dedicated to addressing industry issues and educating the public on internet security. The founding members are the seven largest Certificate Authorities.[27][28]
- Common Computing Security Standards Forum (CCSF) – In 2009 the CCSF was founded to promote industry standards that protect end users. Comodo Group CEO Melih Abdulhayoğlu is considered the founder of the CCSF.[29]
- CA/Browser Forum – In 2005, a new consortium of Certificate Authorities and web browser vendors was formed to promote industry standards and baseline requirements for internet security. Comodo Group CEO Melih Abdulhayoğlu organized the first meeting and is considered the founder of the CA/Browser Forum.[30][31]

Baseline requirements

The CA/Browser Forum publishes the Baseline Requirements,[32] a list of policies and technical requirements for CAs to follow. These are a requirement for inclusion in the certificate stores of Firefox[33] and Safari.[34]

CA compromise

If the CA can be subverted, then the security of the entire system is lost, potentially subverting all the entities that trust the compromised CA.

For example, suppose an attacker, Eve, manages to get a CA to issue to her a certificate that claims to represent Alice. That is, the certificate would publicly state that it represents Alice, and might include other information about Alice. Some of the information about Alice, such as her employer name, might be true, increasing the certificate's credibility. Eve, however, would have the all-important private key associated with the certificate. Eve could then use the certificate to send digitally signed email to Bob, tricking Bob into believing that the email was from Alice. Bob might even respond with encrypted email, believing that it could only be read by Alice, when Eve is actually able to decrypt it using the private key.

A notable case of CA subversion like this occurred in 2001, when the certificate authority VeriSign issued two certificates to a person claiming to represent Microsoft. The certificates have the name "Microsoft Corporation", so they could be used to spoof someone into believing that updates to Microsoft software came from Microsoft when they actually did not. The fraud was detected in early 2001. Microsoft and VeriSign took steps to limit the impact of the problem.[35][36]

In 2011 fraudulent certificates were obtained from Comodo and DigiNotar,[37][38] allegedly by Iranian hackers. There is evidence that the fraudulent DigiNotar certificates were used in a man-in-the-middle attack in Iran.[39]

In 2012, it became known that Trustwave issued a subordinate root certificate that was used for transparent traffic management (man-in-the-middle) which effectively permitted an enterprise to sniff SSL internal network traffic using the subordinate certificate.[40]

Key storage

An attacker who steals a certificate authority's private keys is able to forge certificates as if they were CA, without needed ongoing access to the CA's systems. Key theft is therefore one of the main risks certificate authorities defend against. Publicly trusted CAs almost always store their keys on a hardware security module (HSM), which allows them to sign certificates with a key, but generally prevent extraction of that key with both physical and software controls. CAs typically take the further precaution of keeping the key for their long-term root certificates in an HSM that is kept offline, except when it is needed to sign shorter-lived intermediate certificates. The intermediate certificates, stored in an online HSM, can do the day-to-day work of signing end-entity certificates and keeping revocation information up to date.

CAs sometimes use a key ceremony when generating signing keys, in order to ensure that the keys are not tampered with or copied.

Implementation weakness of the trusted third party scheme

The critical weakness in the way that the current X.509 scheme is implemented is that any CA trusted by a particular party can then issue certificates for any domain they choose. Such certificates will be accepted as valid by the trusting party whether they are legitimate and authorized or not.[41] This is a serious shortcoming given that the most commonly encountered technology employing X.509 and trusted third parties is the https protocol. As all major web browsers are distributed to their end-users pre-configured with a list of trusted CAs that numbers in the dozens this means that any one of these pre-approved trusted CAs can issue a valid certificate for any domain whatsoever.[42] The industry response to this has been muted.[43] Given that the contents of a browser's pre-configured trusted CA list is determined independently by the party that is distributing or causing to be installed the browser application there is really nothing that the CAs themselves can do.

This issue is the driving impetus behind the development of the DNS-based Authentication of Named Entities (DANE) protocol. If adopted in conjunction with Domain Name System Security Extensions (DNSSEC) DANE will greatly reduce if not completely eliminate the role of trusted third parties in a domain's PKI.

