A first look at the CT landscape: Certificate transparency logs in practice

Josef Gustafsson, Gustaf Overier, Martin Arlitt and Niclas Carlsson

The original publication is available at www.springerlink.com:
Gustafsson, J., Overier, G., Arlitt, M., Carlsson, N., (2017), A first look at the CT landscape: Certificate transparency logs in practice, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, , 87-99. https://doi.org/10.1007/978-3-319-54328-4_7

Original publication available at:
https://doi.org/10.1007/978-3-319-54328-4_7

A First Look at the CT Landscape: Certificate Transparency Logs in Practice

Josef Gustafsson[†], Gustaf Overier[†], Martin Arlitt[‡], and Niklas Carlsson[†]

[†] Linköping University, Sweden
[‡] University of Calgary, Canada

Abstract. Many of today's web-based services rely heavily on secure end-to-end connections. The "trust" that these services require builds upon TLS/SSL. Unfortunately, TLS/SSL is highly vulnerable to compromised Certificate Authorities (CAs) and the certificates they generate. Certificate Transparency (CT) provides a way to monitor and audit certificates and certificate chains, to help improve the overall network security. Using an open standard, anybody can setup CT logs, monitors, and auditors. CT is already used by Google's Chrome browser for validation of Extended Validation (EV) certificates, Mozilla is drafting their own CT policies to be enforced, and public CT logs have proven valuable in identifying rogue certificates. In this paper we present the first large-scale characterization of the CT landscape. Our characterization uses both active and passive measurements and highlights similarities and differences in public CT logs, their usage, and the certificates they include. We also provide insights into how the certificates in these logs relate to the certificates and keys observed in regular web traffic.

1 Introduction

The internet today involves billions of devices and millions of services that require private or confidential communication. Unfortunately, it is unthinkable to trust that every entity on the internet is who they claim to be. Instead, protocols such as Transport Layer Security (TLS) and its predecessor Secure Sockets Layer (SSL) rely heavily on the trust in Certificate Authorities (CAs) [2].

With TLS/SSL, CAs are responsible for verifying the identity of entities and issuing electronic proof in the form of X.509 certificates. For example, in the case of HTTPS, a server or domain that wants to prove its identity typically pays a CA (or an organization that a CA has delegated trust to, using chained certificates) to create a signed certificate that connects its identity with a public key that others can use to securely communicate with the server/domain. If that CA's root certificate is available in the browser's root store, the browser can then use the root certificate to validate this certificate. Once validated, the browser trusts that the public key belongs to the claimed server/domain.

Conceptually, certificates enable a user to trust that a service provider they want to use is who they say they are. However, in practice, there are numerous issues that can undermine that trust, including human error, intentional fraud,

This is the authors' version of the paper that will appear in *Proc. Passive and Active Measurement Conference (PAM)*, Sydney, Australia, Mar. 2017. The final publication is available at Springer via http://dx.doi.org/[doi].

etc. [13]. Many of these issues stem from every CA having the power to issue certificates for any domain and that there are no mechanisms to inform the domain owners of issued certificates. This has resulted in many hard-to-detect incidents, including a recent incident where Symantec issued test certificates for 76 domains they did not own (including domains owned by Google) and another 2,458 unregistered domains [23].

To improve the situation, the use of Certificate Transparency (CT) [17] has been proposed and standardized through IETF. In fact, after the Symantec incident mentioned above, Google demanded that Symantec log all of their certificates in public CT logs. With CT, certificates should be published in public append-only logs, whose content is verified by monitors, and whose cryptographic integrity are verified by auditors. Any organization or individual can operate a monitor to verify these public records.

Google's Chrome browser was the first to enforce CT, with Chrome 41 and later requiring CT for Extended Validation (EV) certificates (issued after Jan. 1, 2015). Before displaying visual cues to the user that normally come with EV certificates, the certificate needs to be accompanied by Signed Certificate Timestamps (SCTs), where an SCT is a promise that the certificate is included in a public log. Chrome requires an EV certificate to be included in at least one Google operated log and one non-Google operated log [15]. The choice to start with EV certificates was motivated by the EV certificates themselves being intended to follow stricter issuing criteria than regular Domain Validated (DV) and Organization Validated (OV) certificates.[1] Mozilla is currently drafting their own CT policies (expected to require that certificates are present in logs operated by two organizations separate from the CA) and are on track to start enforcing CT for EV later 2017. Both Chrome and Mozilla are expected to enforce CT also for DV some time in the future.

