Certificate authority

In cryptography, a **certificate authority** or **certification authority** (**CA**) is an entity that stores, signs, and 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.[1] The format of these certificates is specified by the X.509 or EMV 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.[2]

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.[3] 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.[4] 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 publicly-trusted digital

certificates without charge, for example Let's Encrypt. Some large cloud computing and web hosting companies are also publicly-trusted CAs and issue certificates to services hosted on their infrastructure, for example IBM Cloud, Amazon Web Services, Cloudflare, and Google Cloud Platform.

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.

Commercial banks that issue EMV payment cards are governed by the EMV Certificate Authority,[5] payment schemes that route payment transactions initiated at Point of Sale Terminals (POS) to a Card Issuing Bank to transfer the funds from the card holder's bank account to the payment recipient's bank account. Each payment card presents along with its card data also the Card Issuer Certificate to the POS. The Issuer Certificate is signed by EMV CA Certificate. The POS retrieves the public key of EMV CA from its storage, validates the Issuer Certificate and authenticity of the payment card before sending the payment request to the payment scheme.

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,[6] 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

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.[7] While not legally required, new providers may choose to undergo annual security audits (such as WebTrust[8] for certificate authorities in North America and ETSI in Europe[9]) to be included as a trusted root by a web browser or operating system.

As of 24 August 2020, 147 root certificates, representing 52 organizations, are trusted in the Mozilla Firefox web browser,[10] 168 root certificates, representing 60 organizations, are trusted by macOS,[11] and 255 root certificates, representing 101 organizations, are trusted by Microsoft Windows.[12] As of Android 4.2 (Jelly Bean), Android currently contains over 100 CAs that are updated with each release.[13]

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.[14] Let's Encrypt is operated by the newly formed Internet Security Research Group, a California nonprofit recognized as federally tax-exempt.[15]

According to Netcraft in May 2015, the industry standard for monitoring active TLS certificates, "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."[16]

In 2020, according to independent survey company Netcraft, "DigiCert is the world's

largest high-assurance certificate authority, commanding 60% of the Extended Validation Certificate market, and 96% of organization-validated certificates globally.[17]

As of April 2023 the survey company W3Techs, which collects statistics on certificate authority usage among the Alexa top 10 million and the Tranco top 1 million websites, lists the six largest authorities by absolute usage share as below. [18]

Rank	Issuer	Usage	Market Share
1	IdenTrust	38.5%	43.6%
2	DigiCert Group	13.1%	14.5%
3	Sectigo (Comodo Cybersecurity)	12.1%	13.4%
4	GlobalSign	16.1%	16.7%
5	Let's Encrypt	5.8%	6.4%
6	GoDaddy Group	4.8%	5.3%

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

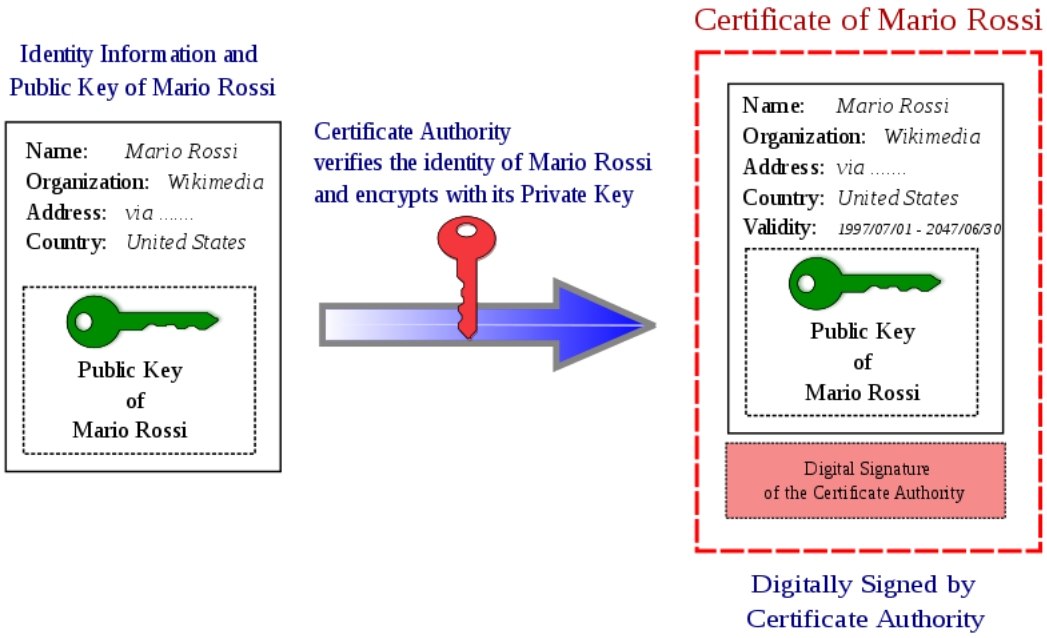
email containing an authentication token or link to an email address that is likely to be administratively responsible for the domain. This could be the technical contact email address listed in the domain's WHOIS entry, or an administrative email like `admin@`, `administrator@`, `webmaster@`, `hostmaster@` or `postmaster@` the domain.[19][20] Some Certificate Authorities may accept confirmation using `root@`, `info@`, or `support@` in the domain.[21] 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.[22]

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.[23]

Issuing a certificate



The procedure of obtaining a Public key 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 issued certificate. 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".[24]

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.[25]

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.[26]

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.[27]

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[28] 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.[29][30]

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.