Software

Various software is available to operate a certificate authority. Generally such software is required to sign certificates, maintain revocation information, and operate OCSP or CRL services. Some examples are:

- DogTag[44]
- EJBCA
- gnoMint
- OpenCA
- OpenSSL, an SSL/TLS library that comes with tools allowing its use as a simple certificate authority
- EasyRSA, OpenVPN's command line CA utilities using OpenSSL.
- r509[45]
- TinyCA, which is a perl gui on top of some CPAN modules.
- XCA[46]
- XiPKI,[47] CA and OCSP responder, with support of SHA3, EdDSA and SM2.

- Boulder is an automated server that uses the Automated Certificate Management Environment[48] (ACME) protocol.
- Windows Server contains a CA as part of Certificate Services for the creation of digital certificates. In Windows Server 2008 and later the CA may be installed as part of Active Directory Certificate Services.
- OpenXPKI

See also

- SAFE-BioPharma Association - an example of a non-HTTPS CA.
- Validation Authority
- Contact page
- People for Internet Responsibility
- Web of trust

References

1. "Mozilla Included CA Certificate List — Mozilla" (https://www.mozilla.org/projects/security/certs/included/index.html). Mozilla.org. Archived (https://web.archive.org/web/20130804123413/http://www.mozilla.org/projects/security/certs/included/index.html) from the original on 2013-08-04. Retrieved 2014-06-11.
2. Zakir Durumeric; James Kasten; Michael Bailey; J. Alex Halderman (12 September 2013). "Analysis of the HTTPS Certificate Ecosystem" (http://conferences.sigcomm.org/imc/2013/papers/imc257-durumericAemb.pdf) (PDF). *The Internet Measurement Conference*. SIGCOMM. Archived (http://archive.wikiwix.com/cache/20131222062343/http://conferences.sigcomm.org/imc/2013/papers/imc257-durumericAemb.pdf) (PDF) from the original on 22 December 2013. Retrieved 20 December 2013.
3. "Members of the CA / Browser Forum" (https://cabforum.org/members/). CA / Browser Forum. Retrieved 2019-12-03.
4. "What is SSL Certificate?" (https://www.instantssl.com/ssl-certificate.html). Archived (https://web.archive.org/web/20151103095720/https://www.instantssl.com/ssl-certificate.html) from the original on 2015-11-03. Retrieved 2015-10-16.
5. "webtrust" (http://www.webtrust.org/). webtrust. Archived (https://web.archive.org/web/20130818182356/http://www.webtrust.org/) from the original on 2013-08-18. Retrieved 2013-03-02.
6. Kirk Hall (April 2013). "Standards and Industry Regulations Applicable to Certification Authorities" (https://casecurity.org/wp-content/uploads/2013/04/Standards-and-Industry-Regulations-Applicable-to-Certification-Authorities.pdf) (PDF). Trend Micro. Archived (https://web.archive.org/web/20160304074157/https://casecurity.org/wp-content/uploads/2013/04/Standards-and-Industry-Regulations-Applicable-to-Certification-Authorities.pdf) (PDF) from the original on 2016-03-04. Retrieved 2014-06-11.
7. "CA:IncludedCAs - MozillaWiki" (https://wiki.mozilla.org/CA:IncludedCAs). *wiki.mozilla.org*. Archived (https://web.archive.org/web/20170325024230/https://wiki.mozilla.org/CA:IncludedCAs) from the original on 2017-03-25. Retrieved 2017-03-18.
8. "Security with HTTPS and SSL" (https://developer.android.com/training/articles/security-ssl.html#ClientCert). *developer.android.com*. Archived (https://web.archive.org/web/20170708031534/https://developer.android.com/training/articles/security-ssl.html#ClientCert) from the original on 2017-07-08. Retrieved 2017-06-09.
9. "Let's Encrypt: Delivering SSL/TLS Everywhere" (https://letsencrypt.org/2014/11/18/announcing-lets-encrypt.html). Let's Encrypt. Archived (https://web.archive.org/web/20141118170934/https://letsencrypt.org//2014/11/18/announcing-lets-encrypt.html) from the original on 2014-11-18. Retrieved 2014-11-20.
10. "About" (https://letsencrypt.org/about/). Let's Encrypt. Archived (https://web.archive.org/web/20150610222600/https://letsencrypt.org/about/) from the original on 2015-06-10. Retrieved 2015-06-07.
11. "Counting SSL certificates - Netcraft" (http://news.netcraft.com/archives/2015/05/13/counting-ssl-certificates.html). *news.netcraft.com*. Archived (https://web.archive.org/web/20150516035536/http://news.netcraft.com/archives/2015/05/13/counting-ssl-certificates.html) from the original on 2015-05-16.
12. "Usage of SSL certificate authorities for websites" (http://w3techs.com/technologies/overview/ssl_certificate/all). 2015-05-13. Retrieved 2015-09-29.
13. "Comodo has become the most widely used SSL certificate authority" (http://w3techs.com/blog/entry/comodo_has_become_the_most_widely_used_ssl_certificate_authority). *w3techs.com*.