Although CT is standardized [17] and used at large scale, it is not publically known how CT logs are used in practice. In this paper we present the first large-scale characterization of the CT landscape. First, we implemented a basic CT monitor [17] that actively monitors all public logs submitted to Chrome up to Dec. 2015 (3 Google operated and 7 CA operated) and one large log operated by NORDUnet.[2] Second, we characterize both differences in basic properties related to how different policies are implemented at the logs and properties related to the log content itself, including the certificates they include, their overlap in coverage, as well as temporal differences between the logs and their usage. Third, to glean some insight into how the certificates in these logs and their usage relate to that seen in regular web traffic, we also use the certificates observed across 232 million HTTPS sessions observed on a university network.

Our results highlight differences and similarities between the different logs. In general, there are significant differences in the certificates included in Google

[1] EV certificates were themselves introduced to address waning user trust.

[2] Technically, Google is also a CA. At the time of the measurements, no other production logs were known - only logs for testing purposes - although more production logs have appeared since. `https://www.certificate-transparency.org/known-logs`.

operated logs (that relies heavily on web crawls to identify certificates) and smaller CA operated logs. The coverage of the logs is broad. For example, for almost all domains observed in the university traces, there is at least one log with a valid DV certificate (despite such logging being voluntary for all CAs except Symantec), and for EV certification there are only small differences between the certificates that are included in Google logs and in CA operated logs.

The remainder of the paper is organized as follows. We first give a brief overview of CT (Section 2) and describe our collection methodology (Section 3). Next, we characterize the logs from the perspective of their properties alone (Section 4) and then based on the HTTPS traffic observed on campus (Section 5). Finally, related work (Section 6) and conclusions (Section 7) are presented.

2 Certificate Transparency

Certificate Transparency attempts to address flaws in the TLS/SSL certificate system [17, 18]. CT extends classic TLS/SSL operation with CT logs, auditors, monitors, as well as new communication interfaces between all these entities. With CT, each log maintains an append-only hash tree based on a binary Merkle Hash Tree [20] and newly issued certificates are appended to one or more CT log. The logs return a signed promise of inclusion, called an SCT, which is used by the TLS server to prove to clients that the certificate is logged.

Logs commit to publishing a Signed Tree Head (STH) within a fixed amount of time of issuing the SCT, called the Maximum Merge Delay (MMD). The STH can be used to prove both that a certain entry was included at a certain point in time and that the log maintains consistency over time (i.e., every new tree is a superset of every old tree). A log that cannot prove consistency between two STHs is likely to be distrusted immediately. In practice, the inclusion process can be broken into an update interval (UI) and the time to publish (TTP), where UI is the time between issuing an SCT and incorporating the corresponding entry into the STH and TTP is the time between signing and publishing STHs. In general, a CT log is itself considered compliant with regards to the MMD (offering an acceptably small attack window) if UI+TTP $<$ MMD.

Once the STH is published, monitors will have access to the certificate chain to detect any irregularities. A log can prove that a certain certificate has been included using an inclusion proof [17]. Auditors and monitors cooperate to ensure that logs are behaving correctly and that the log content corresponds to what the domain owners intended. In contrast to CAs, the CT logs are publicly auditable and enable anyone to verify claims of correctness. Furthermore, anyone can operate logs, monitors and auditors, making it infeasible for an adversary to control all instances.

3 Methodology and Datasets

For our data collection we implemented a basic CT monitor [17] in Python, which monitors the public logs and various domains, but that does not try to

Table 1. Basic properties of the CT logs.

Log name	Operated by	Submitted	URL	Roots	MMD	UI	TTP
Pilot	Google	2013-03-25	ct.googleapis.com/pilot	474	24 hr	1 hr	22 min
Aviator	Google	2013-09-30	ct.googleapis.com/aviator	474	24 hr	1 hr	22 min
Rocketeer	Google	2014-09-01	ct.googleapis.com/rocketeer	474	24 hr	30 m	34 min
Digicert	Digicert	2014-09-30	ct1.digicert-ct.com/log	57	24 hr	1 hr	12 hr
Izenpe	Izenpe	2014-11-10	ct.izenpe.com	40	24 hr	1 min	< 1 min
Certly	Certly	2014-12-14	log.certly.io	183	24 hr	10 min	< 1 min
Symantec	Symantec	2015-05-01	ct.ws.symantec.com	19	24 hr	6 hr	< 1 min
Venafi	Venafi	2015-06-11	ctlog.api.venafi.com	357	24 hr	2 hr	3 min
WoSign	WoSign	2015-09-22	ct.wosign.com	12	24 hr	1 min	< 1 min
Vega	Symantec	2015-11-13	vega.ws.symantec.com	19	24 hr	6 hr	< 1 min
Plausible	NORDUnet	Not Subm.	plausible.ct.nordu.net	442	24 hr*	12 min	2 min