Certificate revocation

A certificate may be revoked before it expires, which signals that it is no longer valid. Without revocation, an attacker would be able to exploit such a compromised or misissued certificate until expiry.[31] Hence, revocation is an important part of a public key infrastructure.[32] Revocation is performed by the issuing CA, which produces a

cryptographically authenticated statement of revocation.[33]

For distributing revocation information to clients, timeliness of the discovery of revocation (and hence the window for an attacker to exploit a compromised certificate) trades off against resource usage in querying revocation statuses and privacy concerns.[34] If revocation information is unavailable (either due to accident or an attack), clients must decide whether to *fail-hard* and treat a certificate as if it is revoked (and so degrade availability) or to *fail-soft* and treat it as

unrevoked (and allow attackers to sidestep revocation).[35]

Due to the cost of revocation checks and the availability impact from potentially-unreliable remote services, Web browsers limit the revocation checks they will perform, and will fail-soft where they do.[36] Certificate revocation lists are too bandwidth-costly for routine use, and the Online Certificate Status Protocol presents connection latency and privacy issues. Other schemes have been proposed but have not yet been successfully deployed to enable fail-hard checking.[32]

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.[37][38]
- 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.[39]

- 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.[40][41]

Baseline requirements

The CA/Browser Forum publishes the Baseline Requirements,[42] a list of policies and technical requirements for CAs to follow. These are a requirement for

inclusion in the certificate stores of Firefox[43] and Safari.[44]

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 a 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.[45][46]

In 2008, Comodo reseller Certstar sold a certificate for mozilla.com to Eddy Nigg, who had no authority to represent Mozilla.[47]

In 2011 fraudulent certificates were obtained from Comodo and

DigiNotar,[48][49] allegedly by Iranian hackers. There is evidence that the fraudulent DigiNotar certificates were used in a man-in-the-middle attack in Iran.[50]

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.[51]

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.[52] 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.[53] The industry response to this has been muted.[54] 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 eliminate the role of trusted third parties in a domain's PKI.

See also

- Validation authority
- Contact page
- People for Internet Responsibility
- Web of trust
- Chain of trust
- Digital signature
- DigiNotar certificate authority breach

- Comodo certificate authority breach

References

1. *Chien, Hung-Yu (2021-08-19). "Dynamic Public Key Certificates with Forward Secrecy" (https://doi.org/10.3390%2Felectronics10162009) . Electronics. **10** (16): 2009. doi:10.3390/electronics10162009 (https://doi.org/10.3390%2Felectronics10162009) . ISSN 2079-9292 (https://www.worldcat.org/issn/2079-9292) .*
2. *"What is a certificate authority (CA)?" (https://searchsecurity.techtarget.com/definition/certificate-authority) .*

3. *Villanueva, John Carl. "How do Digital Certificates Work - An Overview" (https://www.jscape.com/blog/an-overview-of-how-digital-certificates-work) . www.jscape.com. Retrieved 2021-09-05.*
4. *"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.*
5. *"EMV CA" (https://emvca.com/index.html#EMV-CA) . EMV Certificate Authority Worldwide. 2 October 2010. Retrieved February 17, 2019.*

6. *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.*

7. *"What is an SSL Certificate?" (https://www.instantssl.com/about-tls-ssl-certificates) . Archived (https://web.archive.org/web/20151103095720/https://www.instantssl.com/ssl-certificate.html) from the original on 2015-11-03. Retrieved 2022-03-19.*
8. *"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.*

9. *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.*

10. *"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.*
11. *"List of available trusted root certificates in macOS High Sierra" (https://support.apple.com/en-us/HT208127)* . *Apple Support. Retrieved 2020-08-24.*
12. *"Microsoft Included CA Certificate List" (https://ccadb-public.secure.force.com/microsoft/IncludedCACertificateReportForMSFT)* . *ccadb-public.secure.force.com. Retrieved 2020-08-24.*

13. *"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.*

14. *"Let's Encrypt: Delivering SSL/TLS Everywhere" (https://letsencrypt.org/2014/11/18/announcing-lets-encrypt.html) (Press release). 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.*
15. *"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.*

16. *"Counting SSL certificates - Netcraft" (http://news.netcraft.com/archives/2015/05/13/counting-ssl-certificates.html)* . *news.netcraft.com. 13 May 2015. 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.*
17. *"DigiCert - World's Largest High-Assurance Certificate Authority | Netcraft" (https://trends.netcraft.com/www.digicert.com#extended-validation)* . *trends.netcraft.com.*
18. *"Usage statistics of SSL certificate authorities for websites, April 2023 - W3Techs" (https://w3techs.com/technologies/overview/ssl_certificate)* . *w3techs.com.*

19. *"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.*

20. *"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.*

21. *"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.*
22. *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.*

23. *"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*.

