Computers & Security 163 (2026) 104814

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Full Length Article

Certificate revocation – search for a way forward

Takahito Yoshizawa [a,*], Himanshu Agarwal [b], Dave Singelée [c], Bart Preneel [a]

[a] KU Leuven - COSIC, Kasteelpark Arenberg 10 Bus 2452, Leuven, B-3001, Belgium
[b] KU Leuven - Physics Dept., Celestijnenlaan 200d - bus 2412, Leuven, B-3001, Belgium
[c] KU Leuven - DistriNet Group-T, Andreas Vesaliusstraat 13, Leuven, B-3000, Belgium

ARTICLE INFO

Keywords:
Certificate revocation
CRL
OCSP
ACME
STAR
Short-term certificates

ABSTRACT

Revocation of digital certificates represents a series of improvements by IETF in order to standardize a complete and effective solution. This applies to the context of Internet web sites in which web servers and browsers use digital certificates to establish Transport Layer Security (TLS). Despite IETF's effort over the years to establish a reliable revocation mechanism, including Certificate Revocation List (CRL), Online Certificate Status Protocol (OCSP) and its variants, various technical issues hinder complete resolution of the revocation problem. At the same time, all major browser vendors implement their own proprietary solutions to address the revocation problem. As a result, revocation solutions are fragmented, incomplete, and ineffective, and the level of real-world acceptance of standardized solutions is limited. To address this situation, in 2020, IETF has introduced *short-term certificate* concept to avoid revocation altogether. It is called Support for Short-Term, Automatically Renewed (STAR) which recommends a validity period of 4 days. To measure the level of adoption of this new approach in the Internet, we collected and analyzed web server certificates from 1 million websites; the result of our extensive analysis indicates that this scheme has not gained traction in reality. In fact, we found no implementation of a 4-day validity period out of more than 1.5 million server certificates that we collected. This situation indicates that the latest IETF effort to promote short-term certificates has not materialized, with no clear alternative solution in sight to resolve the revocation issue. We present our insights into the reasons for this absence of traction in reality and present our view of a possible way forward.

1. Introduction

The use of the Internet has grown over the years as ubiquitous connectivity, exemplified by technological advances such as wireless systems and battery-powered handheld devices, has become the norm in our daily lives. Various types of IoT devices that rely on Internet access also contribute to this growing trend. A statistic (Statista.com, 2025) shows that 5.52 billion people (~67.5% of the world population) use the Internet as of October 2024. By 2025, the daily volume of global data generated is expected to surpass 163 zettabytes (163×10^{21}) bytes (Berisha et al., 2022).

Increasing use of the Internet also means that its security and privacy aspects continue to be paramount. In recent years, the use of encryption in web browsing (i.e., HTTPS, or Transport Layer Security (TLS)) has become a norm. The TLS protocol uses a digital certificate to establish a session key between the web server and the browser for the subsequent encryption of website traffic. Using the *server certificate*, browsers can verify the authenticity of its owner, i.e., the website, and establish a session key.

An implication of the use of digital certificates is the need to revoke them when necessary, e.g., compromise of the private key.[1] The lack of a reliable revocation mechanism leads to security vulnerabilities, as it blinds browsers from the outdated validity status of web server certificates. As certificates bind the owner's identity to a public key, lack of reliable revocation means that *certificate users* cannot discern whether this binding still holds or not. Consequently, the meaning of certificates without this assurance reduces to a mere fact that this certificate was valid at the time it was issued and nothing more.

Historically, multiple solutions have been designed to revoke certificates, including Certificate Revocation Lists (CRL), Online Certificate Revocation Protocol (OCSP) (Myers et al., 1999) and its enhancements and extensions (Pettersen, 2013). However, these mechanisms also introduced problems. Some of the most representative problems include: (1) the absence of guaranteed freshness of the revocation information

[1] The complete list of the revocation reason is defined in ITU-T X.509 spec, ISO/IEC 9594-8:2020 (E) (ISO, 2019).

* Corresponding author.
E-mail address: takahito.yoshizawa@esat.kuleuven.be (T. Yoshizawa).

https://doi.org/10.1016/j.cose.2025.104814
Received 24 March 2025; Received in revised form 17 November 2025; Accepted 25 December 2025
Available online 3 January 2026
0167-4048/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

and its availability, (2) additional delay in TLS session establishment, and (3) implications on potential privacy compromise. These problems have prevented the truly effective revocation from being realized, resulting in an unresolved open problem.

To address this situation, IETF has shifted its strategy for certificate revocation, namely to eliminate its needs altogether by promoting short-term certificates (Sheffer et al., 2020). Given its short-term nature, certificates that expire within a few days do not require proactive revocation. This passive approach suffices to remove the need for a viable revocation solution.

This begs the question: has this short-term certificate scheme been adopted in the Internet at large? If so, to what extent? To answer this question, we have conducted a data collection campaign to capture TLS server certificates from one million websites and analyzed them. Contrary to our expectations, our analysis reveals that there is no adoption of the short-term certificate scheme in reality. This leads us to the next question: (1) why does the latest RFC fail to catch on?, and more importantly, (2) is there a viable alternative approach? In this paper, we address these questions.

Our contributions in this paper are: (1) a detailed analysis of TLS server certificates to show the level of traction of short-term certificates, (2) a discussion of why the series of attempts by IETF to solve the certificate revocation problem has not been completed thus far, and (3) an exploration of a potential way forward to this situation.

The rest of the document is organized as follows. We first cover the background, including the related standards, related work, and browser implementation trends in Section 2. In Section 3, we then describe our experimental data collection of TLS server certificates, and discuss our analysis and findings in Section 4. In Section 5, we analyze the reasons why short-term certificate has not gained traction. Based on our observations, we explore possible way forward to address the current situation to close the gap. Finally, Section 6 concludes our discussion.

2. Background: standards, related work, and browser implementations

2.1. Historical overview of standardized certificate revocation schemes

The history of certificate revocation and its evolution is as old as the PKI itself (Gutmann, 2002). Table 1 summarizes an overview of historical revocation schemes with their respective pros and cons. We describe the revocation mechanisms in each of these protocols in the following sections as shown in Fig. 1.

Table 1
Overview of standardized revocation schemes

Scheme	RFC / Year	Advantages	Disadvantages
CRL	RFC2459 (Housley et al., 1999) 1980*s*	- Simplicity	- Not scalable since the file grows linearly to MB in size. - Timeliness and availability of the latest revocation status not guaranteed.
OCSP	RFC2560 (Myers et al., 1999) 1999 RFC6960 (Santesson et al., 2013) 2013	- More granular and scalable than CRL.	- Privacy leak of users' browsing behaviour. - Dependency on the availability and reliability of the OCSP responder. - Soft failure nullifies OCSP check.
OCSP Stapling	RFC6066 (Eastlake, 2011) 2011	- No privacy leak of users' browsing behavior. - OCSP query/response is a non-blocking event. - Saves one OCSP query/response round trip between browser and OCSP responder.	- Dependency on the availability and reliability of the OCSP responder. - Soft failure nullifies OCSP check.
OCSP Must Staple	RFC7633 (Hallam-Baker, 2015) 2015	- Enforces hard-fail upon failure to verify the OCSP response along with certificate.	- Negligible level of real-world implementations.
Short-term certs.	RFC8739 (Sheffer et al., 2020) 2020	- Eliminates the need of active revocation.	- Additional processing burden on the CA and web host providers.

2.1.1. Certificate revocation list (CRL)

The original mechanism to revoke certificates was the Certificate Revocation List (CRL). Its mechanism is shown in Fig. 1(a). Certificate Authorities (CAs) periodically generate CRLs and distribute them to *certificate users* (step (1)). When a *certificate user*, e.g., an Internet browser, sends a *Client Hello* message to initiate the TLS establishment procedure with a website, the latter provides a server certificate to the former in its *Server Hello* message (step (2)). Upon receiving this server certificate, the browser checks the serial number of the certificate in question against the obtained CRLs (step (3)); if the certificate's serial number is listed in the CRL, then the browsers reject it; otherwise, the browser proceeds with the TLS session establishment.

The Certificate Management Protocol (CMP), as specified in RFC 4210 (Mononen et al., 2005), provides a wide range of certificate management functions, including revocation by generating CRLs (Crone, 2020). However, this CRL approach has several issues. Its lack of scalability is the most prominent one; its size is known to grow as large as

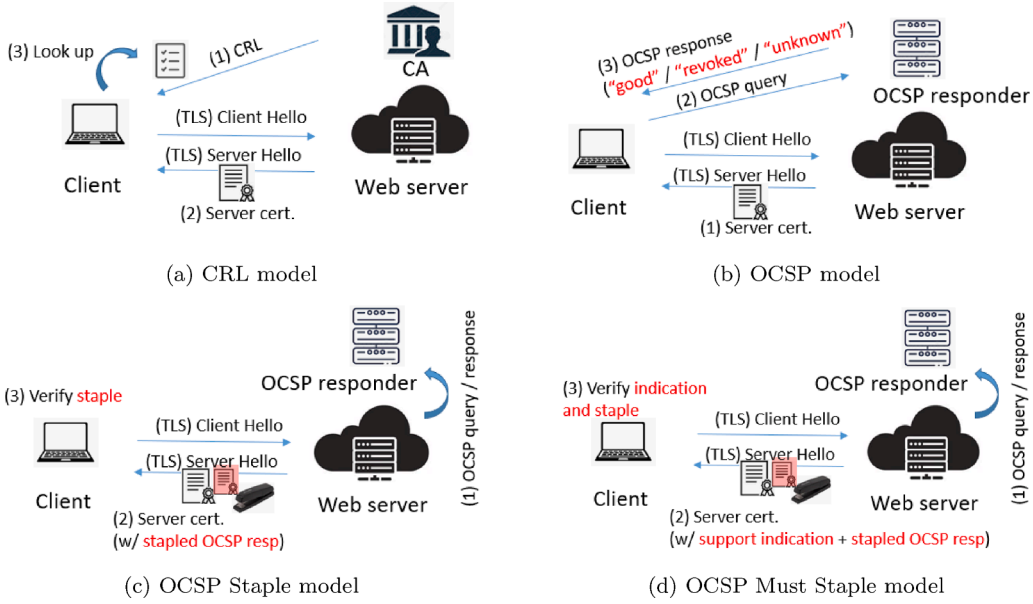


Fig. 1. Revocation mechanisms.

200 Mbytes (Gutmann, 2024). It quickly adds up as CAs of multiple PKI systems generate and issue their own CRL at the same time.[2]

Other issues associated with CRL include timely generation, availability, and retrieval of the latest CRL by certificate users; if the fresh CRL is not guaranteed to be made available for *certificate users* under all circumstances, the CRL's legitimacy becomes questionable. In other words, there is always a relative time difference between its generation by the CA and its retrieval by a user; if a certificate is revoked after the last CRL generation, this condition is not reflected to the CRL until the next CRL is generated. In other words, the absence of revocation information of a given certificate does not guarantee that it has not been revoked. For example, if a CRL is generated once every 7 days, there is a maximum of ~7-day worth of discrepancy between the CRL content and the actual revocation status of the certificates. In other words, the CRL's *freshness* is the key to CRL's effectiveness.

2.1.2. Online certificate status protocol (OCSP)

To address these CRL issues, in 1999, IETF created Online Certificate Status Protocol (OCSP) (Myers et al., 1999). Its mechanism is shown in Fig. 1(b). This protocol allows *certificate users* to query the revocation status of a certificate to a CA server called *OCSP Responder* on an individual certificate basis; thus this is a much more granular and scalable approach than CRL.

As a first step, when a browser initiates the TLS session establishment procedure with a website (*Client Hello* message), the latter provides a server certificate to the former in its *Server Hello* message (step (1)). This event triggers the browser to send an OCSP query message to the *OCSP Responder* to check the validity of the received certificate (step (2)). In response to this query, the *OCSP Responder* sends an OCSP response message that contains either *"good"*, *"revoked"*, or *"unknown"* (step (3)). If the received response is *"good"*, the browser proceeds with the TLS establishment; otherwise, it rejects the certificate and abort the procedure.