*Plausible operates with an unofficial MMD of 24hr.

determine the legitimacy of the certificates. For the purpose of our study, we collected all certificates that have been added to eleven CT logs: the ten public logs submitted to Chrome (3 operated by Google and 7 CA operated logs) at the time of our last measurement (Dec. 2015) and one (non-production) log operated by NORDUnet. We recorded all fields of the individual certificates and validated the certificates against the Mozilla root store, as observed on Dec. 1, 2015.

Furthermore, to understand how representative the observed certificates of the different logs are compared with what a typical internet user sees, we also use a one-week long complementary dataset collected by passively monitoring the Internet traffic to/from the University of Calgary, Canada [22]. Using Bro, we log specific information about the non-encrypted part of the TLS/SSL handshake, including all digital certificates sent. This dataset was collected Oct. 11-17, 2015, and covers 232 million HTTPS sessions, 67,644 unique certificates, and 552 million certificate exchanges. For most of our analysis we focus on the CT logs, and use the university dataset as a reference point.

4 Analysis of Logs

4.1 Basic log properties and operational measures

Table 1 summarizes the basic properties of the eleven logs we used. The logs are ordered based on when they were submitted to Chrome (second column). All logs allow HTTPS to be used when accessing the logs. Furthermore, all logs except Venafi (who uses RSA with SHA-256) use ECDSA (over the NIST P-256 curve) to sign data structures (STHs and SCTs). Both techniques are recommended in RFC6962 [17] and are expected to provide roughly the same security.

The last four columns indicate large differences in how the logs are implemented and maintained. The *roots* column shows the number of accepted certificate-chain roots for the logs. We used the APIs provided by the CT logs to download all roots accepted by each log. Out of the 503 unique roots we observed across all logs, the three Google logs included 474 in their root store. In contrast, the CA operated logs typically included much fewer roots. For example, the two Symantec logs (Symantec and Vega) and the WoSign log only allowed certificates signed by 19 and 12 of the roots, respectively. These observations point to differing usage patterns. Based on the Google CT policy, for example, CAs may be incentivized to log any certificates they issue themselves, but there is little incentive for them to log certificates issued by competitors. In contrast, browser vendors may prefer to log at least the certificates accepted by the browser.

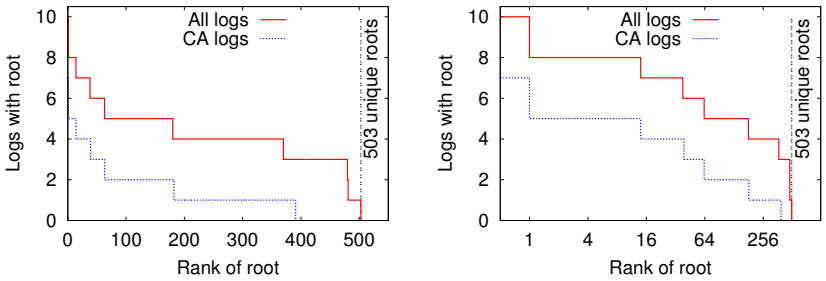


Fig. 1. Number of logs accepting each root for submitted entries.

With browsers increasingly requiring certificates to be found in multiple logs, many roots are starting to be included in several logs. Figure 1 shows the number of logs that include each root. In general, we have found that roughly 10% of the roots are included in six or more of the logs' root stores, and most of the roots are included in 3-5 of the root stores. Again, the three Google operated logs include the majority of the observed roots.

The last two columns provide insights into the time granularity at which the logs operate and how well the MMD is satisfied. First, referring to Section 2, remember that UI+TTP must be less than the MMD for the log to be considered compliant. In general, the (load dependent) UIs are substantially smaller than the 24-hour MMDs, suggesting that all logs typically require much less time to merge the certificate chain than the upper bound. However, the UIs differ substantially between logs. For example, the median UI observed in Table 1 varies from minute scale (e.g., Izenpe and WoSign) to hours (e.g., the Symantec and Google logs). In fact, on Oct. 16, 2016, the Aviator log (Google operated) overshot its MMD by 2.2 hours. As a result, since Dec. 1, 2016, the log has been frozen and is no longer accepting new submissions.[3] This is a form of "soft untrusting" as old SCTs issued by Aviator are still honored. The incident has sparked a debate on if the policy needs to be updated. In general, a shorter interval can be convenient for both operators and clients, as it reduces the size of each merge and reduces the time until clients can request inclusion proofs.