24. *"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.*
25. *"Network World" (https://books.google.com/books?id=GRcEAAAAMBAJ&q=certification+authority+PKI&pg=PA55) . 17 January 2000.*

26. *Applied Cryptography and Network Security: Second International Conference, ACNS 2004, Yellow Mountain, China, June 8-11, 2004. Proceedings (https://books.google.com/books?id=GggFlaVxyVQC&dq=applied+cryptography+certificate+issuance&pg=PA281)* . *Springer. June 2004.* *ISBN 9783540222170.*
27. *The Shortcut Guide to Managing Certificate Lifecycles (https://books.google.com/books?id=BdJi5AQRzsIC&dq=digital+certificate+issuance&pg=PA40)* . *Realtimepublishers.com. 2006.* *ISBN 9781931491594.*

28. *"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.*
29. *"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.*

30. *Laurie, Ben; Langley, Adam; Kasper, Emilia (June 2013). "Certificate transparency" (http://tools.ietf.org/html/rfc6962) . Internet Engineering Task Force. doi:10.17487/RFC6962 (https://doi.org/10.17487%2FRFC6962) . Archived (https://web.archive.org/web/20131122070410/http://tools.ietf.org/html/rfc6962) from the original on 2013-11-22. Retrieved 2013-11-03.*
31. *Smith, Dickinson & Seamons 2020, p. 1.*
32. *Sheffer, Saint-Andre & Fossati 2022, 7.5. Certificate Revocation.*
33. *Chung et al. 2018, p. 3.*
34. *Smith, Dickinson & Seamons 2020, p. 10.*
35. *Larisch et al. 2017, p. 542.*

36. *Smith, Dickinson & Seamons 2020, p. 1-2.*
37. *"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.*

38. *"Major Certificate Authorities Unite In The Name Of SSL Security" (https://archive.today/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.*

39. *"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.*

40. *"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.*
41. *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.*

42. *"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*.
43. *"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*.

44. *"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.*
45. *"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.*

46. *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.*
47. *Seltzer, Larry. "SSL Certificate Vendor Sells Mozilla.com CSSL Certificate to Some Guy" (https://www.eweek.com/security/ssl-certificate-vendor-sells-mozilla.com-cert-to-some-guy/) . eWeek. Retrieved 5 December 2021.*

48. *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.*

49. *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.*

50. *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.*

51. *"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.*

52. *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.*
53. *"Unauthorized Google Digital Certificates Discovered" (https://www.linkedin.com/pulse/20140812224351-2983013-unauthorized-google-digital-certificates-discovered) . linkedin.com. 12 August 2014.*

54. *"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.*

Works cited

- Chung, Taejoong; Lok, Jay; Chandrasekaran, Balakrishnan; Choffnes, David; Levin, Dave; Maggs, Bruce M.; Mislove, Alan; Rula, John; Sullivan, Nick; Wilson, Christo (2018). "Is

the Web Ready for OCSP Must-Staple?" (https://taejoong.github.io/pubs/publications/chung-2018-ocsp.pdf) (PDF). *Proceedings of the Internet Measurement Conference 2018*. pp. 105–118. doi:10.1145/3278532.3278543 (https://doi.org/10.1145%2F3278532.3278543) . ISBN 9781450356190. S2CID 53223350 (https://api.semanticscholar.org/CorpusID:53223350) .

- Larisch, James; Choffnes, David; Levin, Dave; Maggs, Bruce M.; Mislove, Alan; Wilson, Christo (2017). "CRLite: A Scalable System for Pushing All TLS Revocations to All Browsers". *2017 IEEE*

Symposium on Security and Privacy (SP). pp. 539–556. doi:10.1109/sp.2017.17 (https://doi.org/10.1109%2Fsp.2017.17) . ISBN 978-1-5090-5533-3. S2CID 3926509 (https://api.semanticscholar.org/CorpusID:3926509) .

- Sheffer, Yaron; Saint-Andre, Pierre; Fossati, Thomas (November 2022). *Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)* (https://datatracker.ietf.org/doc/html/rfc9325) . doi:10.17487/RFC9325 (https://doi.org/10.17487%2FRFC932

5) . RFC 9325 (https://datatracker.ietf.org/doc/html/rfc9325) .

- Smith, Trevor; Dickinson, Luke; Seamons, Kent (2020). "Let's Revoke: Scalable Global Certificate Revocation". *Proceedings 2020 Network and Distributed System Security Symposium*. doi:10.14722/ndss.2020.24084 (https://doi.org/10.14722%2Fndss.2020.24084) . ISBN 978-1-891562-61-7. S2CID 211268930 (https://api.semanticscholar.org/CorpusID:211268930) .

External links

- How secure is HTTPS today? How often is it attacked? (https://www.eff.org/deep

links/2011/10/how-secure-https-today) , *Electronic Frontier Foundation* (25 October 2011)