14. "Usage of SSL certificate authorities for websites" (http://w3techs.com/technologies/overview/ssl_certificate/all). 2017-11-15. Retrieved 2017-11-15.
15. "Let's Encrypt - Chain of Trust" (https://letsencrypt.org/certificates/). *Let's Encrypt*. Retrieved 2018-06-08. "... [Let's Encrypt's] intermediate is ... cross-signed by another certificate authority, IdenTrust, whose root is already trusted in all major browsers."
16. "DigiCert Closes Acquisition of Symantec's Website Security Business" (https://www.websecurity.symantec.com/en/us/digicert-and-symantec-faq). *Symantec*. October 31, 2017. Retrieved 2018-06-08.
17. "Usage of SSL certificate authorities for websites" (https://w3techs.com/technologies/overview/ssl_certificate/all). 2018-05-28. Retrieved 2018-06-08.
18. "Archived copy" (https://cabforum.org/wp-content/uploads/BRv1.2.3.pdf) (PDF). Archived (https://web.archive.org/web/20150323072323/https://cabforum.org/wp-content/uploads/BRv1.2.3.pdf) (PDF) from the original on 2015-03-23. Retrieved 2015-03-20.
19. "CA/Forbidden or Problematic Practices - MozillaWiki" (https://wiki.mozilla.org/CA/Forbidden_or_Problematic_Practices#Non-Standard_Email_Address_Prefixes_for_Domain_Ownership_Validation). *wiki.mozilla.org*. Archived (https://web.archive.org/web/20170721104255/https://wiki.mozilla.org/CA/Forbidden_or_Problematic_Practices#Non-Standard_Email_Address_Prefixes_for_Domain_Ownership_Validation) from the original on 2017-07-21. Retrieved 2017-07-06.
20. "SSL FAQ - Frequently Asked Questions - Rapid SSL" (https://www.rapidssl.com/learn-ssl/ssl-faq/). *www.rapidssl.com*. Archived (https://web.archive.org/web/20150206224655/http://www.rapidssl.com/learn-ssl/ssl-faq/) from the original on 2015-02-06.
21. Zusman, Mike (2009). *Criminal charges are not pursued: Hacking PKI* (https://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-zusman-hacking_pki.pdf) (PDF). DEF CON 17. Las Vegas. Archived (https://web.archive.org/web/20130415102243/http://defcon.org/images/defcon-17/dc-17-presentations/defcon-17-zusman-hacking_pki.pdf) (PDF) from the original on 2013-04-15.
22. "A Finnish man created this simple email account - and received Microsoft's security certificate" (http://www.tivi.fi/Kaikki_uutiset/2015-03-18/A-Finnish-man-created-this-simple-email-account---and-received-Microsofts-security-certificate-3217662.html). *tivi.fi*. Archived (https://web.archive.org/web/20150808204526/http://www.tivi.fi/Kaikki_uutiset/2015-03-18/A-Finnish-man-created-this-simple-email-account---and-received-Microsofts-security-certificate-3217662.html) from the original on 2015-08-08.
23. "Responsibilities of Certificate Authority" (https://www.instantssl.com/code-signing/code-signing-technical.html). Archived (https://web.archive.org/web/20150212120554/https://www.instantssl.com/code-signing/code-signing-technical.html) from the original on 2015-02-12. Retrieved 2015-02-12.
24. "Electronic Signatures and Records" (http://euro.ecom.cmu.edu/program/law/08-732/Transactions/ElectronicSignatures.pdf) (PDF). Archived (https://web.archive.org/web/20160304001604/http://euro.ecom.cmu.edu/program/law/08-732/Transactions/ElectronicSignatures.pdf) (PDF) from the original on 2016-03-04. Retrieved 2014-08-28.
25. "Certificate transparency" (http://www.certificate-transparency.org). Archived (https://web.archive.org/web/20131101042429/http://www.certificate-transparency.org/) from the original on 2013-11-01. Retrieved 2013-11-03.
26. "Certificate transparency" (http://tools.ietf.org/html/rfc6962). Internet Engineering Task Force. Archived (https://web.archive.org/web/20131122070410/http://tools.ietf.org/html/rfc6962) from the original on 2013-11-22. Retrieved 2013-11-03.
27. "Multivendor power council formed to address digital certificate issues" (https://web.archive.org/web/20130728114851/http://www.networkworld.com/news/2013/021413-council-digital-certificate-266728.html). *Network World*. February 14, 2013. Archived from the original (http://www.networkworld.com/news/2013/021413-council-digital-certificate-266728.html) on July 28, 2013.
28. "Major Certificate Authorities Unite In The Name Of SSL Security" (https://archive.is/20130410174711/http://www.darkreading.com/authentication/167901072/security/news/240148546/major-certificate-authorities-unite-in-the-name-of-ssl-security.html). *Dark Reading*. February 14, 2013. Archived from the original (http://www.darkreading.com/authentication/167901072/security/news/240148546/major-certificate-authorities-unite-in-the-name-of-ssl-security.html) on April 10, 2013.
29. "CA/Browser Forum Founder" (http://www.melih.com/about/). Archived (https://web.archive.org/web/20140823090043/http://www.melih.com/about/) from the original on 2014-08-23. Retrieved 2014-08-23.
30. "CA/Browser Forum" (https://www.cabforum.org/). Archived (https://web.archive.org/web/20130512014830/https://www.cabforum.org/) from the original on 2013-05-12. Retrieved 2013-04-23.

