

# Study of the SSL certificates and their revocation model

*Eirini Aikaterini Degkleri*

Thesis submitted in partial fulfillment of the requirements for the  
*Masters' of Science degree in Computer Science and Engineering*

University of Crete  
School of Sciences and Engineering  
Computer Science Department  
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Assistant Prof. *Evangelos P. Markatos*

---

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).



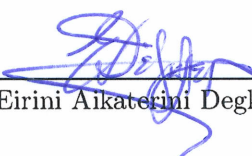
UNIVERSITY OF CRETE  
COMPUTER SCIENCE DEPARTMENT

**Study of the SSL certificates and their revocation model**

Thesis submitted by  
**Eirini Aikaterini Degkleri**  
in partial fulfillment of the requirements for the  
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author:

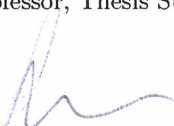
  
\_\_\_\_\_

Eirini Aikaterini Degkleri

Committee approvals:

  
\_\_\_\_\_

Evangelos P. Markatos  
Professor, Thesis Supervisor

  
\_\_\_\_\_

Sotiris Ioannidis  
Research director, Committee Member

  
\_\_\_\_\_

Panagiota Fatourou  
Associate professor, Committee Member

Departmental approval:

  
\_\_\_\_\_

Antonios Argyros  
Professor, Director of Graduate Studies

Heraklion, February 2018



# Study of the SSL certificates and their revocation model

## Abstract

The information shared over the Internet today is enormous, very personal and thus it must be secured. Several studies have pointed out the simplicity of HTTP session hijacking attacks and this stresses the fact that encrypted end to end communication is not a necessity only for websites with economic transactions. HTTP over SSL (HTTPS) is evolving and in 2017 it has reached the point where it is becoming the norm rather than the exception. TLS protocol, the successor of SSL, has room for improvement and so do the SSL/TLS certificates, which are used to secure and authenticate trusted entities.

An SSL or PKIX certificate binds a cryptographic key to a certain subject. It has a predefined validity period, during which it is considered trusted unless it is revoked. To attest its validity it is issued by a Certification Authority (CA), which is a trusted third party. One certificate may secure one or many entities, under a validation process which is performed by the CA. Moreover, to sign this information, the issuer uses a signature algorithm, that is computed by the browser.

This thesis is a measurement study that aims to shed light on the ecosystem of SSL certificates so that the reader can have an overall perspective on how they are adopted. Initially we analyze their basic components and attempt to indicate correlations and trends, and consequently, we discuss interesting cases within the data and possible correlations of certificates with high traffic sites' maintenance and with known attacks. To this end, Certificate Transparency's public data set, along with Alexa's top sites and Hackmagedon statistics on cyber attacks are used.

Additionally, the trust model around different aspects of the SSL certificates is closely examined. First, after reviewing known weaknesses, we explore cases where certificates were used as a mean to conceal rogue behavior and last we show where certification authorities fail to correctly validate secured entities. Furthermore, this study focuses on revocation and measures the trends around it and emphasizes on its the importance of revocation, by demonstrating known cases of attacks, which were due to the negligence of status checking. Additionally, since the main reason that revocation checking mechanisms fail is due to the related protocols applied, we take a step further to analyze and compare existing solutions and newly introduced promising protocols.

As a prime to future work, we contemplate whether the PKIX infrastructure is suitable to support the vast network of the Internet of Things, which is comprised of embedded devices with limited computational capabilities. SSL/TLS protocol proves to be burdensome in its traditional state, so we discuss less demanding protocols and variations tailored to their infrastructure.



# Μελέτη των ψηφιακών πιστοποιητικών πρωτοκόλλου SSL και του μοντέλου ανάκλησής τους

## Περίληψη

Η διαθέσιμη πληροφορία στο διαδίκτυο έχει αυξηθεί με τεράστιους ρυθμούς τόσο σε όγκο όσο και σε πολυπλοκότητα και η τάση αυτή προβλέπεται να συνεχιστεί. Επιπρόσθετα, τα διαμοιραζόμενα δεδομένα αφορούν ευαίσθητες πληροφορίες και ως αποτέλεσμα η διασφάλισή τους είναι απαραίτητη. Στο βασικό πρωτόκολλο μεταφοράς υπερκειμένου Hypertext Transfer Protocol (HTTP) η πληροφορία είναι σε μορφή απλού κειμένου κατά την ανταλλαγή μηνυμάτων, με αποτέλεσμα οποιοσδήποτε να μπορεί να την υποκλέψει. Επίσης, πολλαπλές μελέτες έχουν δείξει την ευκολία μίας επίθεσης για τον έλεγχο μιας συνεδρίας HTTP. Οι παραπάνω λόγοι εντείνουν την ανάγκη για την ύπαρξη κρυπτογραφημένης επικοινωνίας από την πηγή στον προορισμό (E2EE: End-to-End Encryption), όχι μόνο σε ιστοσελίδες στις οποίες πραγματοποιούνται οικονομικές συναλλαγές, αλλά καθολικά. Το πρωτόκολλο HTTPS (HTTP over SSL) αναπτύσσεται για να καλύπτει τις προαναφερθείσες ανάγκες και πλέον το 2017 έχει φτάσει στο σημείο που γίνεται κανόνας και όχι η εξαίρεση.

Ένα ψηφιακό πιστοποιητικό πρωτοκόλλου Secure Sockets Layer (SSL)/Transport Layer Security (TLS) βασίζεται στην κρυπτογραφία δημοσίου κλειδιού Public Key Infrastructure (PKI), ώστε να συνδέσει ένα κρυπτογραφικό κλειδί με την οντότητα που διασφαλίζεται. Εκδίδεται από κάποια αρχή πιστοποίησης - Certification Authority (CA), που θεωρείται «έμπιστο τρίτο μέρος», η οποία πραγματοποιεί μία διαδικασία επικύρωσης. Επιπρόσθετα, έχει μία προδιαγεγραμμένη περίοδο ισχύος, με συγκεκριμένη αρχή και τέλος κατά την οποία θεωρείται έγκυρο, εκτός και αν ανακληθεί. Τέλος, για να την εξακριβώσει της γνησιότητας τους οι πληροφορίες αυτές υπογράφονται με έναν κρυπτογραφικό αλγόριθμο, το αποτέλεσμα του οποίου υπολογίζεται από το πρόγραμμα περιήγησης.

Η μεταπτυχιακή αυτή διατριβή, έχει ως σκοπό της τη διερεύνηση του οικοσυστήματος των ψηφιακών πιστοποιητικών πρωτοκόλλου SSL, αναλύοντας τις βασικές τους συνιστώσες και επιχειρώντας να αναδείξει συσχετισμούς και τάσεις που συνδέονται με αυτά, ώστε ο αναγνώστης να αποκτήσει μία συνολική εικόνα της υιοθέτησής τους. Στη συνέχεια, παρουσιάζονται ενδιαφέρουσες περιπτώσεις που προκύπτουν από τα δεδομένα και συζητούνται σε βάθος πιθανές συσχετίσεις τους με ιστοσελίδες με μεγάλη επισκεψιμότητα. Επιπλέον, ελέγχεται η υπόθεση μας ότι οι ιστοσελίδες αυτές εφαρμόζουν συστηματικότερα τις βέλτιστες πρακτικές ασφάλειας. Για την ανάλυση αυτή χρησιμοποιούνται τα δημοσίως διαθέσιμα σύνολα δεδομένων του Certificate Transparency, το οποίο στοχεύει στην διαφάνεια της εξάπλωσης των ψηφιακών πιστοποιητικών καθώς και του Alexa που διατηρεί τη σειρά κατάταξης ιστοσελίδων βάσει της επισκεψιμότητας τους και των στατιστικών για κυβερνοεπιθέσεις, από το Hackmaggedon.

Κατόπιν, εξετάζεται το μοντέλο εμπιστοσύνης γύρω από τις διάφορες πτυχές των ψηφιακών πιστοποιητικών πρωτοκόλλου SSL. Ορμώμενοι από γνωστές αδυναμίες του

πρωτοκόλλου SSL και του διαδόχου του TLS, αναφερόμαστε σε περιπτώσεις όπου τα ψηφιακά πιστοποιητικά έχουν χρησιμοποιηθεί για να καλύψουν κακόβουλη συμπεριφορά. Συμπληρωματικά αναφερόμαστε σε περιπτώσεις όπου οι αρχές πιστοποίησης φαίνονται ανεπαρκείς στη διαδικασία της επικύρωσης των οντοτήτων για τις οποίες εγγυώνται. Στη συνέχεια, γίνεται ιδιαίτερη μνεία στους μηχανισμούς ανάκλησης των πιστοποιητικών, υπογραμμίζοντας τη σημασία τους δείχνοντας συσχετίσεις και στατιστικά γύρω από αυτούς και αναφέροντας γνωστές επιθέσεις που οφείλονται στην αμέλεια της σωστής εφαρμογής του ελέγχου των μηχανισμών αυτών. Καθώς το φαινόμενο αυτό συνδέεται άμεσα με τις αδυναμίες των υπάρχουσών λύσεων για τον έλεγχο της κατάστασης ισχύος ή την ανάκληση των πιστοποιητικών, συγκρίνουμε τα πιο διαδεδομένα πρωτόκολλα και κάποιες πολλά υποσχόμενες, πρόσφατα προταθείσες λύσεις.

Σαν προοίμιο μελλοντικής ενασχόλησης και έρευνας, συλλογιζόμαστε εάν η υποδομή του δημόσιου κλειδιού X.509 είναι ικανή να υποστηρίξει το εξαιρετικά μεγάλο μέγεθος δίκτυο των διασυνδεδεμένων συσκευών που απαρτίζουν το “διαδίκτυο των πραγμάτων” - Internet of Things (IoT), το οποίο αποτελείται από ενσωματωμένες συσκευές με περιορισμένες υπολογιστικές δυνατότητες. Έχοντας ως έναυσμα την φύση αυτών των συσκευών συζητάμε λιγότερο απαιτητικά πρωτόκολλα, καθώς το πρωτόκολλο SSL/TLS έχει αποδειχθεί επαχθές στην παραδοσιακή του μορφή και προτείνουμε παραλλαγές γνωστών πρωτοκόλλων που μπορούν να προσαρμοστούν στην αρχιτεκτονική τους.

Συνοψίζοντας, στην πτυχιακή αυτή εργασία αναλύεται η εξάπλωση των ψηφιακών πιστοποιητικών δημοσίου κλειδιού και πως αυτή συσχετίζεται με την επισχεψιμότητα των ιστοσελίδων και πως επηρεάζεται από κυβερνοεπιθέσεις και γνωστοποιήσεις γνωστών αδυναμιών του πρωτοκόλλου. Δίνεται επίσης έμφαση στους μηχανισμούς ελέγχου ανάκλησης των πιστοποιητικών και των αδυναμιών που έχουν σημειωθεί σε αυτούς και αναλύονται προτεινόμενες λύσεις σε αυτές. Εν κατακλείδι καθώς οι συσκευές του IoT είναι ήδη αρκετά διαδεδομένες χωρίς να έχουν τις απαραίτητες προϋποθέσεις για την διασφάλιση της ασφαλούς επικοινωνίας, αναφερόμαστε στα βασικά προβλήματά τους και τις λύσεις που μπορούν να προσφέρουν τα ψηφιακά πιστοποιητικά, με σκοπό να προετοιμαστεί το έδαφος για μελλοντικές εργασίες.



Ευχαριστίες

*στην οικογένειά μου*



# Contents

<b>Table of Contents</b>	<b>i</b>
<b>List of Tables</b>	<b>iii</b>
<b>List of Figures</b>	<b>v</b>
<b>1 Introduction</b>	<b>3</b>
1.1 From Secure Sockets Layer (SSL) to Transport Layer Security (TLS) protocol . . . . .	4
1.2 SSL certificates . . . . .	4
1.3 Types of SSL/TLS certificates . . . . .	5
1.3.1 Distinction based on the number of secured entities . . . . .	5
1.3.2 Distinction based on the validation process . . . . .	6
1.4 Related Work . . . . .	7
<b>2 Data analysis</b>	<b>9</b>
2.1 Certificate Transparency (CT) . . . . .	9
2.2 Data sets . . . . .	9
2.3 Valid certificates per subject . . . . .	11
2.4 Certificate Authorities (Certificate Authority (CA)) . . . . .	13
2.4.1 Focusing on Let's Encrypt . . . . .	13
2.4.2 Quality over quantity . . . . .	14
2.4.2.1 The curious case of Google . . . . .	15
2.5 Validity period . . . . .	16
2.5.1 Validity period trends based on our data . . . . .	16
2.6 Signature Algorithms . . . . .	17
<b>3 Trust</b>	<b>19</b>
3.1 Known weaknesses . . . . .	19
3.2 Place of trust in the certificates . . . . .	20
3.2.1 SSL Black List . . . . .	22
3.3 When you cannot even trust the trusted . . . . .	22
3.3.1 Beyond CAs . . . . .	23

<b>4</b>	<b>Certificate revocation</b>	<b>25</b>
4.1	Revocation statistics . . . . .	25
4.2	Revocation responsibility . . . . .	28
4.3	Association of <i>hacks</i> and revocations . . . . .	31
4.3.1	What is the ranking of hacked sites in Alexa? . . . . .	33
4.4	Revocation Protocols . . . . .	34
4.4.1	Certificate Revocation Lists (CRLs) . . . . .	35
4.4.2	Domain Name Server (DNS) based approaches . . . . .	35
4.4.3	OCSP and CCSP . . . . .	36
<b>5</b>	<b>Shift of paradigm and Conclusion</b>	<b>37</b>
5.1	Why is there a distinction ? . . . . .	37
5.2	Transport layer problems . . . . .	38
5.2.1	Security relevant errors . . . . .	39
5.2.2	Securing the Internet of things . . . . .	39
5.3	Solutions . . . . .	40
5.4	Conclusion . . . . .	40
	<b>Bibliography</b>	<b>43</b>

# List of Tables

2.1	Subjects with more than 1000 certificates in the data set . . . . .	11
2.2	Top ten most common validity periods in dataset . . . . .	16
2.3	Signature algorithms in CT data set . . . . .	18
3.1	Common Weaknesses related to certificates . . . . .	20
4.1	Time periods with most revocations and known attacks . . . . .	33
4.2	Summarizing table of the comparison results: DANE - DCSP . . . . .	36
4.3	Summarizing table of the comparison results: OCSP variations - CCSP . . . . .	36