The TTPs also differ substantially between logs. The notable outlier is Digicert with a 12-hour delay between signing and publishing STHs. When we asked, Digicert said that they sign STHs every hour, but use the extra delay for synchronizing between servers located in multiple datacenters. All other logs publish STHs within 1 hour, although some have much shorter TTPs. While Table 1 reports median values, UIs and TTPs were relatively stable with small variations over the time we monitored the logs (up to Dec. 2015). The spike in UI that Aviator saw on Oct. 16, 2016, shows that there since have been larger variations.

4.2 Certificate analysis

CT logs can be a valuable tool for monitoring newly issued certificates. For example, we can examine the strength of the encryption algorithms used, as

[3] `https://bugs.chromium.org/p/chromium/issues/detail?id=389514`

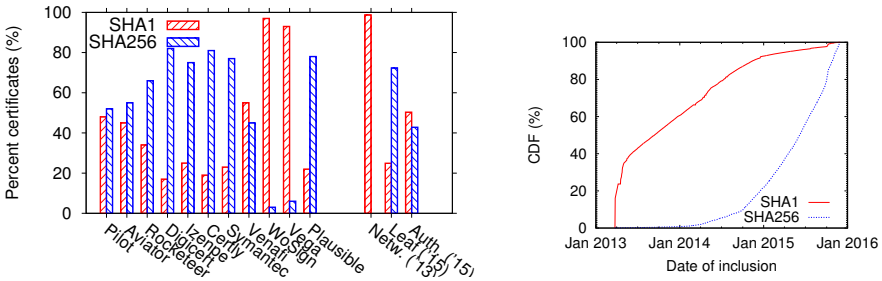
Table 2. Distribution of certificate validation types and signature hashes.

			Validation			Encryption algorithm			
Log name	**Operated by**	**Entries**	**DV**	**OV**	**EV**	**RSA (1024)**	**RSA (2048)**	**RSA (4096)**	**EC (256)**
Pilot	Google	10,831,024	87%	8%	5%	2%	79%	3%	16%
Aviator	Google	10,069,865	87%	8%	5%	1%	78%	3%	17%
Rocketeer	Google	8,140,991	87%	8%	5%	1%	75%	4%	21%
Digicert	Digicert	229,858	18%	5%	78%	0%	96%	3%	0%
Izenpe	Izenpe	65,812	31%	1%	68%	0%	95%	5%	0%
Certly	Certly	161,740	36%	3%	61%	0%	94%	5%	0%
Symantec	Symantec	11,3674	21%	5%	74%	0%	97%	2%	0%
Venafi	Venafi	4,626	85%	10%	5%	0%	93%	5%	1%
WoSign	WoSign	11,188	97%	1%	2%	0%	99%	1%	0%
Vega	Symantec	80	95%	0%	5%	0%	95%	0%	2%
Plausible	NORDUnet	5,893,906	88%	7%	5%	3%	90%	3%	4%

well as detect CAs that backdate certificates to circumvent restrictions. To gain insight into the differences in the certificates logged by the different CT logs, Table 2 shows a breakdown of the different certificate entries of each log.

In general, the logs can be divided into three size-based groups: (i) large logs with more than $5,000,000$ entries, (ii) medium-sized logs with $50,000-1,000,000$ entries, and (iii) small logs with less than $50,000$ entries. We observed significant differences in the types of certificates being stored in each log category. Columns 2-4 in Table 2 show a breakdown between EV, DV, and OV certificates. The large difference in the ratio between DV and EV certificates observed for the four large logs (Pilot, Aviator, Rocketereer, and Plausible; each with 5% EV certificates) and the top-four CA operated logs (Digicert, Izenpe, Certly, and Symantec; all in the 61-78% range) can be explained by the relative log sizes and differences in how the certificates are submitted. While the Google logs and Plausible have been populated by crawling the internet and submitting encountered certificates (capturing all types of certificates, including certificates of domains that may not themselves have chosen to participate in CT), it appears that Digicert, Izenpe, Certly, and Symantec primarily use the logs to store entries with the intent of using the SCTs in EV validation. The focus on EV certificates of both Digicert and Symantec is also visible in the university dataset, where these two CAs are responsible for 27.6% and 56.2% of the EV sessions (and a combined 37.9% of the unique EV certificates). However, in absolute numbers, the four large logs all include more EV certificates than the CA logs. We also note that the fraction of EV certificates observed in the three Google operated logs and Plausible are similar to the fractions observed in the wild. For example, in our university dataset EVs are observed in 4.9% of the observed leaf certificates and 6.3% of all sessions. The small logs (Venafi, WoSign, and Vega) are younger logs that at the time of the measurements still contained a large fraction of test entries, rather than entries intended for CT. These logs therefore have substantially different properties than the other categories.