Similar to CRL, OCSP also introduced its own issues. First, certificate checking with the OCSP server is a blocking event that delays the TLS session setup. Furthermore, less-than-perfect availability and reliability of the *OCSP Responder* result in the OCSP check timing out and the procedure failing. In fact, it is indicated that the OCSP request often fails (Wiki, 2025). This event negatively impacts user experience.[3] To avoid a negative user experience, browsers adopt a *soft failure* in which browsers ignore this failure condition by assuming that the certificate is valid, and render the website content on the browser. Second, the browser's action to check the certificate's validity results in a voluntary exposure of the user's browsing history, the user's interests, and habits. Additional metadata, such as the requesting IP address and its location, can potentially compromise the user's privacy.

2.1.3. OCSP stapling

As an enhancement, IETF introduced *OCSP Stapling* (Eastlake, 2011). Fig. 1(c) shows its mechanism. In this scheme, web servers, instead of browsers, periodically execute the OCSP protocol message exchange with the *OCSP Responder*, fetch the certificate validity status, and store it (step (1)). When a client's browser sends a TLS *Client Hello* message to a website, the latter provides the signed result to the former along with the server certificate in its *Server Hello* message. Thus, this OCSP status information is transmitted *in-band* within the TLS handshake procedure (step (2)). The word *"stapling"* is a metaphorical expression to express the web server's action to attach the OCSP response information to the server certificate. Upon receiving this stapled message from the web server, the client's browser verifies the validity of the OCSP response by checking its signature signed by the *OCSP Responder* (step (3)). Similar to the original *OCSP* protocol, the browser proceeds with the TLS establishment if the response is *"good"*, and rejects the certificate and aborts the TLS establishment otherwise.

This OCSP status query/response between the web server and the CA is expected to be repeated every 4 days to refresh the certificate validity status at a reasonable level (Chung et al., 2018). This enhancement improves three issues in the original OCSP protocol (Myers et al., 1999). First, it addresses the users' privacy issue in OCSP as browsers no longer contact the *OCSP Responder* by itself, thus not exposing its browsing behavior. Second, the OCSP status query/response message exchange is no longer a blocking event for the TLS session setup, as the web server spontaneously attaches the revocation status of its certificate. Third, it also reduces the TLS session setup latency.

[2] One notable incident that resulted in a large number of certificates to be revoked by many CAs was triggered by the *Heartbleed* vulnerability in 2014 (Durumeric et al., 2014).

[3] Discussion on revocation support in Firefox hints that the browser implementations *are* indeed concerned with users' negative perception due to the delay introduced by revocation checking, such as OCSP (Wiki, 2025).

The *soft failure* option is an issue neither *OCSP* nor *OCSP Stapling* has solved up to this point. It is a case where *blacklist*-based mechanism in CRL and less-than-perfect availability of *OCSP Responder* causes ambiguity. In fact, this option nullifies the whole purpose of the OCSP protocol by *assuming* that the absence of a OCSP response means the certificate is *not* revoked, i.e., effectively treating *no-response-due-to-timeout* event as a positive outcome.

2.1.4. OCSP must staple

To enforce the integrity of the OCSP protocol, IETF introduced another enhancement, called OCSP *Must Staple* extension (Hallam-Baker, 2015). This mechanism is shown in Fig. 1(d). In this extension, web servers can indicate their support of this *Must Staple* extension in the certificate. As the first step, similar to OCSP *Stapling*, the web server periodically executes the OCSP protocol with the *OCSP Responder* and stores the response (step (1)). When it receives a TLS *Client Hello* message from a browser, the web server attaches the signed OCSP response, along with the server certificate itself (with an OCSP *Must Staple* support indication), in the TLS *Server Hello* message (step (2)). Upon receiving this message, the browser checks whether this option is indicated (step (3)). If this option is indicated and a valid OCSP response is present, then the browser completes the TLS session establishment. If it is *not* present, browsers must reject the received certificate and terminate the setup of the TLS session.

Despite the robustification of the OCSP *Stapling*, past studies on this subject indicate that the adoption of *Must Staple* extension in websites is negligible. Studies published in 2018 indicate that despite the OCSP support in websites being at 95.4 % since its introduction in 1999, the adoption of OCSP *Stapling* and *Must Staple* options remained at marginal levels. Specifically, support of *Must Staple* was only at 0.02% since its introduction in 2015 (Chung et al., 2018; Dickinson et al., 2019). Furthermore, the availability and reliability of the *OCSP Responder* still remain as a determining factor for the accuracy and freshness of the revocation status. In addition, OCSP *Stapling* holds web servers accountable for the "burden of proof"; thus, they have to carry the extra burden of checking the revocation status with the CA.

2.2. Related work

There are a number of studies that analyze the revocation of TLS certificates. We provide brief reviews of relevant studies from various perspectives on revocation.

2.2.1. Revocation analysis

Liu et al. (2015) analyzed empirical data of how revocation is supported in PKI, CAs, web servers, and browser clients. They described that more recent solutions (e.g., OCSP over CRL) improve the revocation situation, but even the latest solution does not offer a perfect solution. They also clarified that browsers' proprietary solutions (e.g., *CRLSet*, *OneCRL*) cover only a small subset of revoked certificates.

2.2.2. OCSP support analysis

Chung et al. (2018) analyzed a large number of TLS server certificates from the perspective of OCSP protocol support and made a number of notable observations. On the positive side, they observed that a significant number (95.4%) of certificates support OCSP at the time of its publication (2018). However, the support of OCSP *Must Staple* was almost non-existent at 0.02%. Furthermore, their extensive analysis of real-world data demonstrated that *OCSP Responders* have reliability issues for multiple reasons. Given that the OCSP protocol during the TLS session establishment is a blocking event, the detailed description of these failure conditions proved that the OCSP support can negatively impact user experience, and subsequently led browser vendors to implement a *soft failure* option to circumvent these failure situations.

2.2.3. Certificate delegation

Chuat et al. (2020) described the issues of certificate delegation, which has become popular due to the increasing use of Content Delivery Networks (CDN). They pointed out that CDN is fundamentally at odds with TLS, as the CDN concept is equivalent to the Man-in-the-Middle (MitM) attack situation, which TLS intends to avoid. From this perspective, they investigated proxy certificates and delegated credentials in CDN context, and explored their applicability without sharing a private key with delegated entities.

Similarly, in the CDN context, Boureanu et al. (2020) proposed a new scheme to support delegate credentials without sharing a private key with the delegated entity. Although this solution appears viable, its applicability is limited to organizations, e.g., companies, rather than individual end users.

2.2.4. New approach to revocation checking

In the past years, there have been several proposals to improve the revocation checking over the standardized mechanisms. We describe their key concepts and characteristics in this section; Table 2 captures the summary.

Larisch et al. (2017) proposed a scalable revocation scheme called *CRLite*. It is based on multiple Bloom filters organized in a cascading fashion. This data structure, periodically updated and downloaded to browsers, represents the revocation information of all valid TLS certificates in the world. It is space-efficient and fast to check the revocation status. Its primary characteristic is that it avoids both false positives and false negatives, which is the fundamental limitation of Bloom filters. The downside of this scheme is that collecting and maintaining the revocation status of all websites' certificates and generating the filters takes significant resources. This resource usage is exacerbated as this filter contains both revoked and non-revoked certificates issued in the world. Mozila stated that they adopted this *CRLite* scheme in Firefox as of early 2019 (Wiki, 2025).

Dickinson et al. (2019) proposed a new revocation strategy called *Certificate Revocation Table (CRT)*. This scheme, which is inspired by OS memory management, operates in organizations (companies); it maintains a list of certificates that are recently used in the organization, called *Working Set*, and periodically checks their revocation status. They conducted a 60-day experiment and obtained that 99.86% of the TLS sessions established had certificate revocation status cached in advance. One downside of this scheme is that its effectiveness is questionable in non-organizational situations, such as individual end users.

A subsequent paper by the same authors, Smith et al. (2020), presented a new revocation strategy. In this scheme, each certificate holds a unique identifier which is used as an index to a dynamically-sized bit vector, called *Certificate Revocation Vector (CRV)*, that contains the revocation status of certificates. The revocation status search is done using a key called *Revocation Number (RN)*, which is a sequential number within the CRV. The use of a sequential number makes the search efficient compared to a certificate's *Serial Number (SN)* or hash fingerprint. Their analysis indicated that this bit vector approach results in more efficient revocation checking than *CRLite*.

2.2.5. Short-term certificates

Bruhner et al. (2022) conducted a detailed analysis of TLS server certificates collected over the period of 2017 and 2020, mainly from the perspective of certificate reuse (CR). They proposed a short-term certificate solution based on a *parent-child* relationship between the longer-term *parent* and shorter-term *children* certificates. In this solution, only the *parent* certificate is logged in the Certificate Transparency (CT) log. In this way, it keeps the CA overhead low and limits the CT log size from increasing significantly.

Ma et al. (2023) evaluated the impact of reducing the validity period from 398 days to 90 days. They found that 90-day certificates reduce stale certificates by 75–86% as the shorter certificate lifetime increases

Table 2
Overview of revocation check schemes in literature.

Ref.	Year	Description	Advantages	Disadvantages
(Larisch et al., 2017)	2017	CRLLite: a Bloom filter-based revocation checking scheme.	Efficient and fast; No false-positive and false-negative result.	Significant resources are required to maintain the filter from all *issued* certificates world wide.
(Dickinson et al., 2019)	2019	CRT: a cashing scheme based on OS memory management, intended for organizations.	High cashing hit rate in organizational environment.	Questionable to work outside organizations, e.g., individual end users.
(Smith et al., 2020)	2020	CRV: a bit-vector-based revocation checking scheme.	The use of sequential number makes search more efficient than CRLlite.	Significant resources are required to maintain the bit vector from all *revoked* certificates world wide.

Table 3
Browsers' revocation check support.

Ref.	Chrome (Chromium.org, 2025)	Safari (Basile and Wood, 2017)	Edge (Microsoft, 2025)	Firefox (Wiki, 2025)	Opera (SLL.com, 2021)
Market Share	65.2%	18.57%	5.24%	2.74%	2.15%
CRL Support	no	no	no	no	no
OCSP Support	no	(see *1)	no	yes (*2)	no
OCSP Timeout	(n/a)	unknown	n/a	(see *3)	n/a
At OCSP Timeout	(n/a)	unknown	n/a	soft-failure (*4)	n/a
Int. CA Cert. Check	CRLSet (*5)	(see *6)	CRLSet (*5)	OneCRL	CRLSet (*5)
EE Cert. Check	no	(see *6)	no	OCSP	no

*1: Safari runs OCSP for certificates that are indicated as revoked to confirm it; otherwise, no OCSP.
*2: OCSP is used to check EE certificates only.
*3: Timeout value: 2 sec. (EE cert.), 10 sec (EV cert.).
*4: Configurable with the default = soft-failure, optional hard-failure.
*5: Chromium-based browsers support the same proprietary mechanism.
*6: Apple's CA file is publicly available but cannot be parsed due to the tool no longer available.

agility and narrows the window of opportunity for adversaries to compromise the website or private key compared to conventional longer validity periods.

2.3. Browser implementations on revocation checking

While standardized revocation schemes have evolved as discussed in the previous section, web browsers have adopted their own solutions to the revocation problem.[4] Table 3 captures the implementations of revocation schemes in the most notable web browsers. It is in decreasing order of market share as of November 2024 (Statista.com, 2008).

2.3.1. Proprietary solutions and their scope

Google Chrome (*CRLSet* (Chromium.org, 2025)), and Mozilla Firefox (*OneCRL* (Wiki, 2025)) both implement proprietary solutions that are limited to checking of CA certificates.[5] Because the revocation of CA certificates is a rather rare event and the number of CAs is limited compared to the number of end-entities (EEs), periodic checks of CA certificates are a viable solution within the tolerable level of scalability. As a result, voluntary check of the revocation status of EE certificates is outside the scope of these proprietary solutions; this also means that they cover only a small subset of certificates (Liu et al., 2015). They hold a predetermined set of CAs for which they periodically fetch the latest CRLs from their parent CAs and maintain their revocation status. This check is performed at an interval in the order of once a month (Wiki, 2025). Then they periodically push the updated revocation status to browsers as part of the software update.