# List of Figures

2.1	CT additional components to TLS certificates . . . . .	10
2.2	Frequency of subjects in data . . . . .	12
2.3	Top ten issuers . . . . .	14
2.4	Issuers in Alexa top 1000 and Alexa top 100000 . . . . .	15
2.5	Validity periods of certificates measured in days . . . . .	17
3.1	Chain of trust, provided by Yanpas via Wikimedia Commons . . . . .	21
4.1	Revocations in CT data set in log scale. . . . .	26
4.2	CDF of percentage of revocations after the day of issue of the certificate measured in days. . . . .	27
4.3	Revocations after the day of issuance and before the day of expiration of the certificate measured in days. . . . .	28
4.4	Top ten issuers of revoked certificates . . . . .	29
4.5	Number of revocations after issues within a period of 3 months divided to weeks . . . . .	30
4.6	CDF of days of revocation in Alexa top 1000 and in Alexa range 50000 to 51000 . . . . .	31
4.7	Certificates' revocation and significant events from 2011-01 to 2016-09 . . . . .	32
4.8	Monthly attacks chart from Hackmageddon . . . . .	32
4.9	Days between the issue and the revocation for the Alexa top 5000 sites certificates which showed correlation with the hacks from September 2015 to August 2016 . . . . .	34
5.1	Comparison of IoT and Web protocol stacks . . . . .	38





# Acronyms

- AIC** Availability, Integrity and Confidentiality. 3, 40
- CA** Certificate Authority. 4, 7–9, 12–15, 21–23, 25, 28, 35, 36
- CCSP** CCSP: a Compressed Certificate Status Protocol. 36
- CDN** Content Delivery Network. 34, 36
- CN** Common Name. 5, 6, 23
- CRL** Certificate Revocation List. 4, 34, 35
- CT** Certificate Transparency. 9, 10, 13, 14, 17, 22, 23, 25, 30
- CWE** Common Weakness Enumeration. 19
- DANE** DNS-based Authentication of Named Entities. 35, 40
- DCSP** DCSP: Performant Certificate Revocation a DNS-based approach. 35, 36, 40
- DDoS** Distributed Denial of Service. 10
- DNS** Domain Name Server. 10, 35
- DNSSEC** Domain Name System Security Extensions. 35
- DV** Domain Validation. 7
- EFF** Electronic Frontier Foundation. 11, 13, 22
- EV** Extended Validation. 7, 13, 22, 23
- HMAC** Key-Hashing for Message Authentication Code. 4
- HTTP** Hypertext Transfer Protocol. 3, 29
- HTTPS** HTTP over SSL. 3, 7, 8, 11, 14, 19, 29

- IEFT** Internet Engineering Task Force. 4
- IoT** Internet of Things. 3, 37–40
- OCSP** Online Certificate Status Protocol. 4, 34, 35
- OV** Organization Validation. 7, 13, 23
- OWASP** Open Web Application Security Project. 7, 38
- PKI** Public Key Infrastructure. 3, 4, 7, 8, 17, 19, 21, 37, 39
- PKIX** Public-Key Infrastructure (X.509). 3, 6, 10, 13
- RT** Revocation Transparency. 35
- SAN** Subject Alternative Name. 5–7, 27
- SCT** Signed Certificate Time-stamp. 9
- SHA** Secure Hash Algorithm. 10, 17
- SSL** Secure Sockets Layer. 3–11, 19–22, 25, 29, 31, 35, 37
- TLS** Transport Layer Security. 3–6, 8, 9, 13, 22, 29, 39
- WWW** World Wide Web. 3

# Chapter 1

## Introduction

The Internet started in the 1950s, to serve academic and military sectors and today, almost after 70 years, 3.8 billion [27] people are connected, with multiple devices. As a result, the information available at the moment, is enormous and will continue to increase exponentially. As the needs grow, it is vital that all times the Availability, Integrity and Confidentiality (AIC) triad, along with security, privacy and authentication are ensured. Hypertext Transfer Protocol (HTTP), the basic protocol of communication for the World Wide Web (WWW), is not secure, and there may be a time when all Internet traffic is transmitted through the SSL/TLS protocol HTTP over SSL (HTTPS), on the session layer - based on the OSI model, which offers encrypted communication and protects the privacy and integrity of the traffic and assures the identity of the entity it secures, with digital certificates.

The aspiration of this work is to emphasize on the importance of HTTPS adoption and to convince the readers that security is an essential aspect of all the Internet connections. To this end, we analyze the SSL/TLS certificates <sup>1</sup> in the wild, explain their basic components and offer an overview of the related trends as well as a helpful and interesting insight into their less known aspects. Furthermore, the trust model built around the issuing authorities and the certificates' infrastructure is challenged by highlighting several problematic cases. Especially, we aim attention to revocation, which has often been neglected and discuss about well established and promising, newly proposed protocols. Last, we examine whether they can be used as a security solution for the emerging Internet of Things (IoT), which is compiled by embedded devices lacking computational power, so as to predict how they can be deployed in this shift of paradigm.

---

<sup>1</sup>In this study we will use the term SSL certificates, despite that currently the TLS protocol, is in use. Moreover, the term (digital) Public Key Infrastructure (PKI) or Public-Key Infrastructure (X.509) (PKIX) certificates is also adopted.

## 1.1 From SSL to TLS protocol

SSL was adopted by Netscape in 1994, as a mean to establish a secure communication channel. Through the years, it evolved[71], with patches and fixes to version 3.0, but today all SSL versions are proven to be broken and unsecured [52], [44]. TLS is the successor of SSL, now in version 1.2 [41], and the draft version of 1.3 [37]. It started as an attempt to standardize SSL by Internet Engineering Task Force (IETF). Although TLS is not failproof either, it uses the more secure Key-Hashing for Message Authentication Code (HMAC) to ensure that a record cannot be altered during transmission and it is better supported. A walk through to the milestones (supporting initiatives, vulnerabilities, attacks targeted the protocols and certificate authorities) that shaped the SSL/TLS and PKI ecosystem is provided by Ivan Ristic [70] and is also discussed in this work.

## 1.2 SSL certificates

An abstract description of an SSL certificate is that it is a piece of code, that uses a cryptographic key to authenticate and verify the identity of the secured entity and to ensure data encryption when communicating with that entity. In other words, PKI binds SSL certificates to a subject, so as to ensure trust to the identity of the secured entity. Users know that a certificate is in place by the indicative green padlock symbol or trust mark. Encryption safeguards that all sensitive information exchanged via the website cannot be intercepted and read by anyone other than the intended recipient. Thus, digital certificates enhance the customer trust for economical transactions and as an aftermath increase the website's Google Rankings.

Before we move to the study of SSL certificates' ecosystem, it is important that we present the latest structure of a digital certificate. The common fields <sup>2</sup> of an X.509 digital certificate [13] are the following:

- Serial Number: Used to uniquely identify the certificate within a CA's systems. In particular this is used to track revocation information.
- Subject: The entity a certificate belongs to: a machine, an individual, or an organization.
- Issuer: The entity that verified the information and signed the certificate.
- Not Before: The earliest time and date after which the certificate is valid.
- Not After: The time and date past which the certificate is no longer valid.

---

<sup>2</sup>X.509 also includes standards for Certificate Revocation List (CRL) implementations and Online Certificate Status Protocol (OCSP)

- **Key Usage:** The valid cryptographic uses of the certificate's public key. Common values include digital signature validation, key encipherment, and certificate signing.
- **Extended Key Usage:** The applications in which the certificate may be used. Common values include TLS server authentication, email protection, and code signing.
- **Public Key:** A public key belonging to the certificate subject.
- **Signature Algorithm:** The algorithmic rule used to sign [2] the public key certificate.
- **Signature:** A signature of the certificate body by the issuer's private key.

### 1.3 Types of SSL/TLS certificates

SSL server certificates are used to secure hosts like a website, a mail server, a directory server, or any other type of server that needs to be authenticated, or that wants to send and receive encrypted data. They must be tied to one or more domains, or subdomains, server names, hostnames, which are specified in the subject field, by the Common Name (CN) or in some cases the Subject Alternative Name (SAN). To discuss the different types of SSL certificates this section focuses on two main aspects; the number of secured entities and the validation process.

#### 1.3.1 Distinction based on the number of secured entities

Based on the number of domain names or subdomains linked to the certificate, there is a clear distinction to:

**Single domain** The single domain certificate only covers one specific domain or subdomain, so if a website owner/admin wants to secure mydomain.com and mail.mydomain.com, she will have to issue two different certificates, one for each. Note that, if a CN is for mydomain.com, it does not also secure www.mydomain.com, but if the CN is for www.mydomain.com, it also covers the case of mydomain.com.

**Wildcard (\*)** A wildcard certificate (\*.mydomain.com) is a public key certificate which secures all first-level subdomains of a single domain name, like abc.mydomain.com, users.mydomain.com, main.mydomain.com, www.mydomain.com, etcetera. It does not, however, match a.b.mydomain.com or mydomain.com. Due to the effortless management of only one certificate for an unlimited number of prefix names instead of plenty, they are considered a very convenient solution.

For a wildcard certificate to work, all the subdomains need to be sharing the same private key. This creates some security implications since in the case the

private key is stolen, then the certificate will have to be revoked, and this will impact all the servers at the same time. When using distinct certificates per server, the damage is more contained.

**Multi domain** Another interesting case of certificates to examine, are ‘Multi domain’ or SAN certificates. They can contain different base domains, which may have no connection among them, but cannot include wildcard certificates<sup>3</sup>. For this reason, subdomain names must be added as a unique domain name entry in the certificate. For example a SAN may include:

- mydomain.com
- www.thatdomain.eu
- mydomain.net
- myotherdomain.com
- any-domain.any-tld

The Common Name can only contain up to one entry: either a wildcard or a single name. On the contrary, the SAN field lets you specify additional host names (sites, IP addresses, CNs, etcetera.) to be protected by one SSL certificate, such as a multi domain SAN. However, when decoding a multi domain certificate one cannot distinguish all the host names by looking at the CN of the certificate, instead, he sees the group name that contains them, for instance, sni183291.cloudaressl.com.

It is important to note that in order to add or remove websites, as well as, each time a change is made, the certificate must be reissued and replaced on all the websites it protects. As opposed to wildcard certificates, there is a limit to how many different domain names a SAN can support. This number amounts to 100 sites in most cases. This is quite reasonable as the more sites a certificate covers, the more time it takes for it to be downloaded.

### 1.3.2 Distinction based on the validation process

When requesting for a PKIX certificate there are options regarding the validation process for the secured entity. This validation process refers to how thorough the research CAs perform during the domain name validation process is and consequently the level of user trust for SSL/TLS negotiations. Based on that, certificates are divided to:

---

<sup>3</sup>There are some hybrid cases of multi domain certificates including a wildstar certificate, but are quite limited.

**Domain Validation (DV)** Domain Validated certificates offer the simplest type of check for the identification of the secured entity, against the domain registry. Symantec [20] proposes not to use them for commercial purposes since they do not provide identifying organizational information, meaning that the certificate gives no reassurance whether the business/website it secures is legitimate. Hence, they are considered as high-risk certificates to use on a public website. Section 3.3 mentions several cases where they are used to conceal malicious behavior.

**Organization Validation (OV)** Organization validated certificates are considered trusted and therefore can be used to a commercial or public facing website. Contrary to DV, during OV organizations are strictly authenticated by real agents against business registry databases hosted by governments to ensure their authenticity and legitimacy.

**Extended Validation (EV)** The criteria for issuing EV certificates are defined by the Guidelines for Extended Validation [23], in which the secured entity goes through a much stricter process than that for OV certificates. Due to their nature, wildcard certificates do not support Extended Validation Certificate Guidelines. Open Web Application Security Project (OWASP) has a specific rule about not using them [10] which states that ‘they violate the principle of least privilege and ask the user to trust all machines’. In section 3.3, there are additional examples regarding the security of such certificates [55]. On the contrary, in case of SAN, Extend Validation Multi-Domain Certificates can be issued.

## 1.4 Related Work

In the past, there have been notable studies and measurement papers of the SSL ecosystem. The extensive measurement paper by [51] written in 2011, performs HTTPS scan to analyze the deployment of the X.509 PKI. Similar to our approach, it stresses that there are issues with the deployment of the certificates that result in the insufficiency to meet the requirements of a secure PKI. Indeed, 2011, as it will be later discussed, was a very bad year for CAs and many fraudulent certificates were issued at that time. The aftertaste of this analysis is that “the X.509 certification infrastructure is, in great part, in a sorry state”.

Similarly, ‘Measuring HTTPS adoption on the web’ [47], which was a cooperative work by Google, Cisco, and Mozilla, measures the progress of the HTTPS adoption. This is a very recent publication, made in 2017, so it provides up-to-date statistics. This work confirms a hypothesis we also make, that server support for HTTPS follows a long tail distribution, i.e. there are relatively few high traffic sites on the top and many websites residing in the long-tail. Contrary to the aforementioned studies, white papers from Certificate Authorities, mostly emphasize on their services but also help us get a better understanding of financial aspects of digital certificates.

Consequently, the issue of *trust* has caused a lot of discussions. TLS protocol and the infrastructure of certificates have been put under the microscope as there have been serious vulnerabilities and errors linked to them. Some of the most detailed and impactful works are listed here. ‘Here’s My Cert, So Trust Me, Maybe?’ [28], published in 2013, focuses on TLS errors and warnings and how misconfigurations can cause users to misinterpret a real attack. In the same year, ‘SoK: SSL and HTTPS’ [40], evaluates certificate trust model enhancements. Likewise to our study, it ponders upon the attacks to the SSL protocol to challenge the trust to the certificate and the certification authorities. Moreover, “Certificate Authority” Trust Model for SSL [72], which was published in 2010, targets the CAs and the authentication process they use for the entities to be secured.

Furthermore, ‘An End-to-End Measurement of Certificate Revocation in the Web’s PKI’ [46] analyzes the frequency with which certificates are revoked and whether the essential checks are in place. This work, published in 2015, was the motivation for proposing new revocation protocols, which are later discussed, which attempt to complement existing approaches.

This thesis offers measurements of different aspects of SSL certificates, to give a better insight into their key elements and also challenges the trust which is promoted by them. We pay tribute to the often neglected case of revocation, to highlight how important it is and attempt to compare existing protocols. Concluding after having shown the strengths and weaknesses of the SSL certificates’ infrastructure, we briefly discuss whether they are suitable to secure the ‘Internet of things’. Following chapters will cite, contemplate and attempt to extend the aforementioned studies and reports, to offer a holistic view of digital certificates and the trends related to them. We ponder on the past, to understand the present and predict the future of the SSL ecosystem.