In general, the logging of other certificates than EV certificates can be used for testing and to preserve public records of certificates. The use of public logs provides the true owners of domains (or monitors) a much easier means to identify rogue certificates than having to search the web, especially since many rogue certificates may not be reachable from the internet. This has proven valuable in identifying certificates violating regulations, including improper certificates from



(a) All logs (Dec. 1, 2015) (b) Pilot log over time

Fig. 2. Signatures used for certificates.

both Comodo[4] and Symantec[5]. Finally, we note that the certificate ratios of the CA operated logs are expected to change as browsers start to require logging of DV and OV certificates too.

In general, most logged certificates we observe use strong algorithms, with the majority of certificates in all logs using RSA with 2048 bit keys ($\geq$75%). Columns 5-8 in Table 2 break down the distribution of algorithms used for the certificates in each log. In addition to RSA keys (of different lengths), we note that the three Google logs include a significant number (16-21%) of certificates using Elliptic Curve (EC) signatures.

However, the logs also capture many certificates with weak keys or signatures. First, despite that NIST recommended to stop using 1024-bit RSA keys in 2013 [4], before the first entries of any of the CT log, we observed a non-negligible use of such short keys in the logs that use crawling of the web to fill their records. All these four logs include 1-3% such entries. This is consistent with the 1.3% authority and 5.6% leaf certificates we observed on campus [22].

Second, despite that the SHA1 hash algorithm is susceptible to known attacks and CAs no longer sign new certificates with SHA1, SHA1 is observed in 17$-$97% of signatures across the logs. Figure 2(a) breaks down the use of SHA1 and SHA256 across the logs. As reference points we also include values by Durumeric et al. [9] (Aug. 2013) and the university dataset (Oct. 2015) [22]. We note that most of the logs have numbers in-between those observed in the wild in 2013 and 2015, and that Plausible has a smaller fraction SHA1 usage than the three older Google logs. Given the append-only properties of these logs, this is to-be expected and supports observations that there is a reduction of SHA1 usage for new certificates. To understand the shift, Figure 2(b) shows the cumulative distribution function (CDF) of all SHA1 and SHA256 certificates inserted as a function of time for the oldest and largest log (Pilot). As expected, the SHA1 inclusion rate is steadily decreasing, while the SHA256 rate is steadily increasing. Again, the newer logs (with fewer entries) stick out with a large fraction SHA1 certificates. These certificates have been added relatively recently and include a large fraction weaker self-signed SHA1 test certificates from Google CT.

[4] `https://cabforum.org/pipermail/public/2015-November/006226.html`

[5] `https://security.googleblog.com/2015/10/sustaining-digital-certificate-security.html`

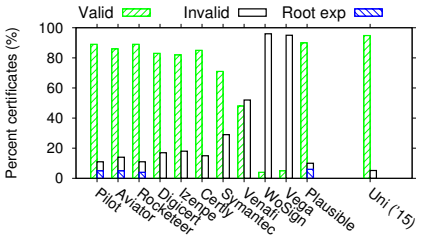


Fig. 3. Validation tests using the Mozilla root store.

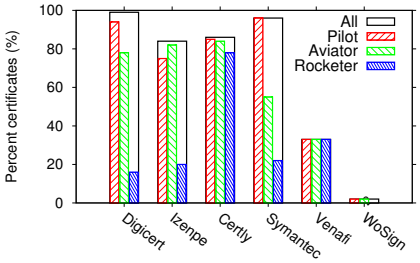


Fig. 4. Percent of entries in CA operated logs seen in at least one Google log.

One explanation that the outphasing of SHA1 is taking a long time is that many service providers, including Facebook and Twitter, are concerned that millions of users with older devices would lose access to their services and therefore want to delay the outphasing of SHA1[6]. With Facebook and Twitter only being responsible for 287 and 9 of the 250,000 most recently logged SHA1 certificates in the Pilot log, many other service providers also appear to be stalling.