2.3.2. Revocation check of EE certificates

Chrome or Chromium-based browsers (*CRLSet*) do not check the revocation status of EE certificates at all. On the other hand, Firefox (*OneCRL*) supports OCSP for EE certificates with different rules depending on the certificate types and extensions. For Extended Validation (EV) and non-EV types (Organization Validation (OV) and Domain Validation (DV)) certificates, the timeout values for an OCSP query response are 10 seconds and 2 seconds, respectively. In both cases, if Firefox does not receive a response from the *OCSP Responder*, it triggers the *soft failure* as default behavior. Firefox also considers that all certificates with a validity period less than 10 days are *short-lived* certificates; for these certificates, Firefox does not trigger a revocation check. Firefox accepts stapled certificates and processes the *must-staple* indication by checking the stapled OCSP response content. For handling OCSP response timeout, OCSP stapling, and *must-staple* extension, Firefox provides user configuration to change its revocation check behavior (Wiki, 2025).

Edge and Opera are two of the Chromium-based browsers.[6] Therefore, they also use *CRLSet* to check the revocation status of CAs and reflect it as part of the software update; it does not check the revocation status of EE certificates using OCSP.

Safari uses a similar but slightly different approach. Apple periodically checks revocation status from selected CAs, packages the collected information, and downloads it to the browser as part of the software update. When the URL is in this *revoked certificate* list, Safari triggers OCSP to double-check if the certificate is indeed revoked or not, unless the web host provides a stapled OCSP response during the TLS handshake. In other words, if the URL in question is not in this list, then Safari does not trigger the revocation check.[7] Apple claims that their solution reduces privacy concerns in OCSP, automatically updates the revocation information without a performance penalty to set up the connection to websites (Basile and Wood, 2017).

[4] Although there exist numerous browsers (Wikipedia.org, 2025), the five most popular browsers (Chrome, Firefox, Safari, Edge, and Opera) take almost all market share (Statista.com, 2008). Thus, we limit our discussion to these browsers only.

[5] In this context, the term CA collectively refers to both the Root CA (RCA) and Intermediate CA (ICA).

[6] Edge migrated from a proprietary to Chromium-based implementation. In its earlier implementation, the use of OCSP was configurable, which is no longer the case (SLL.com, 2021)

[7] Their downloadable CA list is publicly available (12.7 Mbytes of file size); however, its file format is undocumented and their tool to parse this file is no longer available (Apple, 2018), thus it is not possible to identify which CAs are listed.

Revocation check behaviors of other browsers in different OSes are discussed in Chung et al. (2018). It is worth noting that not a single mobile device's (e.g., smartphones, tablets) browser implementation supports revocation check in any form Liu et al. (2015), Bruhner et al. (2022).

In summary, revocation support in web browsers is highly fragmented – multiple proprietary solutions covering only CA certificates, uneven support of revocation status of EE certificates, and varying levels of support in different platforms (e.g., desktop and mobile devices). This situation is far from ideal; at the same time, this current situation also succinctly reflects the long but unsuccessful series of attempts in the history of defining a viable and uniform certificate revocation mechanism.

2.4. Emergence of short-term certificates

Up to this point, we have discussed multiple revocation protocols and extensions along with their associated issues. The unchanged situation is that the outstanding issues have always been significant enough to hinder unanimous adoption. In this sense, none of these solutions has completely resolved the certificate revocation problem.

To address this situation, IETF has adopted a different approach. In 2020, it published RFC 8739 – *Support for Short-Term, Automatically Renewed* (STAR) (Sheffer et al., 2020) to encourage web servers to migrate toward short-term certificates with a mechanism to automatically update them. This is an extension of *Automatic Certificate Management Environment* (ACME) in 2019 as specified in RFC 8555 (Barnes et al., 2019) to define an automatic process for *certificate users* to request certificate issuance. In the STAR scheme, *certificate users* periodically obtain new short-term certificates from an issuing CA on an ongoing basis. Clearly, the cadence of this automatic renewal is determined by the validity period in the certificate. Because the CA uses the original Certificate Signing Request (CSR) from the web host to generate a series of short-term certificates, these certificates are based on the same private/public key pair (Sheffer et al., 2020). Obviously, CA issues the next certificate only when no revocation has been triggered for the previous certificate, e.g., no key compromise.

Short-term certificates are otherwise normal certificates except that the validity period is in the order of days rather than months or years. The intent is to eliminate the need for revocation altogether by letting certificates naturally expire rather than proactively trying to revoke them when necessary. In other words, CAs' proactive action to revoke certificates and deliver that information to *certificate users* is replaced by the passive action of CAs' refusing to provide the next certificate. STAR RFC (Sheffer et al., 2020) recommends a validity period of 4 days, which is equivalent to the validity (caching) period of the OCSP response (Stark et al., 2012; Bruhner et al., 2022; Chuat et al., 2020). Therefore, adopting the STAR scheme is equivalent to supporting the OCSP protocol from a revocation-checking perspective.

3. Certificate collection approach

3.1. Prior work on certificate data collection and analysis

There are several prior works on TLS certificate collection and data analysis. Helme has a website (Helme, 2025) that shares the daily crawling of TLS server certificates from Alexa's Top 1 Million websites (Kaggle.com, 2025). Downloadable raw data in this URL have been added daily and show keyword-matching-based high-level statistics on various data points. However, their analysis is limited to the level of raw keyword-matching data counts only and does not provide any further analysis or insights beyond these data points. We note that the analysis part of this site appears to be no longer maintained.[8] Another website shows a similar type of website crawling (Qualys, 2025). Contrary to what Helme's site shows (Helme, 2025), this site analyzes the collected certificate-related data. However, the number of websites visited is limited to the top 150,000 sites – a significantly smaller sample space from Alexa's top 1 million sites (Kaggle.com, 2025). This site also appears to be no longer maintained.[9]

3.2. Methodology

To check the level of adoption of the STAR scheme, we conducted an extensive data collection and analysis of TLS server certificates. We visited the entire 1 million URLs listed in the Alexa Top 1 Million Sites (Kaggle.com, 2025), collected TLS server certificates, and analyzed them.[10] We consider the sample space of one million websites adequately represents Internet traffic, covering all major and popular websites as well as less-known and less-popular ones, thus serving as a sufficiently large sample for our purpose. Furthermore, covering beyond the top one million websites implies including significantly less-known and less-popular websites, which may risk skewing the statistics toward these types of websites. Furthermore, covering more websites will inevitably include an additional number of malicious websites with invalid certificates which we have already observed as a part of our analysis of one million sample space, and doing so will further skew the true characteristics of legitimate certificates.[11] In that sense, we consider the top one million as a sample space to be a good balance of capturing the true and correct representation of certificates in the Internet while avoiding being skewed by illegitimate or invalid certificates.

For the purpose of our certificate collection and analysis, we used TestSSL (Testssl.sh, 2025).[12] TestSSL is a bash script that captures TLS server certificates of the URL specified in the command line and outputs their content. In addition to the certificate content itself, it also collects and analyzes information related to the TLS session handshake procedure. We modified this script to make it fully automatic and manual intervention-free.[13] To automate the process of visiting a large number of URLs, we wrote a wrapper script to iterate through a list of URLs, trigger the TestSSL script for each URL, and redirect the screen output to a file for analysis. To further optimize the data collection process, we split this 1 million URLs into blocks of 1000 URLs and executed multiple wrapper script jobs simultaneously.[14] After this step is completed, we executed another script to analyze the collected certificate data.

TestSSL script reads received certificates in their original certificate formats, such as pem, der, and crt, and outputs their content in a human-readable text format. TestSSL script retrieves and outputs a much more detailed and extensive level of information than that of alternative methods, such as Openssl command line.[15] For example, a typical TestSSL output contains 21 items in a certificate, while openssl command output

[8] The last detailed analysis shown is from Nov. 2021.

[9] The last monthly scan was done on May 2024.

[10] We used the list downloaded on 4th of June, 2024.

[11] Our analysis of the collected certificates includes detailed classification of anomalies associated with certificates, including various types of invalid certificate content. In this paper, we do not present our analysis of these anomalies as they do not have direct relevance to our purpose of investigating the real-world adoption of short-term certificates.

[12] We used the TestSSL version 3.0.8, which was published in Sept. of 2022. The next 3.0.9 version was published in June 2024. However, we opted for the code stability and decided not to use the newer version as it was only a month before we conducted the data collection.

[13] Under certain circumstances, the TestSSL script stops and waits for a user input in order to determine whether to proceed or not. We removed the logic to wait for a user input so that the script automatically treats these conditions as a "yes" user response and proceed.

[14] During our data collection, we typically launched 20 to 30 wrapper script jobs on our Linux machine and let them run at the same time. We then periodically monitor the progress and launch new jobs so that there are approximately the same number of jobs running concurrently in order to optimize the efficiency of this data collection work.

[15] ''openssl x509 -in <cert_file> -text -noout'' command.

Table 4
Analysis comparison.

Data point	Analysis in Helme (2025)	Analysis in Qualys (2025)	Our analysis
# of URLs surveyed	1 million	~134,500	1 million
IPv4 vs. IPv6	×	×	✓
IP addr. obtained vs. visited	×	×	✓
Total cert. obtained	×	×	✓
# of cert. per IP addr.	×	×	✓
Cert. issuers	△	×	✓
Chain of trust verified	×	✓	✓
Cert. with CT supported	×	×	✓
Cert. types (OV, EV, DV)	✓	✓	✓
Validity period analysis	×	×	✓
Revocation info analysis	×	△	✓
Cipher suite	✓	△	✓
Signature algorithm	△	✓	✓
Server key size	△	✓	✓
Insights from observations	×	×	✓

Note ✓: covered, ×: not covered, △: partly covered.

contains 12 items only. Furthermore, in addition to the certificate content, TestSSL detects response parameters during the TLS session handshake, such as the number of IP addresses and the number of certificates provided by the web host, then it traverses all of these IP addresses and retrieves all certificates. In this way, in case the TLS session establishment involves multiple IP addresses and certificates, TestSSL captures all certificates with a single command. In this sense, the TestSSL output covers much more information at a detailed and extensive level than the openssl command. This level of information is essential for our analysis. The detailed data from the TestSSL output provided a significant level of information that gave us additional insights to our analysis.

We conducted this data collection between the period of July and August 2024. In the rest of this section, we present an overview of our analysis results; we then discuss key observations in detail in the next section. These analyses and resulting insights are our unique contribution not present in (Helme, 2025) and (Qualys, 2025) as shown in Table 4.[16] In addition to the certificate analysis, we integrate the collected information and present our discussion on the reasons why short-term certificates have not been adopted in reality. Based on our discussion, we further present potential alternatives to alleviate the current situation.

3.3. Data collection overview

Out of the 1 million website URLs (Kaggle.com, 2025), we obtained more than 1.5 million IP addresses. Among them, IPv4 and IPv6 addresses were ~1.16 million (~75%) and ~372k (~25%), respectively. We tried to contact all of these IP addresses and obtained a total of more than 1.5 million server certificates. Table 5 captures notable statistics from our analysis of the collected data. Analysis of these certificates reveals a number of observations.

The success rate of establishing a connection with these IP addresses was below 60%, which we consider to be rather low. Furthermore, even after establishing a TCP connection, we were not able to obtain server certificates from many URLs.[17] In our discussion, we limit the analysis of failure cases from the domain name (DN) perspective.