## Chapter 2

# Data analysis

In the past, researchers posed questions and struggled to find the data to answer them. Now, in the era of big data, we have a plethora of data and we attempt to understand them and to extract the proper queries from them. The basis for this work is the Certificate Transparency project, which has been correlated with other public data sets to draw statistics regarding the structural elements of a certificate and their ecosystem, in an attempt to explain how the Secure Sockets Layer (SSL) certificates are developing.

### 2.1 Certificate Transparency (CT)

CT[1] is an open project, initiated in 2013 by Google, under which SSL certificates are logged as they are issued or observed, so as to promote transparency. At the moment, it is the responsibility of a certificate owner or a Certificate Authority (CA) to submit a certificate for logging. Upon submission, the log responds with a Signed Certificate Time-stamp (SCT) based on its type. If a certificate does not have a valid SCT Transport Layer Security (TLS) servers must not accept it. The related Figure 2.1 shows how CT additional components are integrated to the digital certificates.

These logs are: append-only, cryptographically hashed and publicly available. Moreover, they use a binary Merkle Hash Tree for efficient auditing. SCT offers a solution to the issue of rogue or compromised CAs as certificates which are falsely issued can be rejected, since they will not appear in the logs. Thus, by adding an extra layer of logging it helps to maintain better monitoring of the overall process and to reduce the attack surface.

### 2.2 Data sets

Some of the data sets mentioned here are quite common amongst measurement studies and academic papers. The basic data set for measurements is provided by Certificate Transparency [34], counting approximately 25M entries and dates

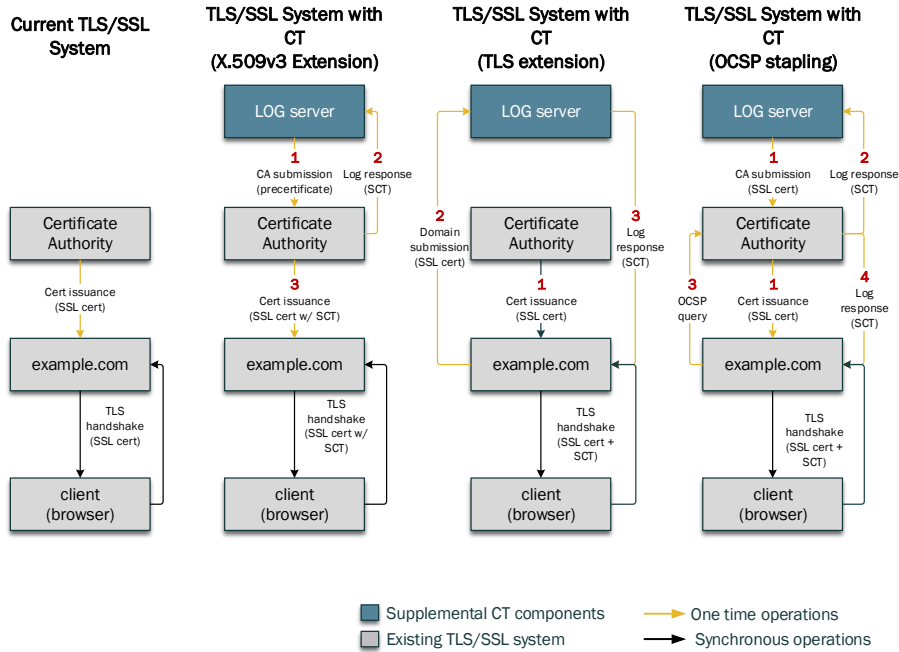


Figure 2.1: CT additional components to TLS certificates

to September 2016. The earliest record found in CT data set, dates back to May 1998, although CT started keeping logs in June 2013, so all the records from May 1998 to May 2013, which are less than 10% of all data, are observations and may be inconsequential. CT logs are used for a large-scale analysis of the SSL Certificates. Next sections discuss the different types of SSL certificates based on the entity being secured, the distribution of certificates to subjects and figure statistics for Certificate Authorities and the validity periods of certificates.

Additionally, to attest how high traffic sites perform with regards to certificates and whether they are frequently attacked, we used Alexa top 1 million, from the Alexa Top Sites service, which provides traffic rankings for websites. Consequently, to demonstrate the attack map we used security analytics from Hackmageddon [3], which is an open platform, based on user-submitted attacks. The attacks may refer to account Hijacking, known vulnerabilities, Brute-force, Distributed Denial of Service (DDoS), Defacement, Domain Name Server (DNS) Hijacking, Phishing, Malvertising, Malware, Targeted Attacks, etcetera and are divided based on the target.

Last, SSL Blacklist [14] data set, which has Secure Hash Algorithm (SHA)-1 fingerprints of certificates associated with malware or botnet activities, assisted to point out vulnerabilities and misuse of the Public-Key Infrastructure (X.509) (PKIX). The oldest record dates from May 2014 and the data we used for our

measurements contain 1761 unique entries.

## 2.3 Valid certificates per subject

Not long ago, only commercial sites were adopting HTTP over SSL (HTTPS) to encrypt their transactions, and in some cases, it was only present at checkout. Major institutions and companies like Electronic Frontier Foundation (EFF), Google, Mozilla, Cisco have tried to push the limit for a universal adoption of HTTPS. The recent study [47] that measures the use of HTTPS, states that Alexa Million and generally IPv4 still have little HTTPS support, while even within the top websites HTTPS full adoption recently surpassed 50%.

An analysis of our data substantiates that SSL certificates still have a long way ahead. The majority of subjects<sup>1</sup> relate to only one certificate and very few subjects have more than 1000. Specifically, Table 2.1 lists only eight records, sorted in decreasing order, which have more than 1000 certificates. The first record which is a subject titled `tls.automattic.com`, does not refer to a specific domain. Based on [75], all custom domains on WordPress.com, are secured with a certificate from the Let’s Encrypt Certificate Authority, under the same Common Name, `tls.automattic.com`.

#certificates	Subject name
139732	tls.automattic.com
29736	incapsula.com
11097	meeting.itmatik.ch
6953	firebaseapp.com
2019	*.firebaseapp.com
1816	bentoboxlinux.org
1373	markng.co.uk
1363	luckybeads.de

Table 2.1: Subjects with more than 1000 certificates in the data set

Our understanding of the SSL ecosystem and the measurements carried out throughout this study, show that only a few sites are actually security aware and thus better maintained, so it is expected that when we plot the distribution of the number of certificates each subject has, the curve will be decreasing. One step further than that, we are also expecting to observe power law as it can be used to describe phenomena where a small number of items is clustered at the top of a distribution (or at the bottom), taking up the vast majority of the resources. In other words, it implies a small amount of occurrences is common, while larger

<sup>1</sup>When the term subject is used in this study, it refers to the entity being secured. Particularly a subject may be associated with a base domain, subdomain or a group name.

occurrences are rare. Indeed, Figure 2.2 shows a strong indication of “piece wise power law” or “broken power law”. The data can be fitted by curves of the following form:  $y = k * (1/(x^\alpha))$  and when representing them on a log-log scale there are several straight line segments. The x-axis represents the number of certificates of each subject and the y-axis the frequency of this value in the data.

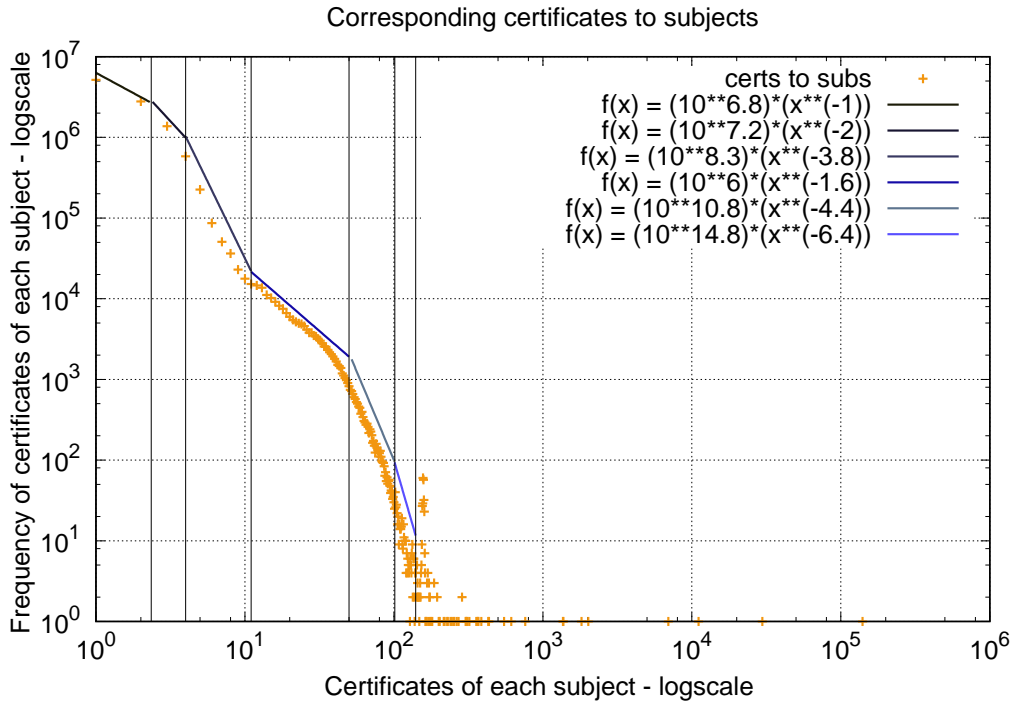


Figure 2.2: Frequency of subjects in data

Based on this plot, almost 50% of subjects appear once in the data, or in other words, almost half of all the distinct subjects only have one certificate. Additionally, 80% of certificates relate to subjects which appear once to ten times in all dataset. Taking into consideration how the issuing authorities have evolved, which will be further discussed in Section 2.4, and that after October 2016 HTTPS traffic reached 50%, in 2017 we can finally say the HTTPS has reached the point where it is becoming the norm rather than the exception [62, 54], this is quite reasonable.

Next sections which discuss the prominent CAs and the validity period of a certificate, answer why certain subjects have from 1 to 150 certificates. Also although the curve is generally decreasing, there is a small range of x-axis values close to 155, where we observe an increase in the frequency on the y-axis. This is due to many different certificates issued for distinct Google regional domains subjects' in batches. After that range, the curve is once again decreasing.

Regarding the change of the decreasing rate, our hypothesis is that most

sites/subjects have started using certificates, after 2014, with the initiative of Google to rank higher the sites using TLS [35], and with the increasing number of attacks and public announcements of vulnerabilities that drove administrators to take security into higher consideration. Additionally, as next sections will show, the establishment of Let's Encrypt CA is also responsible for a great part of certificates, which would only be linked to subjects that own few certificates since it is relatively new.

## 2.4 Certificate Authorities (CA)

Essentially anyone can issue and sign PKIX certificates, however such self-signed certificates are not considered trusted, since they are signed by the same entity, whose identity they certify. To acquire a certificate which is universally accepted, website administrators turn to CAs, which act as a trusted third party, that vouches for the identity of the secured subject. The choice of a CA depends on the needs of a domain and of course what each issuer can provide. The key elements are the CA's reputation, whether it can promote trustworthiness, the services it offers and their price.

At the moment the number of certification authorities that a browser trusts is enormous [8]. In this section, however, we focus on the most preferred authorities in the CT dataset, and their correlation to sites with high traffic. Figure 2.3 shows the 10 most preferred certification authorities as organizations and with their Common Names. We use this distinction of organizations and Common Names, to separate organizations' subsidiaries. For instance, COMODO CA Limited has Common Names related to COMODO, PositiveSSL and EssentialSSL, GeoTrust Inc. has Geotrust and RapidSSL and GlobalSign has GlobalSign, Cybertrust SureServer and AlphaSSL.

Top ten CAs organizations cover 93% of all certificates. The remaining certificates are mainly issued by Starfield Technologies, cPanel, Entrust, Terena, WoSign, GANDI, etcetera.

### 2.4.1 Focusing on Let's Encrypt

Numbers indicate that Let's Encrypt is the most common certificate issuer owning about 36% of all certificates in our data set. It was an initiative by EFF, Mozilla Foundation, The University of Michigan, Akamai Technologies and Cisco Systems and started as a public beta issuer in December 2015 and by September 30, 2016 it has issued over 12 million certificates[48]. It is a free, automated, and open Certification Authority, which is one of the main reasons that make it so popular.

Let's Encrypt certificates are standard Domain Validation certificates and according to their policy [7], they do not offer Organization Validation (OV), Extended Validation (EV), or wildcard certificates however, they support certificates for multiple domain names (SAN certificates). Moreover, their certificates are in

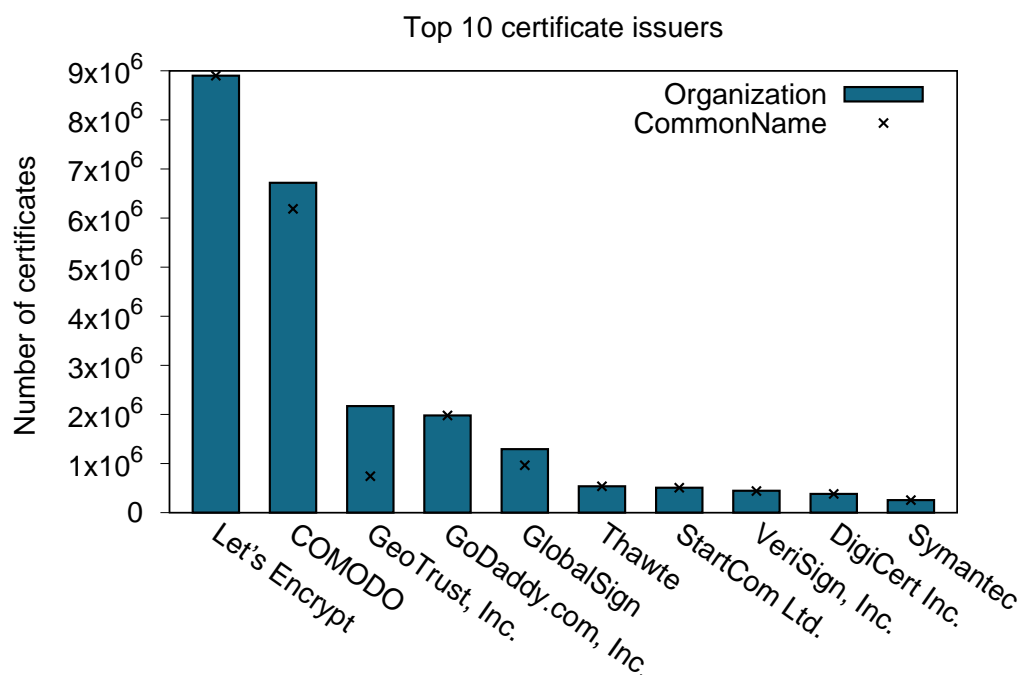


Figure 2.3: Top ten issuers

most cases valid for 90 days, which is the most frequent value noted in our data set.