As mentioned, the small logs (Venafi, WoSign, and Vega) have quite different key strengths properties than the other logs. These logs stick out even more when looking at the validity of the certificates in the logs. Figure 3 shows the percent of the certificates in each log that validate using the Mozilla root store. The large fraction of invalid certificates is again explained by a relatively large fraction of test certificates. For these logs almost none of the invalid certificates are due to expired roots. In contrast, for the other logs about half of the invalid certificates are due to expired roots. However, despite all logs being append-only and certificates eventually expiring, most of the observed certificates for the other logs are still valid. Furthermore, we again observe similar characteristics for the large crawled logs (86-90% still valid certificates) and campus (94.8% as measured by the fraction of HTTPS sessions that had a valid certificate).

4.3 Cross-log publication

To improve security and increase assurance, several SCTs can be used when validating certificates. For example, to pass Chrome's CT checks, an EV certificate must be accompanied by multiple valid SCTs: one operated by Google, one by another operator, and in some cases (depending on the validity period of the certificate) additional SCTs [15]. While Mozilla currently is drafting their own CT policies, it appears that their requirement of at least two independent logs will be similar in flavor to the policy applied by Chrome.

Motivated by Chrome's policy, we considered what fraction of the certificates in the six CA operated logs with at least 10,000 entries was included in at least one Google operated log. Figure 4 shows that at least 80% of the entries in each of the four large CA logs (Digicert, Izenpe, Certly, and Symantec) also are included in at least one of the three Google operated logs. Again, it appears that

[6] `https://blog.cloudflare.com/sha-1-deprecation-no-browser-left-behind/`

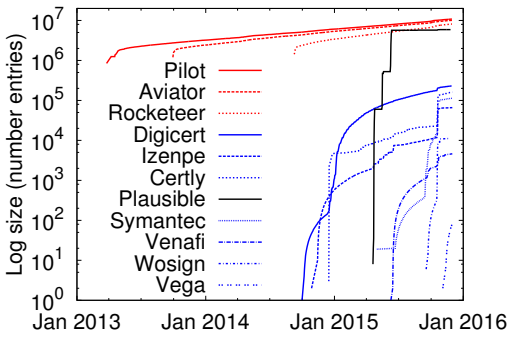


Fig. 5. Number of entries submitted to the logs over time.

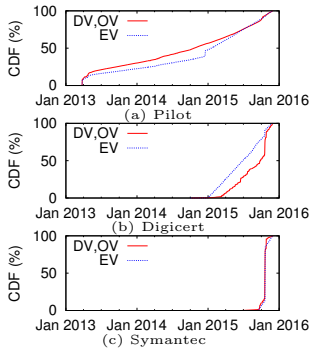


Fig. 6. Submissions of certificates for three example logs.

the remaining two smaller logs (Venafi and WoSign) contain a larger fraction of test certificates. This is expected to change when they become more mature.

The use of the Google logs also varies among the certificates in the top-four CA logs. For example, Certly certificates appear to be submitted to all three logs, whereas the certificates of the other three (Digizert, Izenpe, and Symantec) primarily are submitted to Pilot and Aviator. Part of the bias towards Pilot may be due to it being the first public log and rich-get-richer effects.

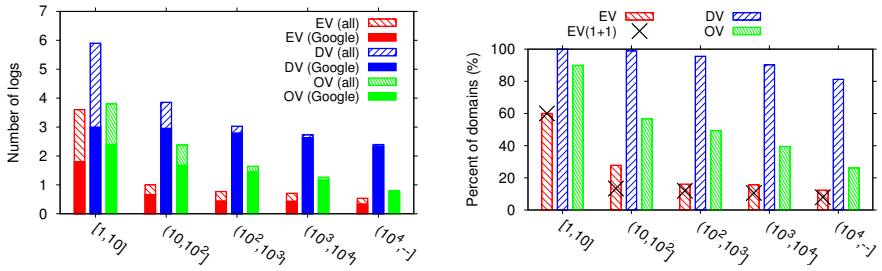
4.4 Temporal analysis

All CT logs are strictly append-only. Figure 5 shows the number of certificate entries (logarithmic scale) as a function of time for the different logs. To tie with the above discussion, we order the logs based on their start dates. While the Google logs (red curves) have a strict size ordering, the size-order changes over time among the CA operated logs (blue). The generally increasing growth rates can be explained by increasing use of short-lived certificates and general use of HTTPS. Some of the spikes can be explained by bulk registrations of certificates and the advent of enforcing CT for DV certificates.