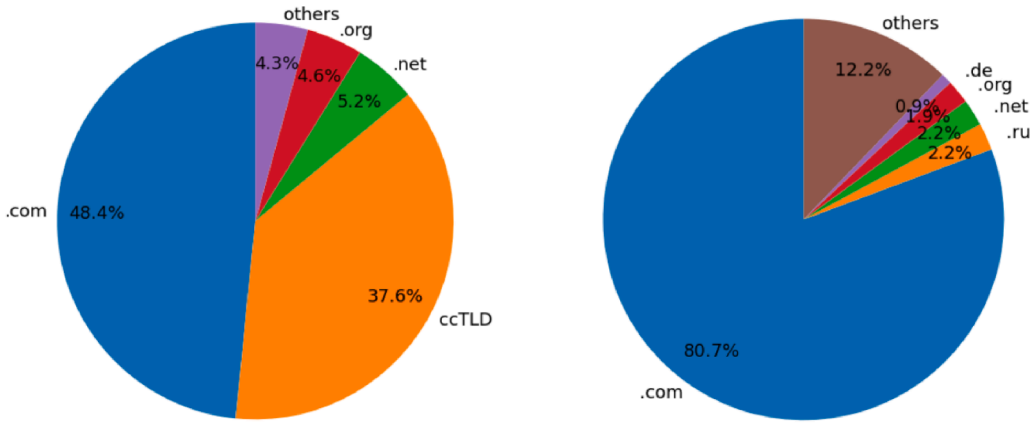
Table 5
High-level data analysis summary.

Metric	Count	%
IP Address:		
# of websites visited	1,000,000	–
Total # of IP addresses obtained	1,537,109	100.00%
# of IPv4 addresses obtained	1,164,701	75.77%
# of IPv6 addresses obtained	372,408	24.23%
# of IPv4/v6 addresses visited	892,718	58.08%
# of IPv4/v6 addresses failed to visit	644,391	41.92%
Certificate:		
# of certificates obtained	1,512,042	100.00%
# of cert. with chain of trust verified	1,465,804	96.95%
# of cert. with CT supported	1,495,275	98.96%
Validity Period:		
- The most frequent (90 days)	1,024,825	67.78%
- 2nd most frequent (365 days)	85,706	5.67%
- 3rd most frequent (396 days)	65,652	4.34%
# of Certificates with Revocation-Related Info:		
- # of cert. with CRL URI	803,225	53.12%
- # of cert. with OCSP URI	1,496,677	98.98%
- # of cert. with OCSP Staple support	858,260	56.80%
- # of cert. with OCSP Must Staple support	509	0.03%
Certificate Types:		
- EV	9277	0.61%
- non-EV (OV + DV)	1,502,765	99.39%
Signature Algorithm:		
- The most frequent (RSA w/ SHA256)	1,061,233	70.19%
- 2nd most frequent (ECDSA w/ SHA256)	301,145	19.92%
- 3rd most frequent (ECDSA w/ SHA384)	126,400	8.36%
- Others	23,264	1.53%
Server Key Size:		
- The most frequent (RSA 2048)	959,609	63.46%
- 2nd most frequent (EC 256)	452,318	29.91%
- 3rd most frequent (RSA 4096)	81,097	5.36%
- Others	19,018	1.26%
# of certificates/IP addr.:		
- The most frequent (3 certs)	716,891	47.41%
- Second most frequent (2 certs)	591,112	39.09%
- Third most frequent (4 certs)	180,029	11.91%
- Others	24,010	1.59%

The distribution of top-level domain names and their proportion in the connection failures are shown in Fig. 2. Fig. 2.(a) captures the proportion of DNs. There were a total of 725 unique top-level DNs in the 1 million URLs. Among them, the most common one is ".com," followed by ".net," ".org," and ".info." At the same time, there are a large number of so-called *country code top-level domains (ccTLDs)*, such as ".uk" for the United Kingdom or ".cn," for China (Ionos.com, 2025). There were

[16] The two references in this table ((Helme, 2025; Qualys, 2025)) are information from websites; they are not academic research papers.

[17] We observed various types of problems, such as TCP connection failure, web server not running TLS, and abnormalities in certificate content itself. Furthermore, the original 1 million website csv file contains 978 cases of ill-formatted URLs where the top-level DN is not present (although this represents 0.1% of the total URLs, it is not a negligible observation). These anomalies impact our data collection and the resulting analysis; this situation is reflected in the reality of our data collection. Although these anomalies by themselves represent an interesting area to investigate, it is outside the scope of this paper. Therefore, we do not discuss them further in detail.



(a) Top-Level DN Distribution

(b) Per-DN Type Connection Failure Distribution

Fig. 2. DN-based distribution analysis.

a total of 243 such unique ccTLDs (i.e., unique countries). Although the number of occurrences for individual ccTLDs is significantly small, the combined total amounts to 37.6% of the total URLs if we consider them as a single *ccTLDs-type* DN. This represents approximately 1/3 of the total URLs. Fig. 2.(b) captures the distribution of total connection failure events from the DN perspective (total of over 644,000 events from 464 top-level DNs). The failure events that occurred in the top three top-level DNs (".com," ".net," and ".org") reflect the proportion of the DN distribution in Fig. 2.(a). However, one notable exception is Russia (".ru") and Germany (".de"), which are at the similar level with ".net" and ".org." The remaining 459 DNs represent the remaining 12% of connection failure events.

The ccTLDs URLs directly map to the website's geographical area. In contrast, more conventional top-level DNs, such as ".com" do not indicate which country or region the website belongs to. The same discussion applies to the relationship between URLs and their business or industry segment. In some cases, relatively new types of top-level DNs indicate the type of business or the nature of the website content.[18] However, they are minorities in several magnitudes compared to the most dominant one, such as ".com." Therefore, their proportion is too small to be meaningful in practice to classify the business and industry of a given URL.

As a result, these *geography/region-neutral and business/industry-neutral* URLs constitute the large majority of the remaining 62.4% of the DNs besides ccTLDs, and the three largest top-level DNs ("com," "net," "org") constitute approximately 85% of all the connection failures. These URL types hinder from gaining accurate insights into the distribution of websites, in both geography/region and business/industry perspectives, that consists the 1 million websites around the world. In this sense, it is difficult, if not impossible, to have an accurate understanding of any geographical/regional or business/industrial concentration of irregular occurrences, such as various failure causes to obtain certificates and anomalies in the certificate content.

4. Analysis

Based on the methodology we discussed in Section 3.2, in this section, we present our analysis of the collected data, focusing on understanding the level of support of short-term certificates in practice. In this context, we also discuss closely-related information that influences the adoption of short-term certificates, including (1) the most popular validity periods, (2) dominant certificate issuers, and (3) support of revocation check protocols. Furthermore, we also examine other related information in certificates: (4) chain of trust check, (5) certificate types, (6) signature algorithms, (7) server key size, (8) number of certificates per IP address, and (9) certificate transparency indication.

Table 6
Certificates with validity period up to 30 days.

Validity period (days)	# of occurrences
1	7
6	2
8	1
9	1
10	2
11	1
14	54
19	3
20	4
26	3
28	1
29	1

4.1. Short-term certificate support

From the perspective of short-term certificates, which is characterized by a validity period of 4 days as stated in STAR (Sheffer et al., 2020), we observed no implementation of this concept. There is clearly no "buy-in" of this idea by CAs and web hosting service providers in reality in the past five years after the STAR RFC was published.[19]

If we extend the definition of *short-term* up to several weeks, then we observe a small number of certificates supporting validity period as shown in Table 6. In our entire data set, there were in total 80 certificates (0.529%) with a validity period of less than 30 days.

4.2. Validity period

Fig. 3 shows the validity period distribution of the top 10 most frequently-occurring validity period. The dominant one is the period of

[18] Some of these examples include ".xxx," ".sex," and ".porn" for adult-content websites. Others self-explanatory top-level DNs include ".law," ".realtor," ".restaurant," ".mortgage," ".investments," etc.

[19] The shortest validity period we observed was 1 day. However, we consider this to be an isolated case and not necessarily the support of STAR as the number of such occurrences is only seven (7), and the next shortest validity period is six (6) days. Even if we consider these seven certificates as legitimate and indicate the support of STAR, it is essentially non-existent compared to the sample space of over 1.5 million certificates.

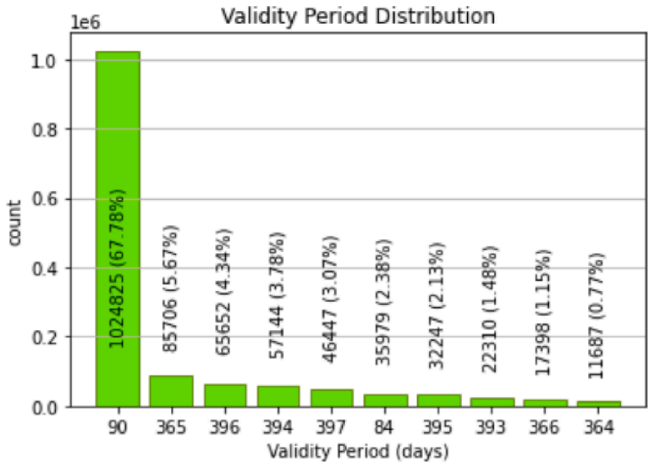


Fig. 3. Validity period distribution.

Table 7
Issuers of 3 most popular validity periods.

Issuer	Count	Percentage
90-Day Certificates:		
Total	1,024,825	100.00%
Let's Encrypt	594,883	58.05%
Google Trust Service	413,754	40.37%
Others (28 issuers)	16,188	1.58%
365-Day Certificates:		
Total	85,706	100.00%
GoDaddy.com	22,387	26.12%
Cloudflare	15,124	17.65%
Sectigo	12,483	14.56%
DigiCert	12,234	14.27%
Others (1962 issuers)	23,478	27.40%
396-Day Certificates:		
Total	65,652	100.00%
Sectigo	38,706	58.96%
DigiCert	17,697	26.96%
Others (70 issuers)	9249	14.08%

90 days, which accounts for over 1 million certificates, or approximately 2/3 of the entire certificates (67.78%). All others are overshadowed by this. The far-distant second and third are 365 days and 396 days with 5.67% and 4.34%, respectively. The list of the 3 most popular validity periods and their top issuers is summarized in Table 7.

These 90-day validity certificates were issued by 30 issuers. Among them, 58.05% (~595k) were issued by Let's Encrypt, followed by 40.37% (~414k) issued by Google Trust Service (GTS). These two issuers accounts for 98.42% of all 90-day certificates combined. The remaining 28 issuers account for merely 1.58% of all 90-day certificates combined. The large number of these certificates issued by Let's Encrypt reflects the growing popularity of free service to provide DV certificates (Bruhner et al., 2022; Dickinson et al., 2019).

4.3. Issuers

Further, Fig. 4 shows the proportion of certificate issuers in decreasing order. Let's Encrypt and Google Trust Services (GTS) are responsible for issuing certificates over 1 million certificates combined, or ~70% of the entire certificates.[20] As a whole, the top 8 issuers issued over 1.4 million, or a little over 94% of all certificates. There are total of 1266 issuers in our collected data. In other words, the remaining ~5% of the certificates were issued by the remaining 1258 issuers. Put in another

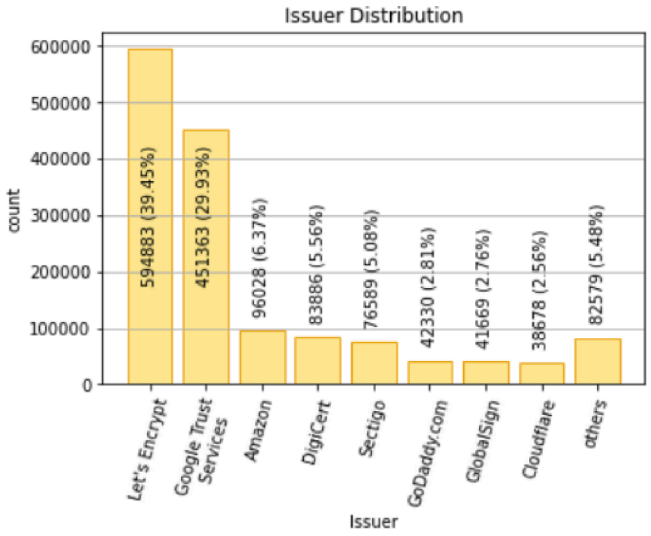


Fig. 4. Certificate issuer distribution.

Table 8
Changes of revocation protocol support in websites since 2018.