### 2.4.2 Quality over quantity

This section deals with the question whether the CAs that are most noted in the dataset, are also preferred by high traffic sites. *Why are the 10 top CAs that popular? Is it because they issue certificates to many less known sites, or are they also popular to high traffic sites?* To resolve that, we used Alexa top 1.000 and Alexa top 100.000 <sup>2</sup> sites. To test this correlation, we used domains, instead of subjects.

The results which again favor the top 10 issuers, as they appear from the CT measurements, are depicted in Figure 2.4. Both axes show the number of certificates (in logscale) and CAs are observed in a linear fashion. This means that we have a power law function of the form  $y = k * (1/(x^\alpha))$ .

<sup>2</sup>running nmap for port 443 to Alexa 100.000 top domains showed that approximately 70% are using HTTPS and 12% are filtered

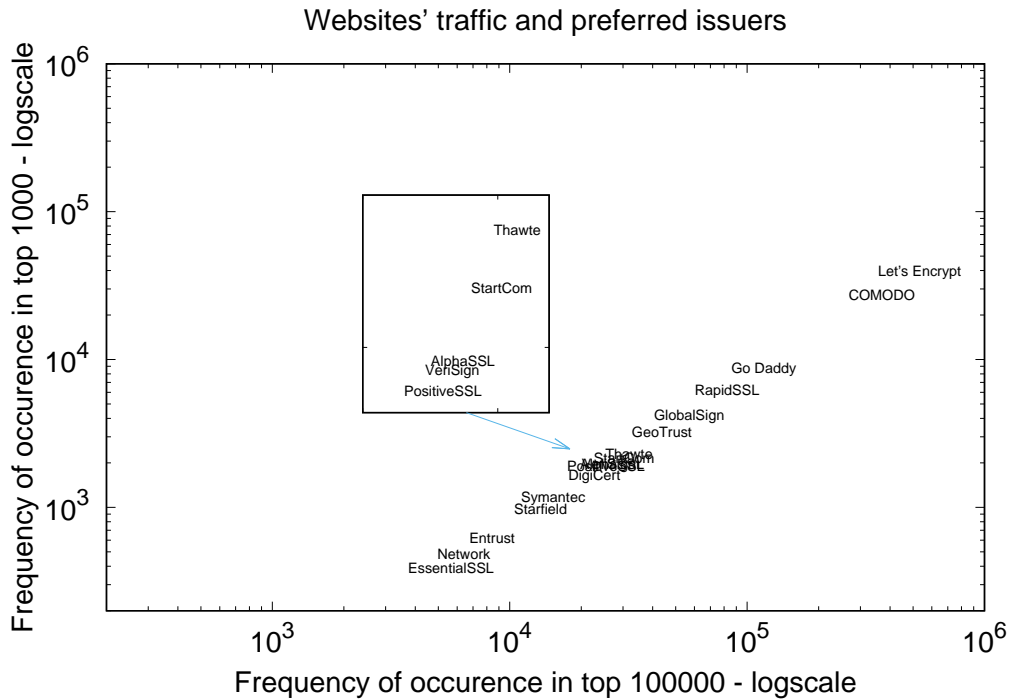


Figure 2.4: Issuers in Alexa top 1000 and Alexa top 100000

#### 2.4.2.1 The curious case of Google

Google domains are both very popular and also own a significant percentage of certificates. As we tried to determine the correlation of websites to the CAs, the results showed that Google owned domains<sup>3</sup> certificates' don't have a single issuer, but on the contrary, one can observe many different issuers at a regional level and various services. The majority of the certificates though are issued by Let's Encrypt, where Google Chrome is a platinum sponsor.

1. Google regional domains ( www.google... ) .  
This case only covers www.google.com, www.google.co.uk and www.google.de which are in Alexa's top five regional Google domains. It is also worth mentioning that even in the same regional domain, issuers change over time.
2. Wildcard certificates (i.e. \*.google.com , \*.google.com.af, \*.google.ws, ...) .  
Google also uses wildcards for \*.mail.google.com, \*.c.docs.google.com, \*.chrome.google.com, \*.clients.google.com, \*.ext.google.com, \*.storage.googleapis.com, \*.vp.video.l.google.com.

<sup>3</sup>this is based on wikipedia's list of Google domains

## 2.5 Validity period

The validity period of a certificate starts the earliest time and date on which the certificate is valid (not before) and ends the time and date past which the certificate is no longer valid (not after). This, of course, does not include the case of a revocation. Each issuer has a different policy, and each website owner/administrator has variant needs. This section will inspect the general trend around validity periods, the extremes and the common case.

### 2.5.1 Validity period trends based on our data

Most certificates are valid for a period of approximately 3 months, 6 months, 1 year, 2 years and 3 years based on Table 2.2 which contains the top ten most common values as a certificate's validity period.

Days	Clustered period	Frequency in data
90	Approx 3 months	7 680 247
365	Approx 1 year	2 657 517
366	Approx 1 year	1 472 129
89	Approx 3 months	1 177 160
1095	Approx 3 years	476 487
368	Approx 1 year	476 307
730	Approx 2 years	425 412
185	Approx 6 months	398 070
187	Approx 6 months	378 971
186	Approx 6 months	376 483

Table 2.2: Top ten most common validity periods in dataset

Related statistics:

- The longest period of time is approximately 50 years (18.263 days), *which is probably a misconfigured entry.*
- The average period of time is approximately 11 months (330 days).
- The minimum period was less than a day
- The validity period most frequently noticed in data is 90 days, with a frequency of occurrence 7.680.247 which is close to 31% of the data set.

The two corresponding graphs of Figure 2.5 show the CDF distribution of values based on the validity periods. Figure 2.5a shows the overall validity periods in the data set, while Figure 2.5b serves as zooming into the most observed validity periods as stated in Table 2.2. It is evident that almost 90% of all certificates are valid for less than 3 years.



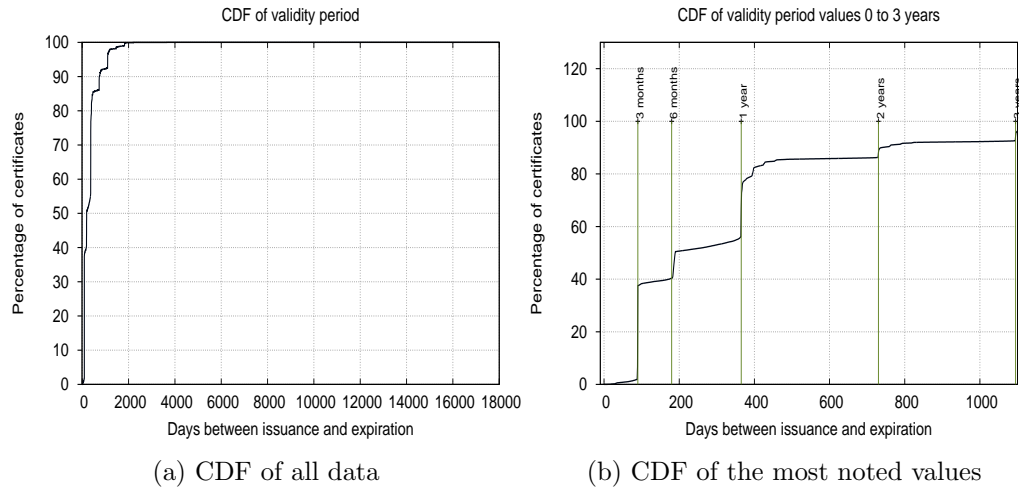


Figure 2.5: Validity periods of certificates measured in days

Microsoft best practices [66] suggest based on the Public Key Infrastructure (PKI) key size, that for key length of 1024 the validity period should not be greater than 6 months to 1 year and for key length of 2048 the validity period should not exceed 2 years.

## 2.6 Signature Algorithms

Before moving to the chapter of *Trust*, we list the signature algorithms observed in the CT data set. The choice of the signature algorithm is important since many known attacks are related to the use of weak ciphers. In all the data set there were only 11 signature algorithms observed in total, which are listed in Table 2.3. Although the last five are used by very few certificates we list them for completion. Also, support for MD5 based signatures was removed in early 2012, this is why we only observe 0,03%. Interestingly although SHA1 is considered outdated, since as computing power has increased, so does the feasibility of breaking it, sha1WithRSAEncryption is on the second place with 20,81%.

Additionally, based on the joint effort of Mozilla Firefox and Google Chrome engineers, **badssl** lists outdated and weak Hashing Algorithms and cipher suites as: CBC, RC4-MD5, MD5, 3DES, NULL.

<b>Signature algorithm</b>	<b>Percentage</b>
sha256WithRSAEncryption	61,93%
sha1WithRSAEncryption	20,81%
ecdsa-with-SHA256	17,19%
sha512WithRSAEncryption	0,04%
md5WithRSAEncryption	0,03%
sha384WithRSAEncryption	0,00%
ecdsa-with-SHA384	0,00%
dsa-with-SHA256	0,00%
sha1WithRSA	0,00%
md2WithRSAEncryption	0,00%
ripemd160WithRSA	0,00%

Table 2.3: Signature algorithms in CT data set

## Chapter 3

# Trust

Secure Sockets Layer (SSL) prevents the unauthorized access to data and Public Key Infrastructure (PKI) certificates promote **trust** to the secured entity. The majority of users are much more likely to share private data to a website with a certificate without fully understanding how exactly this certificate works, who issued it and under which validation process. If someone examines them thoroughly, he will discover several very common problems with their application. Such problems are known weaknesses, errors and cases of misuse related to the SSL certificates and also challenges the mechanisms employed by trusted third parties, responsible for their issuance and status.

Additionally, there has been a serious debate on whether Certification Authorities should be regarded as trusted. Bruce Schneier [43] has stated “Who do we trust, and for what? There’s a risk from an imprecise use of the word trust”, to emphasize on the trust model gaps of CAs. Steven B. Roosa and Stephen Schultze [72] have also discussed over their deficiencies and Jeremy Clark [40], has proposed enhancements to the certificate infrastructure. This chapter discusses the issue of trust, citing known weaknesses of the standards for the issuance, management, administration, distribution and maintenance of digital certificates, so that users become much more cautious and alert.

### 3.1 Known weaknesses

The HTTP over SSL (HTTPS) protocol is meant for securing end to end traffic. “Security is a chain; it’s only as strong as the weakest link”. To claim that an entity is trusted and that the communication with it, is secure, you need to be able to prove that all security mechanisms and their implementation, are flawless.

Table 3.1, has a list of Common Weakness Enumeration (CWE) [22] related to the SSL certificates design and implementation to highlight their consequences. CWE furthermore, informs about the application platform, potential mitigations and offers examples and a likelihood indicator, which also emphasize the severity of the weaknesses.

<b>ID:Name</b>	<b>Description</b>	<b>Consequences</b>
CWE-295: Improper Certificate Validation	The software does not validate, or incorrectly validates, a certificate.	Bypass protection mechanism; Gain privileges / assume identity
CWE-296: Improper Following of a Certificate's Chain of Trust	The software does not follow, or incorrectly follows, the chain of trust for a certificate back to a trusted root certificate, resulting in incorrect trust of any resource that is associated with that certificate.	Gain privileges / assume identity; Execute unauthorized code or commands
CWE-297: Improper Validation of Certificate with Host Mismatch	The software communicates with a host that provides a certificate, but the software does not properly ensure that the certificate is actually associated with that host.	Gain privileges / assume identity; Trust based on the expired certificate may allow for spoofing or redirection attacks.
CWE-298: Improper Validation of Certificate Expiration	A certificate expiration is not validated or is incorrectly validated, so trust may be assigned to certificates that have been abandoned due to age.	The data read from the system vouched for by the expired certificate may be flawed due to malicious spoofing.
CWE-299: Improper Check for Certificate Revocation	The software does not check or incorrectly checks the revocation status of a certificate, which may cause it to use a certificate that has been compromised.	Gain privileges / assume identity; Integrity and Confidentiality of data are compromised
CWE-599: Missing Validation of OpenSSL Certificate	The software uses OpenSSL and trusts or uses a certificate without using the <code>SSL_get_verify_result()</code> function to ensure that the certificate satisfies all necessary security requirements.	Gain privileges / assume identity; Bypass protection mechanism; Read application data

Table 3.1: Common Weaknesses related to certificates

## 3.2 Place of trust in the certificates

Digital certificates can be created by anyone, but to be validated and widely accepted they need to be part of a chain. Figure 3.1 shows the SSL certificate chain, which consists of end-user certificates, succeeded by one or more levels of intermediate certificates, and a root certificate at the top level. Root certificates are

self-signed Certificate Authority (CA) certificates which are the basis of the PKI's trust, and in most cases, they are embedded in all major browsers. Intermediate certificates are the middle links of the chain, and they can vouch for intermediate CAs. Last, the end-user certificates, cannot sign other certificates and are issued for email protection, server/client authorization, code signing, etcetera.

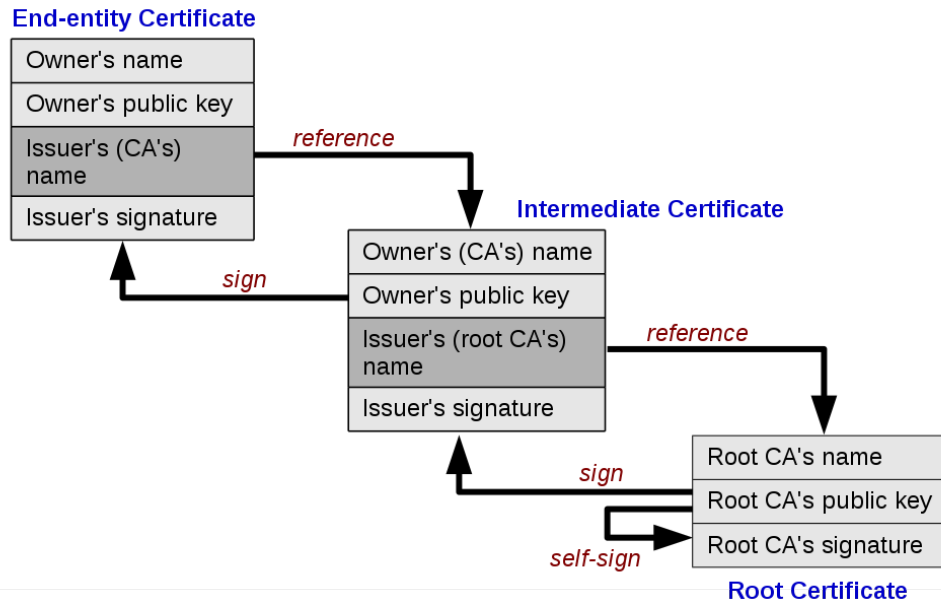


Figure 3.1: Chain of trust, provided by Yanpas via Wikimedia Commons

CC BY-SA 4.0 (<http://creativecommons.org/licenses/by-sa/4.0>)

There are cases of misconfigured certificates which are either not accepted by most browsers or leave the choice to the end user after prompting him with an error message. <https://badssl.com/> lists several cases to which the connection is not secured with respect to the certificate of the related domain.