Among the crawl-based logs we have observed steady inclusion rates of DV and OV certificates (e.g., Figure 6(a)), whereas the inclusion rates of EV certificates have been increasing. This suggests a relative increase in the use of EV certificates in the wild, but may also be affected by how Google extracts certificates. We also observe a significant peak in additions around Jan. 1, 2015, when Chrome's EV policy took effect. This is also around the time that Digicert (Figure 6(b)) started its log. Since then, Digicert have added EV certificates at a fairly steady rate. We also include results for Symantec (Figure 6(c)) as an example where the insertion rates of EV and DV certificates goes hand-in-hand. Again, Google requires Symantec to log all their certificates; not only EVs.

5 Popularity-based Analysis

We next look at the certificates of the domains associated with the HTTPS sessions on campus. For this analysis we extract the domain name associated



(a) Number of logs (b) Percent domains

Fig. 7. Average number of public logs that domains with different popularity occur in.

with each HTTPS session and map them to the certificates observed in the public logs (excluding Plausible). Furthermore, we rank each domain from most popular to least popular and report statistics for domains of different popularity.

Figure 7(a) shows the average number of logs (broken down into Google and non-Google logs) that domains in each popularity category observed (each category given a logarithmic-sized bucket of popularity ranks). The top-10 domains (google.com, apple.com, facebook.com, icloud.com, live.com, fbcdn.net, akamaihd.net, gstatic.com, microsoft.com, doubleclick.net) are observed in more logs than the less popular domains. The difference is largest for the EV certificates, although we see a decrease also for the other types. On average the EV certificates of this top category are observed in 3.5 logs, while DV and OV certificates are seen in 6 and 4 logs, respectively. In general, however, the CA logs have much worse coverage of the less popular domains. Perhaps more encouraging is that the Google logs include DV certificates for almost all domains (regardless of popularity). The total coverage is shown in Figure 7(b). The fraction of domains that have valid EV (or OV) certificates inserted in at least one log is smaller, and sharply decreasing with the domains popularity. We also note that the fraction of domains that satisfies Chrome's 1+1 requirement is even less. This is indicated by the $\times$ markers.

When interpreting the above results, note that the top-10 are responsible for 39% of the sessions and the top-100 for 75% (36% if not including the top-10). This shows that the average session is more likely to be to a domain included in at least one log than if considering a random domain from across all popularities) and that the more popular domains may be more willing to pay the extra cost of EV certificates. It will be interesting to see how websites will adopt if and when browsers start applying stricter CT policies also for non-EV certificates.

6 Related Work

Certificate Transparency (CT) is a fairly new topic. Measurement-based research has instead often focused on the TLS/SSL landscape with CT excluded and only commented that it may significantly change the landscape. Interesting example studies include work that have studied the trust graphs in the HTTPS ecosystem [2], identified occurrences of man-in-the-middle attacks on Facebook [13],

considered the trustworthiness of CAs and the countries they represent [10], and identified SSL error codes and their reasons [1].

CT is not the only attempt to reinforce the CA-based authentication system of TLS/SSL. Most approaches try to reduce the reliance on the trust of the CAs. This includes client-centric approaches that try to bypass the CAs during the certificate validation process [24], approaches that leverage the existing DNS infrastructure to limit the trust in CAs [11,12], and log-based approaches [5,14]. Log-based approaches have also been used to provide key distribution in other contexts [19], and to provide transparency for other data than X.509 certificates [26]. In contrast to CT, these other approaches have seen little adoption.

Other researchers have characterized certificate revocation [25] and developed hybrid techniques for certificate revocation that use transparency logs [16] to resolve some of the problems with current techniques [8]. In this article, we also briefly refer to studies that have examined attacks targeting particular aspects of the TLS/SSL connection establishment [3,7], when discussing the characteristics of the certificates themselves and the included public keys.

7 Conclusions

This paper presents the first large-scale characterization of the CT landscape. Using both active measurements obtained with a basic CT monitor and passively collected measurements in a university network, we characterize eleven CT logs and highlight similarities and differences across multiple dimensions. We find significant differences in the selection of root stores and how new certificates are added. For example, Google operated logs use large root stores and add certificates primarily through crawling, resulting in these logs including broad categories of certificates. The certificates in these crawl-based logs are more representative of the web traffic that browsers may see (e.g., on campus) than the certificates in the CA operated logs are. In general, the crawl-based logs have greater diversity in the types of certificates observed, are much larger, and include many certificates with weak keys or hashes. Analysis of the large CA operated logs and cross-log submissions suggest that CAs try to comply to Chrome's EV certificate policy, but that the submission rates of DV certificates have differed over time between CAs. In addition, by comparing with the certificates, keys, and domains observed in 232 million HTTPS sessions on a university network, we demonstrate how the coverage of the crawled logs captures the certificates observed during typical internet usage and that popular domains appear to be more willing to pay the extra cost of EV certificates. Future work could try to intercept the exchange of SCTs, so to also capture the potential validation that clients could do directly with the CT logs or the additional protection against partitioning that gossiping [6,21] and client-to-client communication may offer.