Data point	Data from Chung et al. (2018) (2018)	Our analysis (2024)
CRL support	not mentioned	53.12%
OCSP support	95.40%	98.98%
OCSP staple support	not mentioned	56.80%
OCSP must staple support	0.02%	0.03%

way, 99.37% of the issuers issued only ~5% of the certificates. Among these remaining issuers, the large majority of them (1063 issuers, or 83.96%) issued less than 10 certificates. In fact, more than half of the total issuers (736 issuers) issued only one certificate.[21] These numbers indicate how dominant the top 8 issuers are in the certificate generation service despite the total number of issuers in this business.

The reason behind the dominance of 90-day certificate is explained by Let's Encrypt being the largest certificate issuer. Let's Encrypt offers a free service to issue certificates, and its validity length is fixed to 90 days (Encrypt, 2025).

4.4. Revocation protocol support

Table 8 captures the changes in the websites' support of revocation protocols and extensions since 2018. This comparison is our data against what is stated in Chung et al. (2018) as a point of reference. The support of OCSP has increased by 3.5% to ~99% of all certificates in the last 6 years. OCSP staple support has gained traction to ~57%. On the other hand, the support of OCSP *Must Staple* extension shows only a slight increase at 0.03% from 0.02% in 2018, indicating only negligible increase in the last 6 years. From this, it appears that the implementation of this extension is at the end of the road, hinting the CA community's lack of appetite to adopt this feature.

4.5. Chain of trust check

Of all the certificates we obtained, 96.95% of them successfully verified the chain of trust. The remaining 3.05% (or 46,151 certificates) indicate failures due to either expired, self-signed, incomplete chain, or unknown reasons. Among them, self-signed and incomplete chain appear to indicate maliciously created fake certificates.

[20] As a side note, a paper in 2014 (Durumeric et al., 2014) states that the largest CA at that time was GoDaddy.com.

[21] We suspect at least a portion of them are malicious certificates such as self-signed certificates.

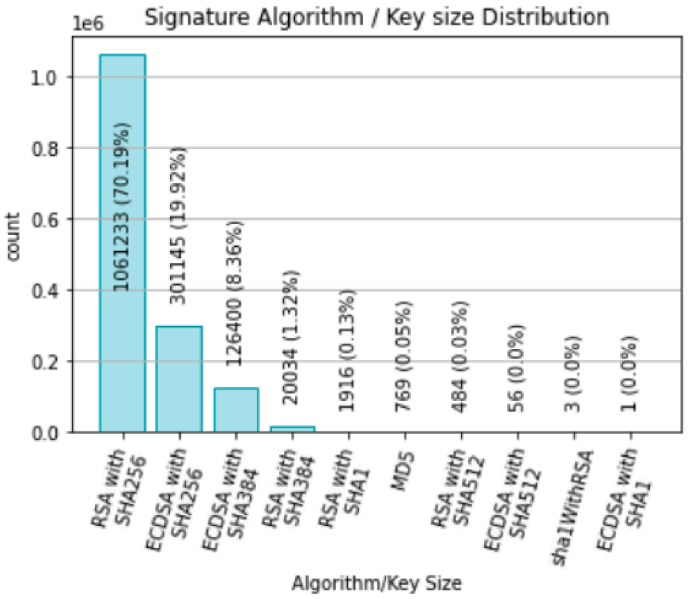


Fig. 5. Signature algorithm distribution.

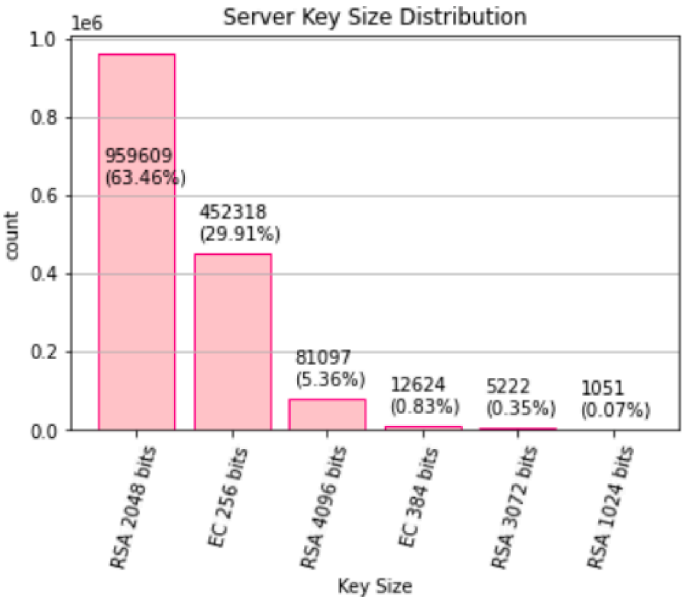


Fig. 6. Server key size distribution.

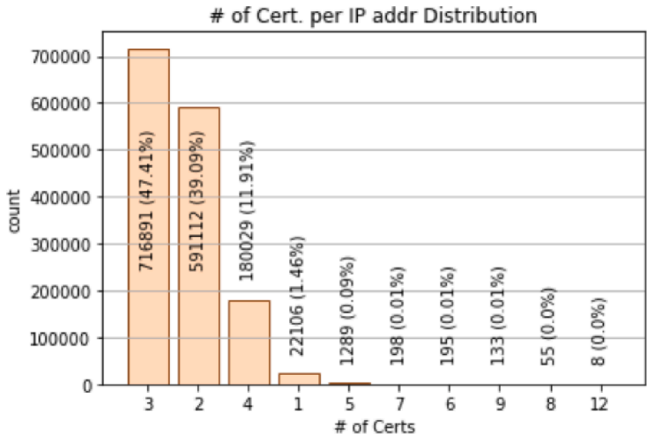


Fig. 7. # of certificate per IP address distribution.

4.6. Certificate types

Regarding the certificate types, OV and DV certificates represent more than 1.5 million certificates, or more than 99% of the entire certificates. The EV type is less than 9300 which is merely 0.61%. It reflects a relatively small proportion of large companies and organizations that tend to hold EV certificates within the list of 1 million websites. Furthermore, due to the change in browsers in the last few years to stop the visual indication of the EV type certificate in the top bar ('green banner') (Bruhner et al., 2022), it gives less advantages of obtaining EV certificates, resulting in less incentives for companies to go through the more expensive, complex, and time consuming process to obtain EV certificates. These circumstances may have contributed to this low proportion of EV certificates.

4.7. Signature algorithms

Fig. 5 shows the signature algorithms used to establish a TLS session. It is notable that RSA with SHA256 is the highest with 70.19% of all certificates. Although ECDSA was introduced over 20 years ago, it's usage (e.g., ECDSA with SHA256, and with SHA384) is still relatively low at ~36% combined. We suspect it is likely due to the default setting in automatic certificate generation services such as Let's Encrypt and other dominant certificate issuers rather than specific technical reasons to avoid using ECDSA.

4.8. Server key size

Fig. 6 shows the server key size distribution. RSA 2048 bit is the most dominant one with almost 1 million certificates (or 63.46% of all certificates). This is followed by EC 256 bits with less than half of RSA 2048 bit (29.91%), followed by the distant third of RSA 4096 bits (5.36%). The remaining key sizes are all under 1%.

4.9. Number of certificates per IP address

Fig. 7 shows the number of certificates per IP address. A large portion of all IP addresses provided either two, three, or four certificates (98.4% of all certificates combined).

4.10. Certificate transparency indication

Certificate Transparency (CT) was first introduced by Google in 2011 and became the IETF standard (RFC 6962) in 2013 (Laurie et al., 2013).[22] It has gained wide adoption in the PKI community since then. Google and Apple require that all certificates be included in the CT log since 2018 (Bruhner et al., 2022; Chuat et al., 2020). Our collected data indicate that approximately 1.5 million (98.96%) certificates are registered in the CT log. The CT gives assurance to the *certificate users* (e.g., browsers) that a certificate is appropriate and was issued by a legitimate CA rather than a compromised CA (Wikipedia.org, 2025).

5. Discussion

Subsequent to our detailed analysis in Section 4, we now turn our attention to the certificate revocation issue from different angles. We first distill the result of our analysis and examine the reasons behind this absence of support for short-term certificates, followed by studying the countertrend to shift toward browser vendors' proprietary solutions as an alternative approach to the standardized solution. Based on this

[22] The latest version is RFC 9162 published in 2021 (Laurie et al., 2021).

understanding, we consider a possible way forward. We consider these points to offer insightful observations.

5.1. State of short-term certificate implementations

As clearly shown in Section 4, there is no real-world adoption of 4-day certificates after five years IETF published the STAR RFC (Sheffer et al., 2020). The STAR concept exemplifies a new approach to the certificate revocation problem by removing the need to actively revoke certificates altogether. In this sense, this is a different approach to address the revocation issue from existing solutions such as CRL and OCSP and its variants.

We consider several possible explanations for the absence of adoption in reality. First, its wide adoption is at the mercy of CAs to carry the burden of increasing amount of certificate generation and distribution to web hosts. They need to process certificate generation for a potentially very large number of web hosts with increased cadence, and provide new certificates on an ongoing basis as the validity period becomes shorter.[23] For this reason, this scheme requires a *buy-in* from the CA community.

Second, this scheme does not offer incentives for CAs. In order for this scheme to generate traction in the CA community, there need to be inherent values or benefits for CAs to adopt it. Without it, the CA community has no motivation to adopt a new scheme (Ma et al., 2023; Register, 2024). As a standard document, the STAR RFC merely defines a technical solution. In this respect, the top-down approach in which a standardization body defines and enforces a particular technical solution to the community without addressing incentives for them may be questionable with respect to its effectiveness. We consider it to be the fundamental obstacle of the short-term certificate approach.

5.2. Entangled situation – standardized vs. proprietary revocation schemes

Gutmann (Gutmann, 2002) argued that the fundamental concept of revocation is flawed as once a certificate is sent for use, it cannot be taken back. From that perspective, the revocation mechanism, such as CRL, is an uphill battle from the beginning. Furthermore, CRL represents *blacklist*-based status checking, which cannot provide a definitive answer regarding a certificate's validity. In other words, the CRL can only answer *whether a given certificate is revoked or not*; a negative response (i.e., an absence of revocation) does not necessarily mean that it is still valid. For example, if a CA generates a new CRL every x days, additional revocation events between those x days are not reflected in the CRL until the next generation cycle. Therefore, nonpositive outcomes represent a combination of true-negative and false-negative cases, making it impossible to discern a definitive conclusion.

The OCSP scheme also contains design issues in its very concept, as it verifies the current status of something that was issued in the past; it is akin to obtaining *a certificate of a certificate* (Helme, 2024), which is a convoluted idea to begin with. In addition, OCSP's *real-time* checking is a false premise if the underlying information in the CA is based on CRL which was generated, e.g., a few days ago. In addition, a negative response can mean a number of nonpositive situations such as the system simply fails to obtain the necessary information to determine the validity status of the certificate in question such as OCSP Responder being unavailable (Wiki, 2025). Further details on OCSP issues are described in Gutmann (2024).

Although the support of OCSP is near 100% according to our observation of our collected certificates (Tables 5 and 8), it only gives a false sense of premise. The actual use of OCSP is dependent on the browser being used, as discussed in Section 2.3; only Firefox and Safari out of the five major browsers trigger OCSP when deemed necessary. Based on the browser market share information as of Aug. 2024 (Statista.com, 2025), OCSP is triggered in only 21.31% (Safari: 18.57% and Firefox: 2.74% combined) out of all TLS session establishment even if the OCSP support becomes 100% of all websites in the world. In other words, irrespective of CA's effort to support OCSP, approximately 80% of TLS session establishment ignores it and bypasses OCSP altogether when either Chrome, Chromium-based variants, or Edge is running on the users' computer. Furthermore, the OCSP Staple support in web servers is less than 60% as shown in Table 8. Therefore, more than 40% of web servers do not voluntarily trigger OCSP. In this sense, a mere indication of OCSP support in server certificates gives a misleading impression of the actual utilization of OCSP in reality.

5.3. Browser vendors gaining the upper hand

In reality, how revocation is done appears to be in the hands of browser vendors already as their voices gain more weight. In this context, the level of influence standards have may also be diminishing. There are several points to be highlighted.