Here are some of the most usual:

- expired, the certificate is served after its validity period.
- wrong.host, the SSL certificate used is meant for any other domain on the server.
- self-signed, the certificate is signed by the same entity whose identity it certifies.
- untrusted-root, the root of the certificate is not considered trusted.
- pinning-test, when trying to associate a host with their expected X509 certificate and it fails.

- revoked, the certificate was revoked, ergo it should not be trusted.

As most browsers do not accept such certificates, we do not consider them as hazardous. Akhawe et al. [28] however, raise the issue of Transport Layer Security (TLS) errors on the web and how they affect users. Due to many misconfigurations, when browsers report TLS errors, there are many false positive, so users may observe a warning but do not give it the appropriate attention. In another work by Akhawe in 2013 [29], there are measurements of 'Click-through rates' (bypassing malware and phishing warnings), which were up to 33.0% of Mozilla Firefox and shockingly, up to 70.2% for Google Chrome's SSL warnings.

### 3.2.1 SSL Black List

There are many cases of intentionally 'tweaked' certificates [53] which may look perfectly legitimate. SSL Blacklist (SSLBL) is a project maintained by <https://ssllbl.abuse.ch/>, which provides a list of known SSL certificates associated with malware and botnet activities. In such activities, attackers steal digital certificates which users trust, to sign malware with a seemingly perfectly legitimate certificate. Such cases have also been studied by [32].

To record how many of these certificates are still in use or archived, we used the SSL abuse data set. From May 2014 to September 2016, the results indicated that 41 records (2.3%), were also found in the Certificate Transparency (CT) data set, although CT claims to only have valid certificates. CT's, role to protect users from bad certificates does not mean that it can prevent them from being issued, instead, because of the logging system it is based upon, it offers the ability to check whether a certificate has indeed been issued by a CA.

## 3.3 When you cannot even trust the trusted

There are numerous fake, fraudulent, or phishing websites, whose names resemble perfectly legitimate ones. Inexperienced users may be victims of scams and phishing attacks, however, you would expect that a certification authority would not vouch for it. Certification Authorities do not check the validity of the entity they secure by default. Verisign issued two certificates for someone claiming to be a representative of Microsoft in 2001 [15]. These certificates were used to push allegedly updates to Microsoft software. Later in 2013, DigiCert issued a certificate for a company that does not exist which was used to sign a malware sample [74].

Additionally, two years prior that incident, in 2011, Electronic Frontier Foundation (EFF) [68] showed that domains with unqualified<sup>1</sup> names are a very common issue and the issuer that the major issuer of such certificates was GoDaddy. What is even more alarming is that some Extended Validation (EV) certificates that are issued to unqualified names. The measurements of this study indicated that the

---

<sup>1</sup>The term unqualified (or not fully qualified) refers to names that do not comply with the tree hierarchy of the DNS and as a result may not be globally unique.







## Chapter 4

# Certificate revocation

Certificates are issued with a pre-determined life span; an explicit issuance and expiration date. If however, for any reason, the trust to a certificate should be nullified, then the responsible Certificate Authority (CA) must revoke it. The most common cases of invalidation are an improperly issued certificate, its private key compromise, and others can be found at [45].

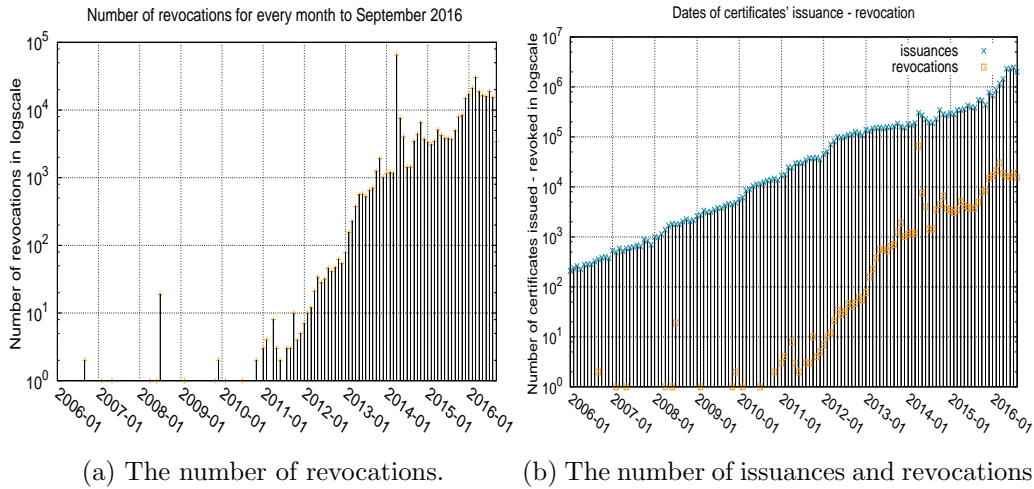
Our hypothesis is that although revocation could have solved many cases of mistrust to certificates [31], it is not widely established as the process of status checking is considered to have high overhead and thus is often neglected. In February 2012, Chrome decided to get rid of the checks altogether. The increasing number of attacks and the reaction of the academic community, however, highlighted its importance and new more scalable methods arose. Following sections will discuss interesting statistics relevant to revocation and consequently the related protocols' efficiency.

### 4.1 Revocation statistics

The analysis of Certificate Transparency (CT) data set indicated that revocations do not occur often. Specifically, there are 1,3% revocations in our approximately 25M records, data set. Similarly, from the aspect of subjects, about 2,2%, had at least one of their certificates revoked and only 14% of that 2,2% had more than one revoked certificate.

Figure 4.1a shows the number of revocations from the earliest date a revocation was noted in CT data set, to September 2016 and Figure4.1a compares them to the number of issuances. We will try to explain why some months have such high revocation rates and whether there is a pattern between attacks and vulnerabilities related to the Secure Sockets Layer (SSL) protocol and the revocation rates in following sections.

**What are the most common intervals between issuance and revocation?**  
In terms of days, figure 4.2 shows the percentages of the period from issuance to



(a) The number of revocations. (b) The number of issuances and revocations.

Figure 4.1: Revocations in CT data set in log scale.

revocation in days, in our data set. Evidently, 50% of the revocations happened within less than 80 days from issuance, and the vast majority, 80% of the revocations, took place less in less than a year. We note that the statistics for revocation follow those of the validity period of a certificate as measured in section 2.5.1.

Additionally, figure 4.3 shows the days after issuance and before expiration with regards to their revocation date, for all revoked certificates. The most common combination of days after the day of issuance and days before expiration for revocations is 0 and 90, which applies to 2,5% of revocations. Furthermore, although a logical assumption would be that all revocations happened within the validity period of the certificate, results reveal that 0,4% actually took place, after the end of the validity period. In 4.3 this is noted by the negative values in x-axis.

**Which are the most revoked certificates?** Our measurements showed that the most frequently revoked certificates are multi domain certificates. As explained in section 1.3.1, SAN certificates which occupy 18% of the records in the data set, may secure up to a hundred distinct subjects/domains behind a single group name.

The following list, enumerates the top 15 of the most frequently revoked certificates and CloudFlare's multi domain certificates cover 13 out of the 15.

1. incapsula.com
2. ssl2988.cloudflare.com
3. ssl4331.cloudflare.com
4. ssl3089.cloudflare.com
5. ssl2746.cloudflare.com

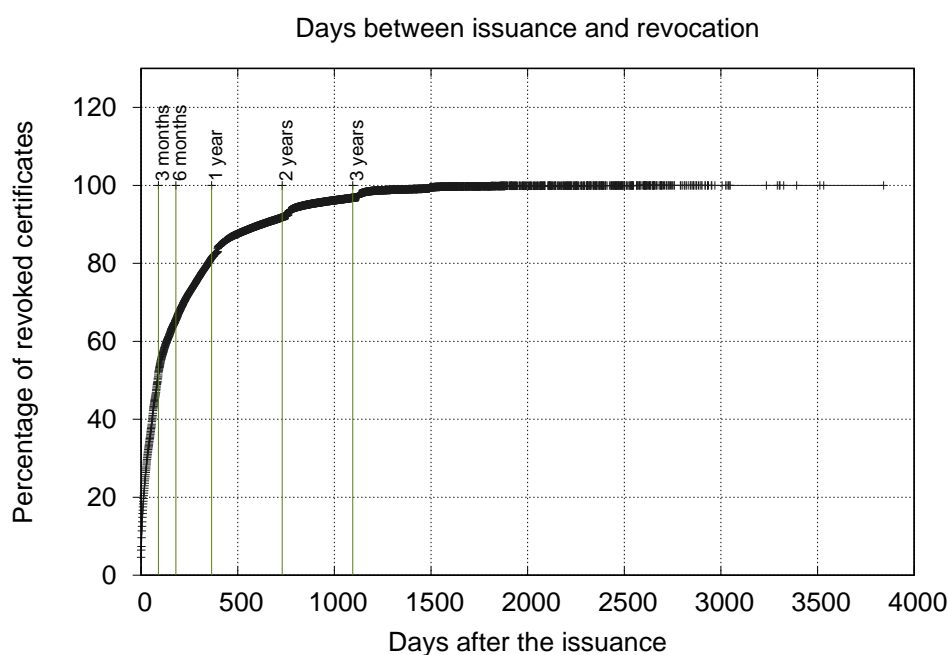


Figure 4.2: CDF of percentage of revocations after the day of issue of the certificate measured in days.

6. ssl4523.cloudflare.com
7. ssl3103.cloudflare.com
8. ssl4554.cloudflare.com
9. ssl3944.cloudflare.com
10. ssl3105.cloudflare.com
11. ssl4337.cloudflare.com
12. ssl2741.cloudflare.com
13. ssl3952.cloudflare.com
14. ssl4435.cloudflare.com
15. 360in.net

Based on their description, it is very logical that they have such high revocation rates, as for each change in one of the subjects within the same group, the Subject Alternative Name (SAN) certificate should be reissued. This, however, also hints that before new certificates are issued, the old ones are revoked. Section 4.2, measures how much this phenomenon occurs in the data.

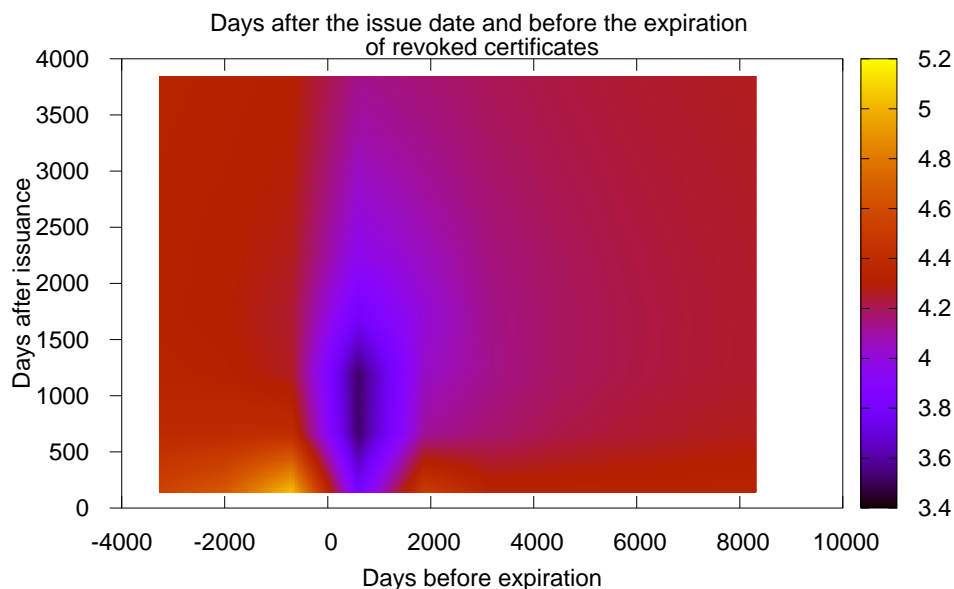


Figure 4.3: Revocations after the day of issuance and before the day of expiration of the certificate measured in days.

**Which issuers have more revoked certificates?** The issuing authorities which are more prevalent in our data set, do not appear to have the same order when it comes to revocation. For example, although, numbers indicate that Let’s Encrypt is the major certificate issuer (about 36%), it comes third to revoked certificates. Also, DigiCert which appears in the top 10 of issuers, has no record of revocation.

The top ranking of Globalsign and COMODO, is the result of their compromise in 2011 and for GoDaddy.com, Inc the great number of unqualified names and also the compromise of GoDaddy windows servers in 2014 [18]. Figure 4.4 shows the top 10 authorities, which have the largest records for revocations. Entrust, Starfield, In Common, cPanel, QuoVadis and Gandi, which are not in the to 10 of issuers, are in the top ten of revocations. On the other hand, Startcom, GeoTrust, and Thawte, are not prevalent to revocations’ top 10.

## 4.2 Revocation responsibility

As is was very well stated by Liu et al. [46] for the revocation to work both clients (browsers) and CAs must take action. Additionally, web administrators should be able to carefully maintain their websites. This section aims to confirm two basic hypotheses; whether people revoke their old certificates, after issuing new and

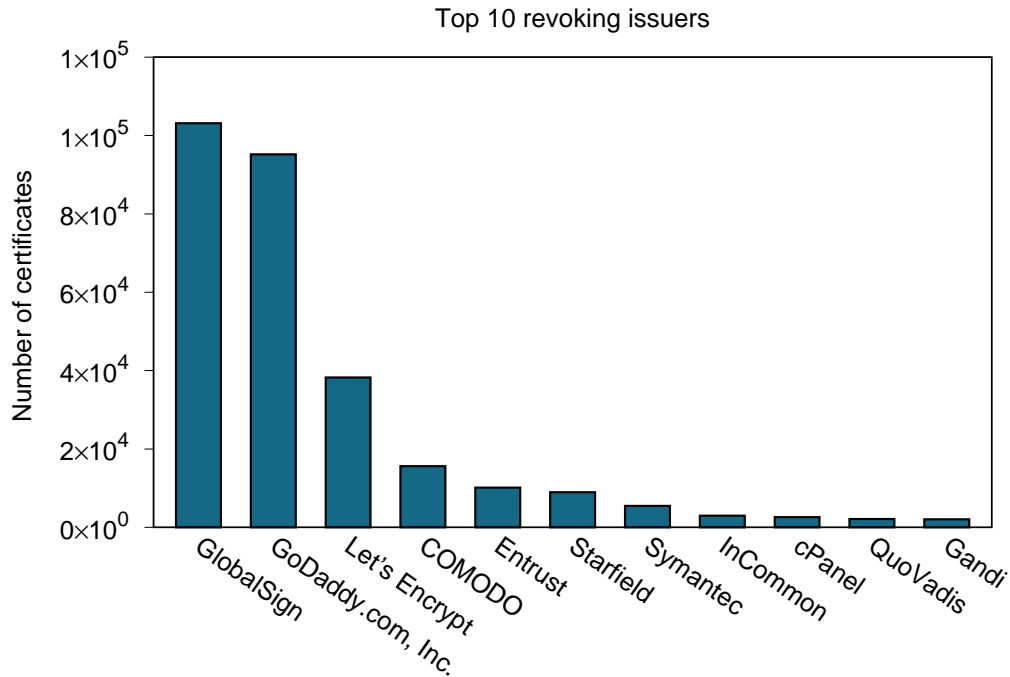


Figure 4.4: Top ten issuers of revoked certificates

whether high traffic sites tend to be more attentive to their certificates' status.