Acknowledgements: The authors are thankful to our shepherd Ralph Holz and the anonymous reviewers for their feedback. This work was funded in part by the Swedish Research Council (VR) and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

1. D. Akhawe, B. Amann, M. Vallentin, and R. Sommer. Here's my cert, so trust me, maybe?: understanding TLS errors on the web. In *Proc. WWW*, 2013.
2. B. Amann, R. Sommer, M. Vallentin, and S. Hall. No attack necessary: the surprising dynamics of SSL trust relationships. In *Proc. ACSAC*, 2013.
3. B. Beurdouche et al. A messy state of the union: Taming the composite state machines of TLS. In *Proc. IEEE S&P*, 2015.
4. E. Barker, W. Barker, W. P. W. Burr, and M. Smid. Recommendation for key management, part 1: General (rev. 3). *NIST Special Publication 800-57*, 2012.
5. D. Basin, C. Cremers, T. H.-J. Kim, A. Perrig, R. Sasse, and P. Szalachowski. Arpki: Attack resilient public-key infrastructure. In *Proc. ACM CCS*, 2014.
6. L. Chuat, P. Szalachowski, A. Perrig, B. Laurie, and E. Messeri. Efficient gossip protocols for verifying the consistency of certificate logs. In *Proc. IEEE CNS*, 2015.
7. D. Adrian et al. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *Proc. ACM CCS*, 2015.
8. R. Duncan. How certificate revocation (doesn't) work in practice, 2013.
9. Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman. Analysis of the HTTPS certificate ecosystem. In *Proc. IMC*, 2013.
10. T. Fadai, S. Schrittwieser, P. Kieseberg, and M. Mulazzani. Trust me, I'm a root CA! analyzing SSL root CAs in modern browsers and operating systems. In *Proc. ARES*, 2015.
11. P. Hallam-Baker and R. Stradling. *RFC6844: DNS Certification Authority Authorization (CAA) Resource Record*. IETF, 2013.
12. P. Hoffman and J. Schlyter. *RFC6698: The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*. IETF, 2012.
13. L. Huang, A. Rice, E. Ellingsen, and C. Jackson. Analyzing forged SSL certificates in the wild. In *Proc. IEEE S&P*, 2014.
14. T. H.-J. Kim, L.-S. Huang, A. Perrig, C. Jackson, and V. Gligor. Accountable key infrastructure (AKI): A proposal for a public-key validation infrastructure. In *Proc. WWW*, 2013.
15. B. Laurie. Improving the security of EV certificates, 2015.
16. B. Laurie and E. Käsper. Revocation transparency. *Google Research, Sept.*, 2012.
17. B. Laurie, A. Langley, and E. Käsper. *RFC6962: Certificate Transparency*. IETF, 2013.
18. B. Laurie, A. Langley, E. Käsper, E. Messeri, , and R. Stradling. *RFC6962-bis: Certificate Transparency draft-ietf-trans-rfc6962-bis-10*. IETF, 2015.
19. M. Melara, A. Blankstein, J. Bonneau, E. Felten, and M. Freedman. Coniks: Bringing key transparency to end users. In *Proc. USENIX Security*, 2015.
20. R. Merkle. *Merkle Tree Patent, US4309569A*, 1979.
21. L. Nordberg, D. K. Gillmor, and T. Ritter. *Gossiping in CT*. IETF, 2015.
22. G. Ouvrier, M. Laterman, M. Arlitt, and N. Carlsson. Characterizing the HTTPS trust landscape: A passive view from the edge. Tech. report, 2016.
23. R. Sleevi. Sustaining digital certificate security, Google Security Blog. https://security.googleblog.com/2015/10/sustaining-digital-certificate-security.html, Oct. 28 2015.
24. D. Wendlandt, D. G. Andersen, and A. Perrig. Perspectives: Improving SSH-style host authentication with multi-path probing. In *Proc. USENIX ATC*, 2008.
25. Y. Liu et al. An end-to-end measurement of certificate revocation in the web's PKI. In *Proc. IMC*, 2015.
26. D. Zhang, D. K. Gillmor, D. He, and B. Sarikaya. *CT for Binary Codes*. IETF, 2015.