First, the numerous issues in the historical standardized solutions we discussed are likely the starting point of this trend. This may explain, at least in part, the reasons why browser vendors defined and opted for their own proprietary solutions such as *CRLSet* and *OneCRL*. Covering trusted CAs' certificates in these proprietary lists allows them to cast a large net over the vast number of certificates and allows browser vendors to detect revocation of CA certificates – a high-impacting event that triggers revocation of all still-valid certificates the revoked CA has issued in the past (Durumeric et al., 2014). At the same time, they may have concluded that validity check on EE certificates is too expensive with little return of investment, thus not worth the effort.

Second, as shown in Table 3 and Fig. 4, a small number of large companies takes almost all browser market share. This situation makes them dominant in the market, thus paves a way for them to reflect their intent on browser implementations without concerns for severe repercussions or negative consequence. Furthermore, as browsers directly interact with end users, end users' perceptions of the browsers' performance are of prime importance to browser vendors. Specifically, the users' perception of the page refresh speed is one of the key elements for users to select browsers (Highspeedinternet.com, 2025; Safetydetectives.com, 2025). From this point of view, any event that can negatively impact user perception is not welcome from browser vendors' perspective (Wiki, 2025). Especially, security-related operations, such as executing OCSP protocol, are invisible to end users. From this point of view, these user-invisible security-related operations are necessary evils that act as a speed bump; it is a balancing act between ensuring security and minimizing negative impacts visible to users. In that sense, their design decision may be driven by such considerations.

Third, the trend driven by browser vendors to set an upper limit and increasingly shorten certificate's validity period appear to be gaining traction (Topalovic et al., 2025). In 2023, Apple and Google announced that the maximum duration of a TLS server certificate should be no longer than 398 days for certificates issued after September 1, 2020 (Apple, 2023; Chromium, 2025).[24] Continuing this trend further, Apple, Google, and Mozilla recently announced to adopt 90-day maximum validity period, followed by Apple and Google to further reduce it to 45 days by 2027 (Nameshield.com, 2024). To back up this trend further, a study shows positive effects of short certificates. One study found that 90-day certificate lifetime reduces the stale certificates by 75–86% due to the increased *agility* compared to longer validity period (Ma et al.,

[23] Due to the variations in practical operating environment in individual CAs, such as processing capacity and available resources, the exact impacts on CAs will vary from one CA to another. Furthermore, the proportion of reduction in the validity period will also be a factor to impact the CAs.

[24] Apple's explanation (Apple, 2023) states that this change affects TLS server certificates issued from the Root CAs preinstalled with iOS, iPadOS, macOS, watchOS, and tvOS for TLS server certificates issued on or after September 1, 2020. Google (Chromium, 2025) justifies this change that short certificate lifetimes protect users by reducing the impact of compromised keys.

2023). Although these are significantly longer than STAR-recommended 4-day period, the implication is that both Chrome and Safari will reject certificates that do not meet these criteria.

Whether these proposals to shorten the validity period in the Certification Authority Browser Forum (CA/B Forum) (Forum, 2025) go through or not, browser vendors can make it their policy on their browsers and force the CA community to comply (Register, 2024). As a result, in this browser-vendor-driven paradigm, the *revocation management* has been embedded within their control by including the necessary information or logic within their infrastructure and browsers – which certificates and web sites are checked and what validity period is accepted or not are up to browser vendors' decision. This may be the *only* viable solution from the perspective of browser vendors.

As more time passes, it most likely becomes increasingly difficult to establish a standardized solution that solves all the outstanding issues associated with CRL and OCSP and its variants, and convince browser and CA communities to change their implementations.

5.4. Risks of browser-vendor-driven revocation paradigm

As discussed in the preceding section, revocation checking becomes more and more dependent on dominant browser vendors' implementation decisions without end users' awareness. This means that browser vendors can implement their preferred solutions without considering whether it is the best one from the end-users' perspective or the most secure solution. One key factor for this situation is that, security-related operations, such as revocation checking, are invisible to end users, thus it is difficult, if not impossible, to raise their awareness.

This browser-vendor-driven paradigm implies several potential risks, which may possibly lead to a new type of problems. These risks can be characterized as: (1) inconsistent revocation checking, (2) unclear boundary of responsibilities and erode CA's authority and accountability, and (3) potential increase of threats and vulnerabilities in browsers. We elaborate each of these points.

First, vendor-specific implementations mean that different browser vendors employ similar but different approaches to verify the validity of certificates, as we discussed in Section 2.3. These solutions have similar objectives, but differ in detail. These differences include the level of CA certificates to check, which CAs to check, the frequency to check those CAs, how frequently the revocation database within the vendors (i.e., databases in CRLSet, OneCRL) is refreshed with the latest revocation information, and how frequently those updated information are disseminated to the client browser end users use as part of the software update. These differences lead to inconsistent and nonuniform effectiveness, i.e., varying browsing experience among users of different browsers while accessing the same website depending on the relative timing difference of these refresh events. In other words, if a user tries to access a website for which the certificate has been revoked recently, one browser may correctly treat this certificate as revoked and block access to it, while another browser considers it to be valid and renders the website content on the user's screen.

Second, the increasing role of browser vendors to disseminate revocation information makes the boundary of responsibility between CAs and browser vendors less clear. In this paradigm, CA's revocation database becomes an upstream data source from which browser vendors update their internal database. In other words, although the revocation information originates from CAs' CRLs and OCSP Responder, the downstream database in the browser vendors becomes the final *authority* in reflecting those information to update the revocation information to the browser. In this sense, the browser vendors' increasingly active role diminishes the CAs' authority and accountability in managing the dissemination of revocation information, making it less clear who (i.e., CAs or browser vendors) is responsible for enforcing revocation information.

Third, the increasing role in both browser itself and the browser vendor's infrastructure will result in increased vulnerabilities and threats against adversarial attacks. If either the browser client software itself or the vendor's revocation database is compromised, it can potentially circumvent all revocation checking, rendering the revocation useless. Because browser client software operates on the end users' computers, the increased accessibility by adversaries results in the increased attack surface and higher threat level. Because of this paradigm of the browser vendor (client software and their revocation database) acting as the front-end of dissemination of revocation information to end users, failures can occur even though CAs correctly process the revocation information.

Mitigating these risks is not trivial, especially from the perspective of end users. The first challenge regarding end users is to raise their awareness of the revocation mechanism. However, this is a challenge by itself as any cybersecurity-related functionalities on computers are largely invisible to them, and they may not necessarily be interested in this topic. Considering this in mind, a few possible mitigation approaches may include the followings: (1) make security-related operations in browsers more visible to users to raise their awareness and the effectiveness of these features visible, and (2) user-friendly and easy-to-understand browser configuration at its initial use to aid users to determine security-related configuration settings so that users can consciously select preferred browser behavior.

5.5. Device-type-dependent revocation checking variations

Another aspect of the variation in the revocation check is the difference driven by the device types. Three types of devices are most representative in this discussion: (1) desktop and laptop computers, (2) mobile devices such as smartphones and tablets, and (3) IoT and embedded devices. There are several key determining factors that drive the preferred revocation behavior. These factors include assumed operating conditions and constraints, such as (1) whether stable Internet connection is available or not, (2) whether the device is power-supplied or battery-operated, and (3) how much computing resources are available. This categorization and characterization is shown in Table 9.[25]

Mechanisms that require the on-demand connection to the Internet, such as OCSP and its variants, are possible if the device has stable Internet connection with less concern on power usage and computing resource constraints. Unavailable or unstable network connection can lead to delay in web page rendering due to OCSP timeout, and can negatively impact user experience. On the other hand, device types that cannot assume stable connection or limited computing resources tend to gravitate toward less communication-demanding, less-processing, and less-memory-intensive options.[26] From the perspective of practical operation stability and user-experience, the *safest* option is to bypass revocation checking altogether under such circumstances.

Desktop and laptop computers tend to have stable power sources, such as AC power from the wall, and have relatively large computing resources relative to other device types. On the other hand, due to their limited form factor, mobile devices are equipped with less powerful resources, with possibly less stable connectivity than desktop computers, and are powered by a battery. IoT or embedded devices tend to have even less computing resources, with limited connectivity and running on a battery with possibly an expectation to operate for an extended period of time, such as months or years.

From the security perspective, the use of device types that have fewer computing resources and less-stable connectivity has higher dependency on less-optimal revocation checking or no checking at all. Furthermore, adoption of short-term certificates for mobile and IoT devices implies a more frequent replenishment of certificates, which requires additional processing and battery power consumption. In this sense, both more

[25] This categorization and the characterization of each category in this Table is an indicative in nature based on their form factors and usages.

[26] In this context, computing resources include, such as processing power, memory and storage capacity, and power usage (i.e., battery drain in case of battery-operated devices).

Table 9
Device-type-dependent variations and the suitable revocation checking mechanisms.

Device Type	Connectivity	Power Source	Computing Resource	Likely revocation behavior
Desktop	Stable	AC	Large	Most dynamic solution (OCSP)
Mobile	Less Stable	Battery	Medium	More static solution (No or limited revocation check)
IoT/Embedded	Less stable or none	Battery	Small	More static solution (No revocation check)

sophisticated revocation check mechanisms and the use of short-term certificates are more taxing for these device types, to the extent that they cannot support either of these approaches. This situation poses a challenge as consumers increasingly use or rely on mobile devices in daily life, and numerous IoT devices are deployed for a variety of applications. In other words, practical options are limited due to hardware limitations and operational constraints. As a result, user-invisible functionalities, such as security-related mechanism may be ignored.

5.6. Possible way forward

Further to our discussion, we present other views on a possible avenue to address the revocation problem. These ideas are not part of the existing standardized solutions we discussed in Section 2.1. In this respect, we explore new insights from different perspectives and a possible way forward out of them.

5.6.1. Relaxing the 4-day validity period

The first point we raise is the legitimacy of the 4-day certificate as recommended in the STAR RFC, and propose a change in this area. We consider that this 4-day validity period is an arbitrary value rather than the result of a concrete and justified reasoning. In fact, the reason behind this 4-day period is merely to replicate the existing OCSP behavior.[27] Therefore, this rigid and inflexible validity period definition is worth reconsideration.

Regardless of the actual validity period, there is always a time gap associated with the revocation triggering event and its time of notification to the *certificate user*. The difference between 4-day and 90-day as the validity period is the difference in relative time gap during which the *certificate users* are unaware of the revocation status change since it was issued (i.e., ~4 days or ~90 days). To put it in another way, if 90 days is too long, is 4 days short enough? This is a question of risk tolerance: how much risk one is willing to accept; it is a case-by-case matter depending on the unique factors of the website owner. The discussion in the preceding paragraph clearly indicates that there is no definitive and universally applicable answer to this question. In this sense, we argue that there is no absolute reason to specify and enforce 4 days as the validity period.

In the rest of this subsection, we describe two possible approaches to determine the validity period, one based on self-conducted risk assessment and another based on website categorization.

5.6.2. Risk assessment-based validity period selection

To perform a risk assessment, organizations must follow a well-defined and methodical approach. For this, standardized guidelines for risk assessment serve as a useful guide; these include ISO 31000 (ISO, 2018) and NIST SP 800-30 (NIST, 2012). These standards involve a structured approach consisting of several distinctive steps in its process.[28] For our discussion, we focus on one of these steps: *risk assessment*. Risk assessment involves three components: (1) *identification*, (2) *analysis*, and (3) *evaluation* of risks. In risk *identification*, an organization describes potential risks and their consequences. In risk *analysis*, an organization evaluates the level, sources, and likelihood of risks. Then, in the risk *evaluation*, an organization determines the significance of risks based on the results of the risk analysis.

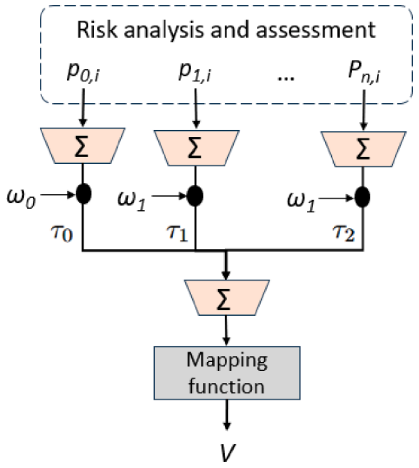


Fig. 8. Validity period generation scheme.