31. Wilson, Wilson. "CA/Browser Forum History" (http://docbox.etsi.org/workshop/2012/201201_CA_DAY/5_Wilson_CAB-Forum.pdf) (PDF). DigiCert. Archived (https://web.archive.org/web/20130512052041/http://docbox.etsi.org/workshop/2012/201201_CA_DAY/5_Wilson_CAB-Forum.pdf) (PDF) from the original on 2013-05-12. Retrieved 2013-04-23.
32. "Baseline Requirements" (https://cabforum.org/baseline-requirements-documents/). CAB Forum. Archived (https://web.archive.org/web/20140107191853/https://cabforum.org/baseline-requirements-documents/) from the original on 7 January 2014. Retrieved 14 April 2017.
33. "Mozilla Root Store Policy" (https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy/#conformance). Mozilla. Archived (https://web.archive.org/web/20170415013337/https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy/#conformance) from the original on 15 April 2017. Retrieved 14 April 2017.
34. "Apple Root Certificate Program" (https://www.apple.com/certificateauthority/ca_program.html). Apple. Archived (https://web.archive.org/web/20170320054143/https://www.apple.com/certificateauthority/ca_program.html) from the original on 20 March 2017. Retrieved 14 April 2017.
35. "CA-2001-04" (https://www.cert.org/advisories/CA-2001-04.html). Cert.org. Archived (https://web.archive.org/web/20131102170603/http://www.cert.org/advisories/CA-2001-04.html) from the original on 2013-11-02. Retrieved 2014-06-11.
36. Microsoft, Inc. (2007-02-21). "Microsoft Security Bulletin MS01-017: Erroneous VeriSign-Issued Digital Certificates Pose Spoofing Hazard" (http://support.microsoft.com/kb/293818). Archived (https://web.archive.org/web/20111026052552/http://support.microsoft.com/kb/293818) from the original on 2011-10-26. Retrieved 2011-11-09.
37. Bright, Peter (28 March 2011). "Independent Iranian hacker claims responsibility for Comodo hack" (https://arstechnica.com/security/news/2011/03/independent-iranian-hacker-claims-responsibility-for-comodo-hack.ars). Ars Technica. Archived (https://web.archive.org/web/20110829214335/http://arstechnica.com/security/news/2011/03/independent-iranian-hacker-claims-responsibility-for-comodo-hack.ars) from the original on 29 August 2011. Retrieved 2011-09-01.
38. Bright, Peter (2011-08-30). "Another fraudulent certificate raises the same old questions about certificate authorities" (https://arstechnica.com/security/news/2011/08/earlier-this-year-an-iranian.ars). Ars Technica. Archived (https://web.archive.org/web/20110912164822/http://arstechnica.com/security/news/2011/08/earlier-this-year-an-iranian.ars) from the original on 2011-09-12. Retrieved 2011-09-01.
39. Leyden, John (2011-09-06). "Inside 'Operation Black Tulip': DigiNotar hack analysed" (https://www.theregister.co.uk/2011/09/06/diginotar_audit_damning_fail/). *The Register*. Archived (https://web.archive.org/web/20170703005353/https://www.theregister.co.uk/2011/09/06/diginotar_audit_damning_fail/) from the original on 2017-07-03.