Furthermore, we test whether there is a correlation between revocations and known attacks against the SSL/Transport Layer Security (TLS) protocol as well as cyber attacks to evaluate whether web administrators re-evaluate their certificates' and revoke them if an attack compromises the integrity of their services.

**Do people revoke old certificates after issuing new ones?** Certificate revocations which can be associated with the issue of new certificates, constitute approximately 28% of all revocations. Figure 4.5 focuses on a period of 3 months, to show the number of observed revocations of invalid certificates, which associate to the issuance of new certificates.

If an attacker manages to steal the private key of a certificate, this certificate should be nullified via the revocation process. The issuance of a new certificate for a subject does not suggest that the previous one will be considered invalid, so the revocation process is not optional but mandatory.

**Do high traffic sites have greater revocation rates?** By September 2016, Alexa top 1 million had approximately 14% sites turning from Hypertext Transfer Protocol (HTTP) to HTTP over SSL (HTTPS) [54]. For a site to be secure though, a site administrator should not only make sure that a certificate is in place, but

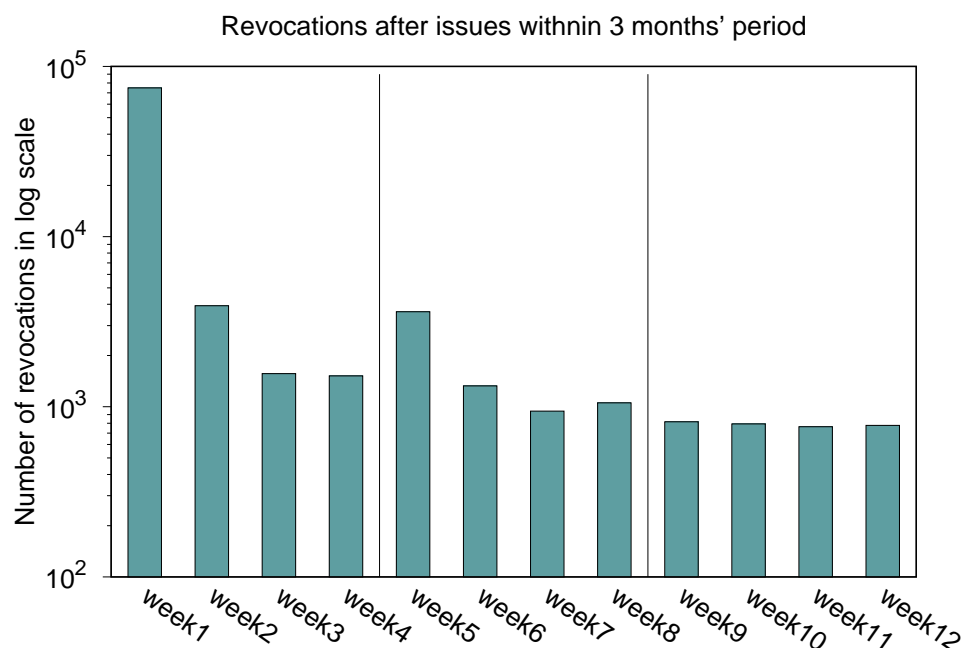


Figure 4.5: Number of revocations after issues within a period of 3 months divided to weeks

also that it is revoked when the need arises.

Focusing on the Alexa top 1000, we attempted to test if our hypothesis that high traffic sites are better maintained were true. Indeed, 223 sites in Alexa top 1000 were found to have at least one certificate revoked. Furthermore, we counted the sites which existed in Alexa 50.000 to 51.000, to remove any possible ‘noise’ and had at least one of their certificates revoked, and found them to be 44.

For those two cases, we plotted the days from issuance to revocation in Figure 4.6. In 4.6a one can observe a very distinct increase close to 200 days, this is mainly due to many Google domain certificates. Generally 4.6a shows an increasing behavior from 0 to approximately 600 days and 4.6b for 0 to 1200 days (approximately 3 years) there is increasing behavior which is compliant to the 4.2, which describes the general trend of revocations in the CT data set.

Furthermore in 4.6a and 4.6b the difference on the percentage of revocations is quite notable. In Alexa top 1000, there are approximately 23% of certificates revoked, while in Alexa top 50.000 to 51.000 the percentage barely reaches 4,5%. This again demonstrates that revocations are much more frequent to high traffic sites.

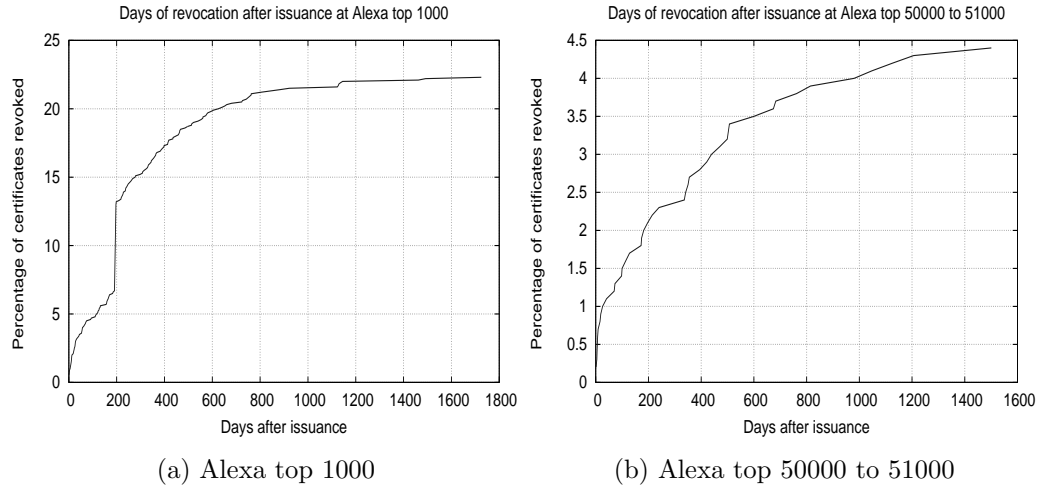


Figure 4.6: CDF of days of revocation in Alexa top 1000 and in Alexa range 50000 to 51000

### 4.3 Association of *hacks* and revocations

We use the term *hacks* in this context, to refer to all malicious acts meant to exploit weaknesses and vulnerabilities of the SSL protocol to perform an attack and also the generic web attacks. To test whether there is a correlation of *hacks* and the revocations numbers as depicted in Figure 4.1a, we cite information for known SSL based attacks based on [6]. Figure 4.7 shows the number of revoked certificates, with accordance to favorable and negative events for the SSL ecosystem that took place from 2011-01 to 2016-09.

Additionally, the data set regarding the quantity of cyber attacks in 2014, 2015 and 2016, which was provided by Hackmageddon [9], is demonstrated at Figure 4.8. These cyber attacks may refer to various incidents like hacking, poor security, SQLi, DDOS, leaks, etcetera.

In Table 4.1 we list the top ten months with highest revocation rates. In the first position, we see April 2014, which is when the Heartbleed vulnerability[42] was publicly announced and created a big wave of revocations and reissues[4]. It is also clear that 2016 is the most prevalent year in the top ten, which had the attacks of SLOTH (Security Losses from Obsolete and Truncated Transcript Hashes) [56], DROWN[64] and SWEET32[57].

One can observe that not all weakness/vulnerabilities or attacks can be directly associated with greater revocation rates. However, on the time periods that most revocations occur, the number of public announcements of vulnerabilities of the SSL certificate-protocol and generic cyber attacks, is also increased.

**Website hacks and revocation of their certificates** To further test the correlation of revocations to attacks, we used a data set provided by [www.hackmageddon.com](http://www.hackmageddon.com).

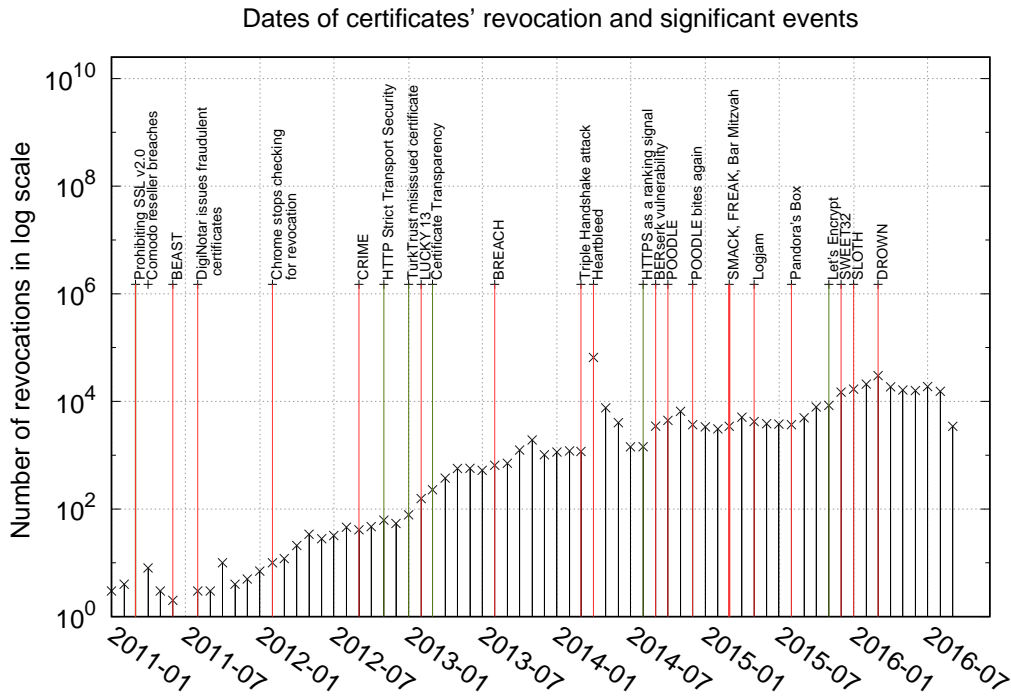


Figure 4.7: Certificates' revocation and significant events from 2011-01 to 2016-09

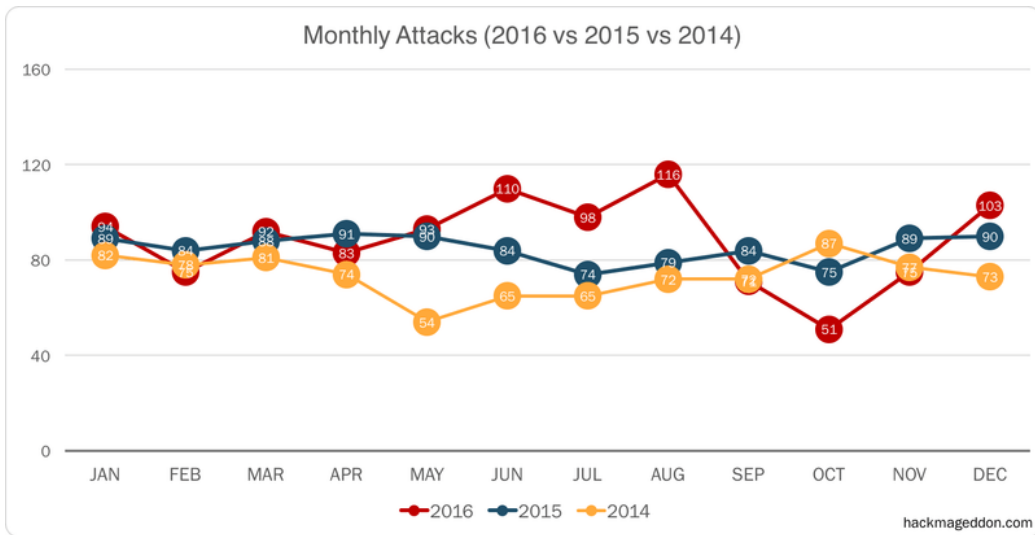


Figure 4.8: Monthly attacks chart from Hackmageddon

<http://www.hackmageddon.com/2017/01/19/2016-cyber-attacks-statistics/>



Time period	Revocations%	SSL based attacks	#Cyber attacks
04.2014	20%	Heartbleed	74
03.2016	9%	DROWN	92
02.2016	6.3%		75
07.2016	5.7%		98
04.2016	5.6%		83
01.2016	5%	SLOTH	94
05.2016	4.9%		93
06.2016	4.8%		110
08.2016	4.6%	SWEET32	116
12.2015	4.4%		95

Table 4.1: Time periods with most revocations and known attacks

com, which dates from September 2015 to August 2016. Based on the websites that were in some way hacked, we measured those who had at least one of their certificates revoked. The two main questions we attempted to answer were:

1. What percentage of the hacked websites had their certificates revoked?

To measure that, we used a threshold of two months period after the day of the hack of the site and the day of the revocation of its certificate. Out of 836 hacked records, 36 showed correlation with revoked certificates for the corresponding site, so the percentage is about 4,3%.

2. What percentage of the non-hacked websites had their certificates revoked?  
To find the non hacked sites we gathered all valid and revoked certificates of that time period and extracted the subjects which were not hacked at the same time. The non hacked sites are 9.191.927 and 181.979 of them were revoked at that same time period, so in this case, the percentage is less than 2%.