Using this approach, we describe one possible scheme to derive the certificates' validity period. Fig. 8 and Table 10 illustrate this general concept. This scheme is based on the self-evaluation of risk analysis and assessment conducted within the organization that owns a website. Based on the self-evaluation score, a total point is derived. This self-evaluation may include multiple categories, which likely consist of multiple criteria for self-evaluation. In this example, we show three types of categories - *organizational characteristics*, *organizational capability*, and *organizational policy*. Under each category, there are multiple areas to analyze and assess to accurately reflect the organization. Our example in the right column in Table 10 is a point in the scale of 1 to 10. As the organization fills this column, the total points for all categories are obtained.

The mathematical representation of deriving the assessment value (AV), as illustrated in Fig. 8, is shown in Eqs. (1) and (2). The τ_i value in the first equation represents the total points for category i ($p_{i,j}$ and c_i are the points of the criterion j in category i, and the number of criteria for category i, respectively). AV in the second equation is the percentage value of the total points for all categories, with a weight factor taken into account. This weight, w_i ($0 < w_i \leq 1$), represents the relative importance of each category perceived by the organization.[29]

$$\tau_i = \sum_{j=0}^{|c_i|-1} p_{i,j} \tag{1}$$

$$AV = \left\lfloor 100 * \frac{\sum_{i=0}^{n-1} \tau_i * w_i}{\sum_{i=0}^{n-1} \tau_i} \right\rfloor \tag{2}$$

[27] In fact, STAR (Sheffer et al., 2020), Section 4.1 states: *'"Short Term" is a relative concept; therefore, trying to define a cutoff point that works in all cases would be a useless exercise.'* At the same time, it also states (same section): *'Given OCSP responses can be cached, on average, for 4 days [STARK], it is RECOMMENDED that a STAR certificate that is used on the Web has an "effective" lifetime (excluding any adjustment to account for clock skews) no longer than 4 days.'*

[28] These steps include: (1) communication and consultation, (2) scope, context and criteria, (3) risk assessment, (4) risk treatment, (5) monitoring and review, and (6) recording and reporting (IRM, 2018).

[29] The weight factor (w_i) can also be determined by the organization based on their relative importance of these categories.

Table 10
Risk assessment and evaluation criteria (example).

Category 1: Organizational Characteristics	Pt (1~10)
Assessment and Evaluation of:	
•Business criticality / sensitivity (e.g. mission critical, secret)	x
•Business impact / exposure (e.g. damage to reputation/credibility)	x
•Business mission / operation / functions (e.g. significance of business, risk of financial loss)	x
•Business relationship to external entities (e.g. clients/partners)	x
•External dependencies (e.g. suppliers, clients)	x
Category 1 total (τ_0)	**X**
Category 2: Organizational Capability	Pt (1~10)
Assessment and Evaluation of:	
•Organizational governance (level of org. maturity)	y
•Intrusion detection (timely detection)	y
•Incident response capability (timely response/recovery)	y
•Threat event/source detectability / identifiability (e.g. effective event/log analysis)	y
•Impact analysis (effective damage assessment)	y
•Assessment of adversary (understand adversaries' capabilities/intent)	y
•Assessment of impacts/damages on business operation	y
•High availability / infra. hardening (robustness/resilience of infra.)	y
•Expertise/knowledge/skill of staff	y
Category 2 total (τ_1)	**Y**
Category 3: Organizational Policy	Pt (1~10)
Assessment and Evaluation of:	
•Business / management and operational (maturity of policy/guideline)	z
•Contingency / recovery plan (readiness for immediate action)	z
•System mgmt./monitoring/maintenance (proactive prevention)	z
•Event monitoring / logging (continuous monitoring)	z
•Periodic penetration testing (proactive prevention)	z
•Periodic risk analysis/assessment report (response readiness)	z
•Regulatory compliance (incident reporting)	z
•Continuous personnel / employee training (ongoing staff skill level)	z
Category 3 total (τ_2)	**Z**

The resulting validity period V is derived from AV using a mapping function as shown in Eq. (3).[30]

$$V = \begin{cases} 55 & \text{for} \quad 95 \leq AV \\ 45 & \text{for} \quad 75 \leq AV < 95 \\ 35 & \text{for} \quad 55 \leq AV < 75 \\ 25 & \text{for} \quad 35 \leq AV < 55 \\ 15 & \text{for} \quad 15 \leq AV < 35 \\ 5 & \text{for} \quad AV < 15 \end{cases} \tag{3}$$

5.6.3. Categorization-driven validity period selection

World-Wide Web (WWW) consists of a diverse spectrum of websites, owned by different types of entities serving different purposes. Website owners include government or public offices, various sizes and types of commercial companies, organizations, educational institutions, individuals, and many others. Some of them benefit from marketing higher security to project trustworthy images, such as banks, investment companies, insurance companies, law firms, etc.[31]

Despite this diversity of websites, the categories of digital certificates are quite limited; there are only three categories to differentiate the certificate types (i.e., OV, DV, and EV), and their differentiation is diminishing.[32] Furthermore, these certificate types merely reflect the level of rigorousness during the approval process of certificate issuance. To put it simply, ownership of EV certificates means that the owner of the website is legitimate by being approved as a result of a more rigorous process than other certificate types. However, it does not represent the type of owners, nature of websites, their objective, charter, and purpose. In other words, the existing certificate definition lacks an attribute that characterizes the type of owner and entity that the certificate represents. This is an imbalance between the diversity of *certificate users* (websites) and the certificate's ability to express attributes associated with them. Given the growth and proliferation of Internet usage in the past decades, the certificate definition should also evolve to reflect such changes. From this perspective, we believe it is worth considering ways to resolve this evolving gap going forward and utilize such a solution to help optimize the selection process of certificates' validity period.

In conventional certificate generation, its validity period is determined by the certificate issuers (CAs) at the time of its generation without considering other factors. In this sense, introducing other factors, such as website types and their services, and reflecting this information to derive a validity period may represent a novel approach.

Website categorization is a classification mechanism that translates website identity (domain name or URL) to its corresponding types. It utilizes content analysis and classification algorithms as the key ingredient in this service. Due to this nature, this is achieved by the use of machine learning (ML) algorithms (Bruni and Bianchi, 2020). There are many uses for this service, including website filtering and malicious site detection (Cohen et al., 2021), market and advertising (Kagan and Bekkerman, 2018), data management and analysis (Roberts et al., 2021), and others.

There are multiple online website categorization services (Medium.com, 2022).[33] These online platforms provide an API for others to query categorization by providing the domain name

[30] This mapping function is intended for illustrative purposes only. The exact value range and granularity can be adjusted.

[31] The 'green banner' at the top of the browser, indicating the use of EV type certificate, is a prime example, although all major browser implementations stopped this practice around 2019 (Namecheap.com, 2025).

[32] As discussed in Section 4.6, the EV type certificates represent only 0.6% of the entire certificates we collected from 1 million websites. The value of EV type appears to diminish over the years, partly due to the discontinuation of 'green banner' discussed earlier.

[33] Examples of online categorization services include *Websitecategorization-api, Cyren, WhoisXML, SimilarWeb, and SafeDNS* (Websitecategorizationapi.com, 2025; Techslang.com, 2025).

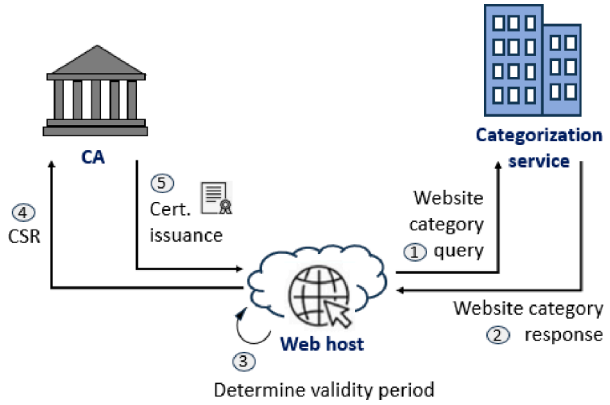


Fig. 9. Website category-based validity period.

or URL in question. Many of these platforms use the Internet Advertising Bureau (IAB) taxonomy (Bureau, 2025), which is tailored for marketing and advertising segments. There are other taxonomies that are specific to E-commerce, including the *product taxonomy* compiled by Google (Google.com, 2025) and Facebook (Facebook.com, 2025).

As a potential approach to realize such *website-type-driven* validity period differentiation, the use of these categorization services can be integrated in the CA's certificate generation process. This idea is illustrated in Fig. 9.

First, a web host sends a category query request of itself to the external categorization service platform by providing its domain name or URL (step ①), to which this service returns the corresponding category in a response message (step ②).[34] This response contains a signed result of this service platform. Based on the response in step ②, the web host derives a suitable validity period that matches its website (step ③). Next, the web host sends a CSR to the CA, including the requested validity period along with the signed response in the previous steps (step ④). The CA verifies the signed results from the categorization service platform provided in the CSR message. Upon successful verification, the CA generates a certificate with the requested validity period and sends it to the web host (step ⑤).

There are several points worth mentioning to realize this scheme in reality, i.e., these points pose challenges to various stakeholders, such as:

- The underlying assumption of this mechanism is that actual validity periods for various website categories need to be a well-established common definition for uniform resolution on the Internet. This requires coordination in the web community at large, which requires leadership by an industry group such as CA/B Forum.
- The CA needs to support additional processing to verify and validate the signed category response. In this respect, there is a certain impact on the issuing CA.
- If this categorization service is controlled by a small number of large companies, this may possibly become the same situation as the current browser market, where a relatively small number of browser vendors dominate the large majority of browsers in use today.

As this scheme involves multiple stakeholders, their coordination is essential to realize it. This situation poses challenges similar to the evolution of OCSP-based solutions. In this sense, we acknowledge potential pitfalls in this approach. However, we consider this scheme represents a new approach to address the gap we discussed in Section 5.6.1, thus worth considering toward deriving validity period in a meaningful and uniform manner. Furthermore, we consider designing a concrete method to achieve a meaningful translation from categories to validity periods require further research, which is beyond the scope of this paper.[35]

[34] Furthermore, other attributes of websites may be used in place of or in addition to their categories in these steps.

5.6.4. *Possible actions by the industry and standard bodies*

The idea we presented in Sections 5.6.2 and 5.6.3 makes the fixed-duration short-term-certificate-based approach no longer viable due to the use of multiple different validity periods based on factors such as self-conducted risk assessment and website types. In this sense, this idea goes against the STAR approach and it may appear to bring the status of the revocation schemes back to the conventional CRL and OCSP-based mechanism. However, this situation is no different from the current state given that the STAR approach has no real-world traction to begin with, as we discussed in Section 4. In this sense, we believe that considering our idea is a step forward.

As we discussed in Section 5.3, browser vendors have already announced their intent to make the validity period progressively shorter from 398 to 90 to 45 days in the next few years. Keeping this situation in mind, it will be helpful to establish a *recommended best practice* on the validity period for future certificate use and to publish a guideline on this subject. To achieve this goal, collaboration among relevant organizations, including standardization bodies (e.g., IETF, ITU-T) and industries (browser vendors, CA communities, CA/B Forum) is beneficial for creating and endorsing this *best practice*. Doing so will help converge on the subject of revocation handling. Possible way forward is to consider the approach we discussed in Section 5.6.3. Several key points for this effort include: (1) maximizing the revocation agility by keeping the balance between timely revocation and the certificate issuance overhead, and (3) impacts on the CA and browser communities.

5.6.5. *Research on addressing device-dependent constraints*

As we discussed in Section 5.5, mobile and IoT/embedded device types tend to settle with less secure revocation check mechanism due to their unique constraints. They are also the first ones to be squeezed when the adoption of short-term certificates occurs as browser vendors move toward progressively shorter certificate validity periods as we discussed in Section 5.3. In this sence, these devices types are *weaker citizens* in the Internet. This is a concern; especially the use of mobile devices continue to grow in the general public. In this sense, research is needed to specifically address these resource-constrained device types and how to mitigate the impact on them. The possible outcome of such research may include new approaches to revocation checking and accommodating short-term certificates on such device types. To put it in a wider perspective, future research needs to address how to ensure end-user invisible-nonetheless-important functionalities, such as security mechanisms are not victimized in these resource-constrained device types for the benefit of maintaining usability and performance.