40. "Trustwave issued a man-in-the-middle certificate" (http://www.h-online.com/security/news/item/Trustwave-issued-a-man-in-the-middle-certificate-1429982.html). *The H Security*. 2012-02-07. Archived (https://web.archive.org/web/20120313085319/http://www.h-online.com/security/news/item/Trustwave-issued-a-man-in-the-middle-certificate-1429982.html) from the original on 2012-03-13. Retrieved 2012-03-14.
41. Osborne, Charlie. "Symantec sacks staff for issuing unauthorized Google certificates - ZDNet" (https://www.zdnet.com/article/symantec-sacks-staff-for-issuing-unauthorized-google-certificates/). *zdnet.com*. Archived (https://web.archive.org/web/20161002045808/http://www.zdnet.com/article/symantec-sacks-staff-for-issuing-unauthorized-google-certificates/) from the original on 2016-10-02.
42. "Unauthorized Google Digital Certificates Discovered" (https://www.linkedin.com/pulse/20140812224351-2983013-unauthorized-google-digital-certificates-discovered). *linkedin.com*. 12 August 2014.
43. "In the Wake of Unauthorized Certificate Issuance by the Indian CA NIC, can Government CAs Still be Considered "Trusted Third Parties"?" (https://casecurity.org/2014/07/24/unauthorized-certificate-issuance/). *casecurity.org*. 24 July 2014. Archived (https://web.archive.org/web/20161003100318/https://casecurity.org/2014/07/24/unauthorized-certificate-issuance/) from the original on 3 October 2016.
44. "Dogtag Certificate System" (https://pki.fedoraproject.org/wiki/PKI_Main_Page). Pki.fedoraproject.org. Archived (https://web.archive.org/web/20130129071620/http://pki.fedoraproject.org/wiki/PKI_Main_Page) from the original on 2013-01-29. Retrieved 2013-03-02.
45. "reaperhulk/r509 · GitHub" (https://github.com/reaperhulk/r509). Github.com. Archived (https://web.archive.org/web/20131018033218/https://github.com/reaperhulk/r509) from the original on 2013-10-18. Retrieved 2013-03-02.
46. "xca.sourceforge.net" (http://xca.sourceforge.net/). xca.sourceforge.net. Archived (https://web.archive.org/web/20121203213403/http://xca.sourceforge.net/) from the original on 2012-12-03. Retrieved 2013-03-02.

47. "xipki/xipki · GitHub" (https://github.com/xipki/xipki). Github.com. Archived (https://web.archive.org/web/20170831004530/https://github.com/xipki/xipki) from the original on 2017-08-31. Retrieved 2016-10-17.
48. "letsencrypt/acme-spec" (https://github.com/letsencrypt/acme-spec). github.com. Archived (https://web.archive.org/web/20141121192641/https://github.com/letsencrypt/acme-spec) from the original on 2014-11-21. Retrieved 2014-11-20.

External links

- How secure is HTTPS today? How often is it attacked? (https://www.eff.org/deeplinks/2011/10/how-secure-https-today), *Electronic Frontier Foundation* (25 October 2011)