#### 4.3.1 What is the ranking of hacked sites in Alexa?

Throughout this study, we are using Alexa top sites to test whether more popular sites also have better security measures. To this end, we measured how many sites from the Alexa top 100, 1000, 5000 and 10000 have been hacked at least once, based on our prior measurement which indicated that a total of 836 sites being hacked from September 2015 to August 2016. Alexa top 10000 had approximately 18,2%, top 5000 13%, top 1000 7% and last top 100 had 19 hacked sites which approximate to 2,3

For the next Figures 4.9, we focus on the Alexa top 5000 correlated hacks. Assuming that the ranking of hacked sites in Alexa is  $k_1, k_2, \dots, k_n$ , plot 4.9a shows the time period between issue of the related certificate and its revocation

for sites which have been ranked as  $k_1+1, k_2+1, \dots, k_n+1$  (case  $k+1$ ) and 4.9b for sites which have been ranked as  $k_1+2, k_2+2, \dots, k_n+2$  (case  $k+2$ ). The time period of days between the issue and the revocation is the average for all the revoked certificates of a site, which are correlated with the hacks.

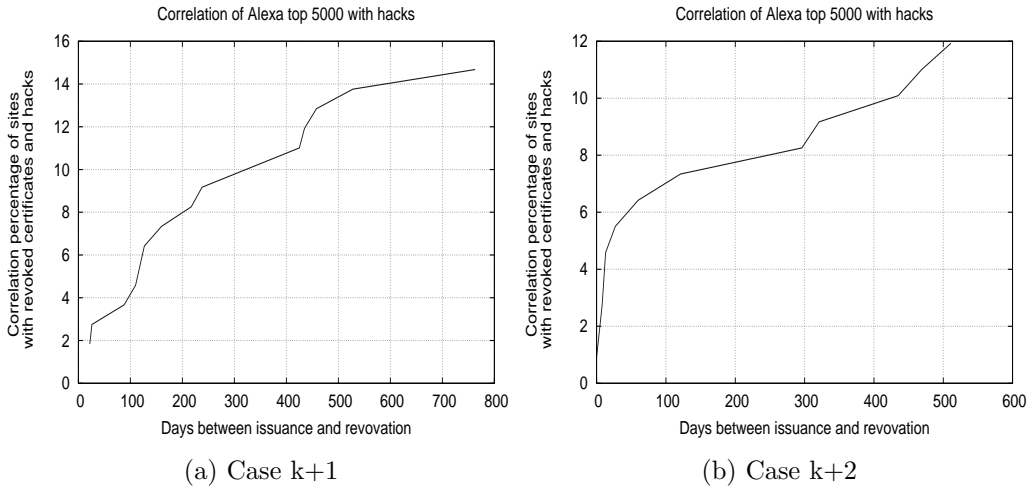


Figure 4.9: Days between the issue and the revocation for the Alexa top 5000 sites certificates which showed correlation with the hacks from September 2015 to August 2016

The first case:  $k+1$  has a slightly larger percentage of correlation of revocations and hacks than case:  $k+2$ . This means that sites which are very close to Alexa top sites, present similar behavior. The different time periods between the day of issuance and revocation, spans on case  $k+1$  to a little less than 800 days due to a few sites such as `adobe.com`, `avg.com`, `foxnews.com`, while on case  $k+2$ , to a little more than 500 days.

## 4.4 Revocation Protocols

As it has been stated many times, a certificate is meant to provide trust amongst those who use it. To ensure that it is not revoked, the most usual methods in place, are either via Certificate Revocation List (CRL), or Online Certificate Status Protocol (OCSP) servers. In its traditional mode, OCSP uses a cryptographic NONCE, which is a measure against replay attacks. An accelerated version is frequently applied on Content Delivery Networks (CDNs), which do not have the NONCE and may also lack freshness. Last, OCSP Stapling relies on the supplier of the Certification Authorities to assure the client that the certificate is valid.

Revocation protocols should ensure privacy, the freshness of information and do not pose too much overhead, neither for CAs nor for the browsers, otherwise, they will not be supported. OCSP, for instance, is supported by almost all major

browsers, apart from Google Chrome which disabled OCSP checks by default in 2012[76]. This section focuses on the aforementioned existing revocation protocols and two newly proposed protocols; DCSP and CSSP and compares them with their long-established equivalents.

#### 4.4.1 Certificate Revocation Lists (CRLs)

CRLs contain information about revoked certificates. To prevent such lists from being tampered a CRL file is signed by the CRL issuer. CAs are responsible for indicating the revocation status of the certificates that they issue and in most cases the CA is also the CRL issuer. The main drawback of CRLs is that they can grow too large in size[16], so most browsers do not download them as often as they should, resulting in stale certificate status information. An alternative to downloading the entire list is delta CRLs [45] which serve only the changes, based on the last owned complete CRL.

Another proposition is CRLsets[59], which are used by Google Chrome and the Chromium browser, for quickly blocking certificates in emergency situations and have a limited size of 250KB. If they have the related revocation information, they have great performance, but if not, they need to resort to an alternative approach. Additionally, Google has introduced Revocation Transparency (RT) [60], which follows the concept of Certificate Transparency, but focuses on the revocation status information and allows deletion. RT has still room for improvement since it has logarithmic overhead [73].

#### 4.4.2 Domain Name Server (DNS) based approaches

DCSP: Performant Certificate Revocation a DNS-based approach (DCSP) [38], uses the DNS system to store certificate revocation information. When a web browser is required to check whether a certificate has been revoked, it queries the DNS to find revocation information regarding that certificate. To ensure the authenticity of that information, each CA signs the revocation status of every certificate it has issued. To remedy the threat of possible replay attacks, DCSP employs epochs, where the information of each certificate is timestamped before signed. To mitigate the additional overhead this may impose, and to reduce the number of signatures that need to be performed in each epoch, we introduce the notion of collective records.

The DNS-based Authentication of Named Entities (DANE) [50] similar to DCSP also leverages DNS infrastructure to authenticate SSL certificates. DANE introduces a new type of DNS record, named TLSA, inside which it stores the whole certificate of a domain and uses Domain Name System Security Extensions (DNSSEC) to validate its integrity. According to a 2015 study [77] the adoption of DANE in the wild is yet far too low; from 485,000 collected signed zones, there were only 997 TLSA names. It is worth emphasizing, that if adopted, DANE can replace or highly limit the role of CAs.

Table 4.2 compares DANE and DCSP. They are very closely related, however, DCSP resolves to CAs for information regarding the revocation status of certificates and because it uses collective records this information is delivered very fast.

Method	Privacy	Low Latency	Freshness	Trusted Entity
DANE	✓	✓	✓	Administrator
DCSP	✓	✓✓	✓	CA

Table 4.2: Summarizing table of the comparison results: DANE - DCSP

#### 4.4.3 OCSP and CCSP

Let's assume that Bob wants to visit `https://example.com`. Since `example.com` is using HTTPS it needs to validate the status of the certificate. With *Original OCSP* [33] the browser connects to an OCSP responder, and he responds with the revocation status and a cryptographic nonce or an error message if it cannot process the request. The connection is secure as nonce can deter replay attacks, but connecting to a third party, results to partly revealing Bob's browsing history, to it and also that the loading time of `example.com` is increased. If *OCSP requests are served over CDNs* the response time is greatly reduced, as the requests are cached based on request history, but nonce is not used and the response may be stale. In case of cache misses or OCSP clients that create POST requests, the responses are slower because it falls back to traditional OCSP. In case of *OCSP Stapling* [49], the web server queries the OCSP responder, who replies with the status of the certificate and a digitally signed time-stamp. When Bob's web browser connects to the `example.com`, the server returns the SSL certificate with its stapled signed and time-stamped revocation status. The connection is secure, private and faster.

OCSP stapling may be the best alternative so far, but it forfeits freshness. CCSP: a Compressed Certificate Status Protocol (CCSP) [39], is a complementary protocol, which uses time and space compression techniques. Furthermore, it offers preservation of privacy and fast response to the user and due to its design, it requires a lower number of signatures than its equivalent solutions.

Table 4.3 compares OSCP variations to CCSP. In practice only OCSP stapling is a close match to CCSP, but in addition CCSP ensures freshness of information.

Method	Privacy	Fewer signatures	Low Latency	Freshness
OCSP-CDN	-	-	✓	≈
OCSP	-	-	-	✓
OCSP Stapling	✓	≈	✓	≈
CCSP	✓	✓✓	✓	✓

Table 4.3: Summarizing table of the comparison results: OCSP variations - CCSP

## Chapter 5

# Shift of paradigm and Conclusion

To this point, we have analyzed the use of certificates in web and mobile browsers. However, there is a ‘new’ paradigm for their application. The Internet of Things is a fast-emerging ecosystem of connected devices. Oxford dictionary formally defines them as: “A proposed development of the Internet in which everyday objects have network connectivity, allowing them to send and receive data”.

Due to their limitless applications, Internet of Things (IoT) has become a very popular trend with an estimated market of approximately 30 billion connected devices by 2020 according to IEEE Spectrum [65]. Without adequate protection though, these devices can become a real threat to privacy and security. This section mentions very briefly Public Key Infrastructure (PKI) and Secure Sockets Layer (SSL) based solutions to these problems.

### 5.1 Why is there a distinction ?

Connectivity may be included in home applications as well as critical security endpoints. In most cases, the information collecting and monitoring take place via a mobile or web application and is stored at some cloud service. As these devices become more integrated into our everyday lives, it is imperative to protect end users from eavesdropping or malicious attacks.

We make this distinction, as embedded devices differ from personal computers and even mobile devices, like smart phones and tablets, in terms of scale. The information shared may be much more personal and ergo more sensitive and significantly larger in number. Additionally, their computational capabilities are very limited, - low energy, and limited memory, and as a result the protocols are different. Figure 5.1 shows an overview of the IoT and web protocol stacks to indicate their differences. In essence, both mean to support the same principles, but the IoT stack supports more lightweight protocols.

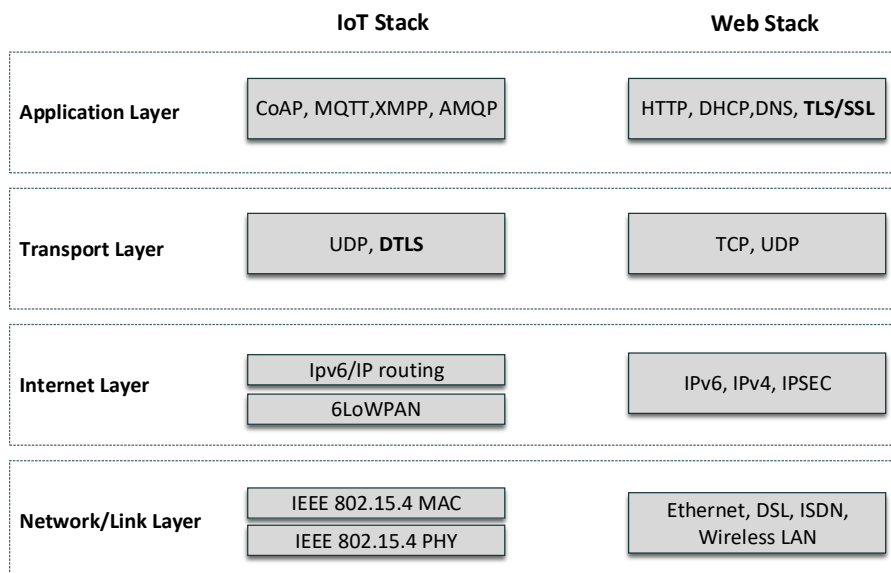


Figure 5.1: Comparison of IoT and Web protocol stacks

## 5.2 Transport layer problems

Numerous security experts agree that before IoT's great deployment, privacy and security have not been taken into account and as a result, such devices have many vulnerabilities and potential weaknesses. This is supported by numerous incidents of cyber security breaches based on such devices. In a research study by Hewlett Packard in 2015 [21] it was found that 70% of such devices did not encrypt communications to the Internet and local network.

As a future work, this thesis superficially addresses inadequate transport encryption implementation examples. Open Web Application Security Project (OWASP) [19] states that the most basic issues associated with 'Lack of Transport Encryption' on IoT are:

- Unencrypted Services via the Internet
- Unencrypted Services via the Local Network
- Poorly Implemented SSL/TLS
- Misconfigured SSL/TLS

### 5.2.1 Security relevant errors

As it has been stated before, although misconfigurations may not affect the operation of a certificate, they open ways for attacks and thus need to be fixed. In short, some very usual problems are: **(a)** Use of insecure protocols or features - all SSL versions are broken and should neither be used nor supported, the same applies to broken ciphers like DES-CBC-SHA or RC4-SHA, **(b)** Insecure certificate checks.

There many tools that check the state of a website's certificate and point out misconfigurations. The basic aspects that need to be checked are:

- Ciphersuite
- SSL & TLS Version
- RSA Key Size
- Certificate signature
- Vulnerabilities

### 5.2.2 Securing the Internet of things

Amongst the security mechanisms proposed for IoT are the PKI and the Transport Layer Security (TLS) protocol. However, due to the wide heterogeneity of connected systems, communication technologies as well as the unbounded number of interacting entities, existing technologies don't apply as efficiently as expected. SSL/TLS protocol proves to be burdensome and on several occasions, it is not correctly implemented or even used on IoT devices. According to a Symantec research [61] around 19 percent of all tested mobile applications that are used to control IoT devices do not use Secure Socket. In the same concept, Google's transparency report [25], indicated that mobile devices account for 95% of unencrypted end user traffic used in their services.

Moreover, millions of embedded devices use the same hard coded SSH cryptographic keys or HTTPS certificates, something that vastly increases the possibility that the encrypted data be recovered. According to a study in 2015,[58] the most common use of their static keys are:

- SSH host keys
- X.509 HTTPS certificates

The issue of trust is also present here and it resides to the basis of the devices' manufacturing stage. NIST [36] also, gives very detailed recommendation for key management.

### 5.3 Solutions

Succeeding the SSL/TLS architectural model there are solutions that are less demanding and more importantly tailored for the Things' limited computing resources. DTLS [5] which derives from the TLS protocol, provides the same security services Availability, Integrity and Confidentiality (AIC) but under the UDP protocol. Both, however, share known vulnerabilities, such as LUCKY 13. A variation of the above is GUARD TLS Toolkit [24], which poses minimum memory footprint and efficient RAM utilization. Additionally, mbed TLS (or as formerly known PolarSSL) [12] allows developers to include cryptographic and SSL/TLS capabilities in their (embedded) products.

PKI certificates may not be able to resolve all security problems [17], but in this context, they offer solutions for such devices [26]. Regarding certification revocation, NanoSSL [63] is proposed as a lightweight protocol for IoT devices' encrypted connections. Moreover, since DNS-based Authentication of Named Entities (DANE) and DCSP: Performant Certificate Revocation a DNS-based approach (DCSP) designs are based on simple DNS requests, they could serve as revocation protocols, with the right adjustments and refinements to their implementation. There are certainly, alternative solutions to secure the IoT, however here we favor, tested and well-documented options.