6. Conclusion

In this paper, we showed that the short-term certificate concept as exemplified in the STAR RFC has no real-world implementation after five years IETF issued it as the result of our analysis of more than 1.5 million certificates. Despite the near-100%-level support of OCSP and CT in web host certificates today, the absence of STAR implementation in practice indicates that the revocation-related measure has yet to resolve the revocation problem. At the same time, the growing trend set forth by browser vendors represents more practical but non-standardized solutions, and this trend indicates their increasing level of dominance in the direction of how *revocation management* is handled in reality. We

[35] The innovative part of our approach is the mechanism in which website categories are translated to validity periods in a meaningful manner. It may involve multiple factors to achieve well-reasoned mapping between them. We consider this to be more of a theoretical endeavor that requires a methodical approach. With respect to the scope of our paper, it is our future work.

discussed our insights and views behind the reasons for this current situation from multiple perspectives and suggested a potential way forward, including a new approach of validity period selection based on website categorization. Further research on establishing viability of our proposed approach in Section 5.6.3 and the research direction discussed in Section 5.6.5 are our next steps.

CRediT authorship contribution statement

Takahito Yoshizawa: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing; **Himanshu Agarwal:** Data curation, Software; **Dave Singelée:** Project administration, Supervision; **Bart Preneel:** Funding acquisition.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Flemish Government through the Cybersecurity Research Program with grant number: VOEWICS02.

References

Apple, I., 2018. Check Apple's Revocation System #52. https://github.com/crtsh/certwatch_db/issues/52.
Apple, I., 2023. About Upcoming Limits on Trusted Certificates. https://support.apple.com/en-us/102028.
Barnes, R., Hoffman-Andrews, J., McCarney, D., Kasten, J., 2019. Automatic Certificate Management Environment (ACME). RFC 8555. IETF.
Basile, B., Wood, C., 2017. Your apps and evolving network security standards. Worldwide Developers Conference (WWDC).
Berisha, B., Mëziu, E., Shabani, I., 2022. Big data analytics in cloud computing: an overview. J. Cloud Comput. 11 (1), 24.
Boureanu, I., Migault, D., Preda, S., Alamedine, H.A., Mishra, S., Fieau, F., Mannan, M., 2020. LURK: Server-controlled TLS delegation. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, pp. 182–193.
Bruhner, C.M., Linnarsson, O., Nemec, M., Arlitt, M., Carlsson, N., 2022. Changing of the guards: certificate and public key management on the internet. In: International Conference on Passive and Active Network Measurement. Springer, pp. 50–80.
Bruni, R., Bianchi, G., 2020. Website categorization: a formal approach and robustness analysis in the case of E-commerce detection. Expert Syst. Appl. 142, 113001.
Bureau, I.A. Content Taxonomy. https://www.iab.com/guidelines/content-taxonomy/.
Chromium, G. Certificate Lifetimes. https://chromium.googlesource.com/chromium/src/+/master/net/docs/certificate_lifetimes.md.
Chromium.org. CRLSets. https://www.chromium.org/Home/chromium-security/crlsets/.
Chuat, L., Abdou, A., Sasse, R., Sprenger, C., Basin, D., Perrig, A., 2020. SoK: delegation and revocation, the missing links in the Web's chain of trust. In: 2020 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, pp. 624–638.
Chung, T., Lok, J., Chandrasekaran, B., Choffnes, D., Levin, D., Maggs, B.M., Mislove, A., Rula, J., Sullivan, N., Wilson, C., 2018. Is the web ready for OCSP must-staple? In: Proceedings of the Internet Measurement Conference 2018, pp. 105–118.
Cohen, D., Naim, O., Toch, E., Ben-Gal, I., 2021. Website categorization via design attribute learning. Comput. Secur. 107, 102312.
Crone, M. . Protocols for Issuing and Managing Digital Certificates. https://max-crone.org/assets/docs/2020-certificate-management-protocols.pdf.
Dickinson, L., Smith, T., Seamons, K., 2019. Leveraging locality of reference for certificate revocation. In: Proceedings of the 35th Annual Computer Security Applications Conference, pp. 514–528.
Durumeric, Z., Li, F., Kasten, J., Amann, J., Beekman, J., Payer, M., Weaver, N., Adrian, D., Paxson, V., Bailey, M., et al., 2014. The matter of heartbleed. In: Proceedings of the 2014 Conference on Internet Measurement Conference, pp. 475–488.
Eastlake, 3rd, D., 2011. Transport Layer Security (TLS) Extensions: Extension Definitions. RFC 6066. IETF.
Encrypt, L. FAQ. https://letsencrypt.org/docs/faq/.
Facebook.com. Product Categories. https://developers.facebook.com/docs/marketing-api/catalog/guides/product-categories/.
Forum, C. CA/Browser Forum. https://cabforum.org/.
Google.com. Google_Product_Taxonomy_Version: 2021-09-21. https://www.google.com/basepages/producttype/taxonomy.en-US.txt.
Gutmann, P., 2002. PKI: It's not dead, just resting. Comput. (Long Beach Calif.) 35 (8), 41–49.
Gutmann, P., 2024. Engineering Security, Book Draft. https://www.cs.auckland.ac.nz/~pgut001/pubs/book.pdf.
Hallam-Baker, P., 2015. X.509v3 Transport Layer Security (TLS) Feature Extension. RFC 7633. IETF.
Helme, S., a. Revocation Is Broken. https://scotthelme.co.uk/revocation-is-broken/.
Helme, S. Top 1 Million Sites Security Analysis. https://crawler.ninja/.
Highspeedinternet.com. The Best Web Browsers of 2025. https://www.highspeedinternet.com/resources/best-web-browsers.
Housley, R., Ford, W., Polk, T., Solo, D., 1999. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459. IETF.
Ionos.com. Country Domains: A Comprehensive ccTLD List. https://www.ionos.com/digitalguide/domains/domain-extensions/cctlds-a-list-of-every-country-domain/.
(IRM), I. o. R.M., 2018. Standard Deviations – A Risk Practitioners Guide to ISO 31000 – 2018. Whitepaper. IRM. https://www.theirm.org/media/6884/irm-report-iso-31000-2018-v2.pdf.
(ISO), I. O. f.S., 2018. ISO 31000:2018 Risk Management – Guidelines, Edition 2. Standard. ISO.
(ISO), International Organization for Standardization and (ITU-T), I. T.U., 2019. Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks, ISO Standard 9594-8:2001, ITU-T Recommendation X.509. Technical Report ITU-R. International Telecommunication Union.
Kagan, S., Bekkerman, R., 2018. Predicting purchase behavior of website audiences. Int. J. Electron. Commer. 22 (4), 510–539.
Kaggle.com. Alexa Top 1 Million Sites. https://www.kaggle.com/datasets/cheedcheed/top1m.
Larisch, J., Choffnes, D., Levin, D., Maggs, B.M., Mislove, A., Wilson, C., 2017. CRLite: A scalable system for pushing all TLS revocations to all browsers. In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE, pp. 539–556.
Laurie, B., Langley, A., Kasper, E., 2013. Certificate Transparency. RFC 6962. IETF.
Laurie, B., Messeri, E., Stradling, R., 2021. Certificate Transparency Version 2.0. RFC 9162. IETF.
Liu, Y., Tome, W., Zhang, L., Choffnes, D., Levin, D., Maggs, B., Mislove, A., Schulman, A., Wilson, C., 2015. An end-to-end measurement of certificate revocation in the Web's PKI. In: Proceedings of the 2015 Internet Measurement Conference, pp. 183–196.
Ma, Z., Faulkenberry, A., Papastergiou, T., Durumeric, Z., Bailey, M.D., Keromytis, A.D., Monrose, F., Antonakakis, M., 2023. Stale TLS certificates: investigating precarious third-Party access to valid TLS keys. In: Proceedings of the 2023 ACM on Internet Measurement Conference, pp. 222–235.
Medium.com, 2022. Website categorization. https://medium.com/website-categorization/website-categorization-api-ca6c3e0f6c4d.
Microsoft. Enable Online OCSP/CRL Checks. https://admx.help/?Category=EdgeChromium&Policy=Microsoft.Policies.Edge::EnableOnlineRevocationChecks.
Mononen, T., Kause, T., Farrell, S., Adams, D.C., 2005. Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP). RFC 4210. IETF.
Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C., 1999. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560. IETF.
Namecheap.com. What Was The Green Bar and Where Did It Go? https://www.namecheap.com/blog/ssl-green-bar-disappearance/.
Nameshield.com, 2024. SSL/TLS Certificate Duration Reduced to 45 Days by 2027: Apple Takes the First Step. https://blog.nameshield.com/blog/2024/10/17/ssl-tls-certificate-duration-reduced-to-45-days-by-2027apple-takes-the-first-step/.
NIST, 2012. Guide for Conducting Risk Assessments, Information Security. SP 800-30. National Institute of Standards and Technology (NIST).
Pettersen, Y.N., 2013. The Transport Layer Security (TLS) Multiple Certificate Status Request Extension. RFC 6961. IETF.
Qualys. SSL Pulse. https://www.ssllabs.com/ssl-pulse/.
Register, T., 2024. Sysadmins Rage over Apple's 'nightmarish' SSL/TLS Cert Lifespan Cuts Plot. https://www.theregister.com/2024/10/15/apples_security_cert_lifespan/.
Roberts, R., Holly, M., Perez, L., 2021. Categorizing online climate change interpretation across the national park system. J. Interpret. Res. 26 (1), 6–23.
Safetydetectives.com. 5 Best & Fastest Web Browsers in 2025: Expert Ranked. https://www.safetydetectives.com/blog/best-fastest-web-browsers/.
Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C., 2013. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 6960. IETF.
Sheffer, Y., Lopez, D., de Dios, O.G., Perales, A. A.P., Fossati, T., 2020. Support for Short-Term, Automatically Renewed (STAR) Certificates in the Automated Certificate Management Environment (ACME). RFC 8739. IETF.
SLL.com, 2021. How Do Browsers Handle Revoked SSL/TLS Certificates? https://www.ssl.com/blogs/how-do-browsers-handle-revoked-ssl-tls-certificates/.
Smith, T., Dickinson, L., Seamons, K., 2020. Let'S revoke: scalable global certificate revocation. In: Network and Distributed Systems Security (NDSS) Symposium 2020.
Stark, E., Huang, L.-S., Israni, D., Jackson, C., Boneh, D., 2012. The case for prefetching and prevalidating TLS server certificates. In: NDSS. Vol. 12.
Statista.com, a. Chrome's Rise to Browser Dominance. https://www.statista.com/chart/1438/browser-market-share-since-2008/.
Statista.com. Global Market Share Held by Leading Internet Browsers from January 2012 to August 2024. https://www.statista.com/statistics/268254/market-share-of-internet-browsers-worldwide-since-2009/.

Statista.com. Number of Internet and Social Media Users Worldwide as of October 2024 (in billions). https://www.statista.com/statistics/617136/digital-population-worldwide/.
Techslang.com. Website Categorization: Definition, Applications, and Best Tools. https://www.techslang.com/website-categorization-definition-best-tools/.
Testssl.sh, Testing TLS/SSL Encryption. https://testssl.sh/.
Topalovic, E., Saeta, B., Huang, L.-S., Jackson, C., Boneh, D., Towards Short-Lived Certificates. https://www.ieee-security.org/TC/W2SP/2012/papers/w2sp12-final9.pdf.
Websitecategorizationapi.com. Instantly Classify Any Website with Powerful Website Categorization API. https://www.websitecategorizationapi.com/.
Wiki, M. CA/Revocation Checking in Firefox. https://wiki.mozilla.org/CA/Revocation_Checking_in_Firefox.
Wikipedia.org, a. Certificate Transparency. https://en.wikipedia.org/wiki/Certificate_Transparency.
Wikipedia.org. Comparison of Web Browsers. https://en.wikipedia.org/wiki/Comparison_of_web_browsers.