### 5.4 Conclusion

We move to an only HTTPS Internet, where PKI certificates are widely used, but not very well understood. This thesis' goal was to explain their role, their general trends and where they fail. With this intention we have analyzed and measured the basic components and key elements of a certificate like the secured entities, the issuers and their deployment. Moreover, we attempted to bring to light correlations to show how they are deployed, by showing the trends amongst high traffic sites and how they are related to attacks and vulnerabilities.

Additionally, we highlighted several cases where the mechanisms around SSL certificates are not implemented correctly and that the model of trust has some *fine print*, which should be carefully considered. For a malicious attacker, there are loopholes, such as vulnerabilities, that he can exploit and there is still a lot of work that should be done for the deficient validation process performed by Certification Authorities.

Furthermore, although, there are also mechanisms which could help the prevention of attacks, they are neglected. In this work, revocation trends and correlations are presented and discussed in depth to highlight the importance of status checking. Also, to investigate the main reasons revocation is unheeded, the assets and liabilities of existing and newly proposed methods are reviewed.

Last, we have superficially discussed how certificates can secure the IoT, which has several misconfiguration problems similar to Web and Mobile devices, but in



addition due to the nature of the devices' architecture it has less computational power and even more pervasive information. Hence, we placed emphasis on implementing security standards at manufacturing level and pointed that the solutions should be tailored to their needs and capabilities.



# Bibliography

- [1] Certificate transparency project. <https://www.certificate-transparency.org/>.
- [2] Digital signature. ”[https://en.wikipedia.org/wiki/Digital\\_signature](https://en.wikipedia.org/wiki/Digital_signature)”.
- [3] Hackmageddon, security statistics and analytics. <http://www.hackmageddon.com/category/security/cyber-attacks-statistics/>.
- [4] Heartbleed certificate revocation tsunami yet to arrive. <https://news.netcraft.com/archives/2014/04/11/heartbleed-certificate-revocation-tsunami-yet-to-arrive.html>.
- [5] Iot standards and protocols. <https://www.postscapes.com/internet-of-things-protocols/>.
- [6] Known vulnerabilities related attacks to the wolfssl embedded ssl/tls library. <https://wolfssl.com/wolfSSL/security/vulnerabilities.php>.
- [7] Let’s encrypt policies faq. <https://letsencrypt.org/docs/faq/>.
- [8] Map of cas. [https://www.eff.org/files/colour\\_map\\_of\\_cas.pdf](https://www.eff.org/files/colour_map_of_cas.pdf).
- [9] Monthly attacks chart, from hackmageddon. <http://www.hackmageddon.com/2017/01/19/2016-cyber-attacks-statistics/>.
- [10] Owasp guidelines on transport layer protection. [https://www.owasp.org/index.php/Transport\\_Layer\\_Protection\\_Cheat\\_Sheet](https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet).
- [11] Owasp ssl best practices. [https://www.owasp.org/index.php/SSL\\_Best\\_Practices](https://www.owasp.org/index.php/SSL_Best_Practices).
- [12] Polar ssl. <https://tls.mbed.org/>.
- [13] Public key certificate. <https://en.wikipedia.org/wiki/X.509>.
- [14] Ssl blacklist (sslbl). <https://sslbl.abuse.ch/>.
- [15] Erroneous verisign-issued digital certificates pose spoofing hazard. <https://technet.microsoft.com/en-us/library/security/ms01-017.aspx>, 2001.
- [16] Average crt size and download time. <http://unmitigatedrisk.com/?p=351>, 2013.

- [17] Certificate management for embedded systems. <https://realtimelogic.com/blog/2013/10/Certificate-Management-for-Embedded-Systems>, 2013.
- [18] <https://blog.sucuri.net/2014/12/iis-compromised-godaddy-servers-and-cyber-monday-spam.html>, 2014.
- [19] Internet of things top ten - 2014. <https://www.owasp.org/images/7/71/Internet\of\Things\Top\Ten\2014-OWASP.pdf>, 2014.
- [20] Types of ssl certificates - choose the right one. <https://www.symantec.com/connect/blogs/types-ssl-certificates-choose-right-one>, 2014.
- [21] Internet of things research study. <http://h20195.www2.hp.com/V4/GetDocument.aspx?docname=4AA5-4759ENW>, 2015.
- [22] Cwe-295: Improper certificate validation. <https://cwe.mitre.org/data/definitions/295.html>, 2017.
- [23] Ev ssl certificate guidelines. <https://cabforum.org/extended-validation/>, 2017.
- [24] Guard tls-tk, a compact tls/dtls stack for embedded security. <https://www.insidesecure.com/Products/Data-Communication/Secure-Communication-Toolkits/GUARD-TLS-TK>, 2017.
- [25] Https encryption on the web. <https://transparencyreport.google.com/https/overview?hl=en>, 2017.
- [26] Pki: The security solution for the internet of things. [https://www.digicert.com/wp-content/uploads/2017/05/Whitepaper\\_PKISolutionforIoT\\_4-12-17.pdf](https://www.digicert.com/wp-content/uploads/2017/05/Whitepaper_PKISolutionforIoT_4-12-17.pdf), 2017.
- [27] World internet usage and population statistics. <http://www.internetworldstats.com/stats.htm>, 2017.
- [28] Devdatta Akhawe, Bernhard Amann, Matthias Vallentin, and Robin Sommer. Here's my cert, so trust me, maybe?: Understanding tls errors on the web. In *Proceedings of the 22Nd International Conference on World Wide Web, WWW '13*, pages 59–70, New York, NY, USA, 2013. ACM.
- [29] Devdatta Akhawe and Adrienne Porter Felt. Alice in warningland: A large-scale field study of browser security warning effectiveness. In *Proceedings of the 22Nd USENIX Conference on Security, SEC'13*, pages 257–272, Berkeley, CA, USA, 2013. USENIX Association.

- [30] Mike Zusman Alexander Sotirov. Breaking the security myths of extended validation ssl certificates. <http://www.blackhat.com/presentations/bh-usa-09/SOTIROV/BHUSA09-Sotirov-AttackExtSSL-SLIDES.pdf>, 2009.
- [31] Rick Andrews. The importance of checking for certificate revocation. <https://casecurity.org/2013/03/08/the-importance-of-checking-for-certificate-revocation/>.
- [32] Kristen Beneduce Armisha Roberts and Margot Kimura. Ssl certificate black-listing. <https://www.osti.gov/scitech/servlets/purl/1373152>, 2016.
- [33] M. Myers at al. X.509 internet public key infrastructure online certificate status protocol - ocsp. <https://tools.ietf.org/html/rfc2560>, 1999.
- [34] E. Kasper B. Laurie, A. Langley and Google. rfc6962: Certificate transparency. <https://tools.ietf.org/html/rfc6962/>, 2013.
- [35] Zineb Ait Bahajji and Gary Illyes. Https as a ranking signal. <https://webmasters.googleblog.com/2014/08/https-as-ranking-signal.html>, 2014.
- [36] Elaine Barker. Recommendation for key management. <https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final#pubs-abstract-header>, 2016.
- [37] Marc Fischlin Benjamin Dowling, Felix Günther, and Douglas Stebila. A cryptographic analysis of the tls 1.3 handshake protocol candidates. <http://dl.acm.org/citation.cfm?id=2813653>, 2015.
- [38] Antonios A. Chariton, Eirini Degkleri, Panagiotis Papadopoulos, Panagiotis Iliia, and Evangelos P. Markatos. Dcsp: performant certificate revocation a dns-based approach.
- [39] Antonios A. Chariton, Eirini Degkleri, Panagiotis Papadopoulos, Panagiotis Iliia, and Evangelos P. Markatos. Ccsp: a compressed certificate status protocol. 2017.
- [40] Jeremy Clark and Paul C van Oorschot. Sok: Ssl and https: Revisiting past challenges and evaluating certificate trust model enhancements. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 511–525. IEEE, 2013.
- [41] T. Dierks. The transport layer security (tls) protocol version 1.2. <https://tools.ietf.org/html/rfc5246>.
- [42] Zakir Durumeric, James Kasten, David Adrian, J. Alex Halderman, Michael Bailey, Frank Li, Nicolas Weaver, Johanna Amann, Jethro Beekman, Mathias Payer, and Vern Paxson. The matter of heartbleed, 2014.

- [43] Carl Ellison and Bruce Schneier. Ten risks of pki: What you're not being told about public key infrastructure. <https://www.schneier.com/academic/paperfiles/paper-pki-ft.txt>.
- [44] Martin Georgiev et al. The most dangerous code in the world: validating ssl certificates in non-browser software. <http://dl.acm.org/citation.cfm?id=2382204>, 2012.
- [45] R. Housley et al. Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. <https://www.ietf.org/rfc/rfc5280.txt>, 2002.
- [46] Yabing Liu et al. An end-to-end measurement of certificate revocation in the web's pki. <http://conferences2.sigcomm.org/imc/2015/papers/p183.pdf>, 2015.
- [47] Adrienne Porter Felt, Richard Barnes, April King, Chris Palmer, Chris Bentzel, and Parisa Tabriz. Measuring https adoption on the web. 2017.
- [48] Gennie Gebhart and Seth Schoen. Is let's encrypt the largest certificate authority on the web? <https://www.eff.org/deeplinks/2016/10/lets-encrypt-largest-certificate-authority-web>, 2016.
- [49] Robert Gibb. What is ocsf stapling? <https://www.maxcdn.com/one/visual-glossary/ocsp-stapling/>, 2016.
- [50] P. Hoffman and J. Schlyter. The dns-based authentication of named entities (dane) transport layer security (tls) protocol: Tlsa. <https://tools.ietf.org/html/rfc6698>, 2012.
- [51] Ralph Holz, Lothar Braun, Nils Kammenhuber, and Georg Carle. The ssl landscape: a thorough analysis of the x.509 pki using active and passive measurements. In *Internet Measurement Conference*, 2011.
- [52] Erich Nahum Homin K. Lee, Tal Malkin. Cryptographic strength of ssl/tls servers: Current and recent practice. <http://www.cs.columbia.edu/~homin/papers/psst/LeeMalNah-2007.pdf>, 2007.
- [53] Lin Shung Huang, Alex Rice, Erling Ellingsen, and Collin Jackson. Analyzing forged ssl certificates in the wild. In *Security and privacy (sp), 2014 ieee symposium on*, pages 83–97. IEEE, 2014.
- [54] Troy Hunt. Https adoption has reached the tipping point. <https://www.troyhunt.com/https-adoption-has-reached-the-tipping-point/>, 2017.

- [55] Nick Hunter. Wildcard certificates make encryption easier, but less secure. <https://www.venafi.com/blog/wildcard-certificates-make-encryption-easier-but-less-secure>, 2017.
- [56] Gaetan Leurent Karthikeyan Bhargavan. Security losses from obsolete and truncated transcript hashes, cve-2015-7575. <https://www.mitls.org/pages/attacks/SLOTH>.
- [57] Gaetan Leurent Karthikeyan Bhargavan. Sweet32: Birthday attacks on 64-bit block ciphers in tls and openvpn. On the Practical (In-)Security of 64-bit Block Ciphers - Collision Attacks on HTTP over TLS and OpenVPN.
- [58] Swati Khandelwal. Millions of iot devices using same hard-coded crypto keys. <http://thehackernews.com/2015/11/iot-device-crypto-keys.html>, 2015.
- [59] Adam Langley. Revocation checking and chrome's crl. <https://www.imperialviolet.org/2012/02/05/crlsets.html>.
- [60] Ben Laurie and Emilia Kasper. Revocation transparency. <http://www.links.org/files/RevocationTransparency.pdf>.
- [61] Candid Wueest Mario Ballano Barcena. Insecurity in the internet of things. <https://www.symantec.com/content/dam/symantec/docs/whitepapers/insecurity-in-the-internet-of-things-en.pdf>, 2015.
- [62] Michael Mimoso. Half of chrome pageloads are https. <https://threatpost.com/half-of-chrome-pageloads-are-https/121798/>, 2016.
- [63] whitepaper Mocana. The hidden cost of free openssl. <https://www.mocana.com/iot-security/nanossll>.
- [64] et al. Nimrod Aviram. Drown: Breaking tls using sslv2. <https://drownattack.com/drown-attack-paper.pdf>.
- [65] Amy Nordrum. Ieee tech talk. <http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated>, 2016.
- [66] Yvette O'Meally. Recommendations for pki key lengths and validity periods with configuration manager. Microsoft Enterprise Mobility and Security Blog, 2009.
- [67] Pierluigi Paganini. Ssl replacement? convergence for replacing ca... maybe. <http://securityaffairs.co/wordpress/151/digital-id/ssl-replacement-convergence-for-replacing-ca-maybe.html>, 2011.
- [68] Chris Palmer. Unqualified names in the ssl observatory, technical analysis. <https://www.eff.org/deeplinks/2011/04/unqualified-names-ssl-observatory>, 2011.

- [69] Ivan Ristic. Ssl and tls deployment best practices. <https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices>, 2017.
- [70] Ivan Ristic. Ssl/tls and pki history. <https://www.feistyduck.com/ssl-tls-and-pki-history/>, 2017.
- [71] Ph.D. Rolf Oppliger. Ssl and tls: Theory and practice. <http://swrdfish.github.io/assets/ssl/SSLandTLSTheoryandPractice.pdf>, 2009.
- [72] Steven B. Roosa and Stephen Schultze. The “certificate authority” trust model for ssl: A defective foundation for encrypted web traffic and a legal quagmire. <https://search.proquest.com/openview/8487a898bc9a0c6234511f41e90ade6a/1>, 2010.
- [73] Mark Dermot Ryan. Enhanced certificate transparency and end-to-end encrypted mail. 21st Annual Network and Distributed System Security Symposium, NDSS 2014, 2014.
- [74] Jerome Segura. Digital certificates and malware: a dangerous mix. <https://blog.malwarebytes.com/threat-analysis/2013/02/digital-certificates-and-malware-a-dangerous-mix/>, 2013.
- [75] Wordpress HTTPS support. Why do i see tls.automattic.com in my certificate’s common name (cn)? <https://en.support.wordpress.com/https/>.
- [76] Wikipedia. Online certificate status protocol. [https://en.wikipedia.org/wiki/Online\\_Certificate\\_Status\\_Protocol#Browser\\_support](https://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol#Browser_support).
- [77] Liang Zhu, Duane Wessels, Allison Mankin, and John Heidemann. Measuring dane tlsa deployment. In *International Workshop on Traffic Monitoring and Analysis*, pages 219–232. Springer, 2015.