UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Certificate Status Information Distribution and Validation in Vehicular Networks

by

Carlos Hernández Gañán

Advisor: José L. Muñoz Tapia
Co-Advisor: Oscar Esparza Martín

A thesis presented to the Department of
Telematics Engineering
in fulfillment of the requirement for the Degree of
Doctor of Philosophy
of the
Universitat Politècnica de Catalunya (UPC)

Barcelona, Spain, July 2013

Acknowledgements

Words fall short to express my gratitude and appreciation to my advisor José Luis Muñoz and Oscar Esparza, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate their vast knowledge and skills in many areas, and their assistance in writing reports. Above all and the most needed, he provided me unflinching encouragement and support in various ways. His truly scientist intuition has made him as a constant oasis of ideas and passions in science, which exceptionally inspire and enhance my growth as a student, a researcher and a scientist want to be.

I would like also to acknowledge the work carried out by the examining committee members. I appreciate their time and effort devoted to review and analyze my Ph.D. thesis and providing me with their insightful suggestions and invaluable comment, thereby further improving the quality of this thesis.

I am deeply indebted to Jonathan Loo. Without his guidance, support and good nature, I would never have been able to develop this thesis successfully. I benefited greatly from his ideas and insights. His involvement with his originality has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

Some debts are hard to put into words. My research colleagues Juan Caubet, Sergi Reñé, Jorge Mata, Juan José Alins all know why their names are here.

My last, but not least gratitude is for my parents, it is difficult to find words to express my gratitude and thanks to both of you.

I realize that not all people who contributed either directly or indirectly to my study are mentioned in this page. From the deepest of my heart, I would like to thank all of you...

Abstract

Vehicular ad hoc networks (VANETs) are emerging as functional technology for providing a wide range of applications to vehicles and passengers. Ensuring secure functioning is one of the prerequisites for deploying reliable VANETs. However, the open-medium nature of these networks and the high-speed mobility of a large number of vehicles harden the integration of primary security requirements such as authentication, message integrity, non-repudiation, and privacy.

Without security, all users would be potentially vulnerable to the misbehavior of the services provided by the VANET. Hence, it is necessary to evict compromised, defective, and illegitimate nodes. The basic solution envisioned to achieve these requirements is to use digital certificates linked to a user by a trusted third party. These certificates can then be used to sign information. Most of the existing solutions manage these certificates by means of a central Certification Authority (CA). According to IEEE 1609.2 standard, vehicular networks will rely on the public key infrastructure (PKI). In PKI, a CA issues an authentic digital certificate for each node in the network. Therefore, an efficient certificate management is crucial for the robust and reliable operation of any PKI. A critical part of any certificate-management scheme is the revocation of certificates. The distribution of certificate status information process, as well as the revocation process itself, is an open research problem for VANETs.

In this thesis, firstly we analyze the revocation process itself and develop an accurate and rigorous model for certificate revocation. One of the key findings of our analysis is that the certificate revocation process is statistically self-similar. As none of the currently common

formal models for revocation is able to capture the self-similar nature of real revocation data, we develop an autoregressive Fractional-integrated moving average (ARFIMA) model that recreates this pattern. Neglecting the self-similarity of the revocation process leads to inefficient revocation release strategies. With synthetic revocation traces, current revocation schemes can be improved by defining more accurate revocation data issuance policies. We show that traditional mechanisms that aim to scale could benefit from these traces to improve their updating strategies.

Secondly, we analyze how to deploy a certificate status checking service for mobile networks and we propose a new criterion based on a risk metric to evaluate cached status data. With this metric, the PKI is able to code information about the revocation process in the standard certificate revocation lists. Thus, users can evaluate a risk function in order to estimate whether a certificate has been revoked while there is no connection to a status checking server. Moreover, we also propose a systematic methodology to build a fuzzy system that assists users in the decision making process related to certificate status checking.

Thirdly, we propose two novel mechanisms for distributing and validating certificate status information (CSI) in VANET. This first mechanism is a collaborative certificate status checking mechanism based on the use based on an *extended-CRL*. The main advantage of this *extended-CRL* is that the road-side units and repository vehicles can build an efficient structure based on an authenticated hash tree to respond to status checking requests inside the VANET, saving time and bandwidth. The second mechanism aims to optimize the trade-off between the bandwidth necessary to download the CSI and the freshness of the CSI. This mechanism is based on the use of a hybrid delta-CRL scheme and Merkle hash trees, so that the risk of operating with unknown revoked certificates remains below a threshold

during the validity interval of the base-CRL, and CAs have the ability to manage this risk by setting the size of the delta-CRLs. For each of these mechanism, we carry out security analysis and performance evaluation to proof the efficiency and reliable security of the proposed schemes.

Finally, we also analyze the impact of the revocation service in the certificate prices. We model the behavior of the oligopoly of risk-averse certificate providers that issue digital certificates to clients facing identical independent risks. We found the equilibrium in the Bertrand game. In this equilibrium, we proof that certificate providers that offer better revocation information are able to impose higher prices to their certificates without sacrificing market share in favor of the other oligarchs.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today's transportation systems face serious challenges in terms of road safety, efficiency and environmental friendliness. With a huge improvement in technological innovations, Vehicular Communication (VC) emerges as a solution to palliate many issues of our modern day communication system in roads. This type of communication involves the use of short-range radios in each vehicle. This technology allows various vehicles to communicate with each other which is also known as (V2V) communication and with road side infrastructure (V2I) communication. Vehicular communication systems (VCS) are a direct response to the increasing demands of Intelligent Transportation Systems (ITS) services and the expectations of the automotive industry. In this sense, vehicular communication is designed for a wide range of applications related to safety, traffic management, and passenger comfort.

Safety applications are the main motivation for the development of these systems. They are conceived to spread accurate data quickly and reliably, in order to avoid accidents and life losses. In this sense, vehicles collaborate to avoid accidents, e.g., they disseminate emergency warning messages when a hazardous status is detected, such as slippery road conditions. In the same way, VCSs improve road safety by enabling traffic lights and signs to communicate with vehicles. In addition to these safety applications, VCSs are also employed in a variety of ITS traffic management applications. Road traffic management applications focus on optimizing traffic flow in order to avoid traffic congestion, to reduce travel time, and to use the transportation infrastructure effectively. A third type

of applications relates to the comfort and well-being of passengers, named infotainment applications. Infotainment applications provide additional information or entertainment to the passengers and/or driver, e.g. multimedia services, radio channels, Internet connection or advertises from some local merchants or gas stations.

Vehicular networks have attracted the attention of both academic and industrial communities, which is reflected in the interest of governments and standardization organizations. For example, European car manufacturers have instituted the Car-to-Car Communication Consortium (C2C-CC) [2] to improve road safety and efficiency, and the U.S FCC (Federal Communication Commission) has approved a 75 MHz spectrum for vehicular networks [3]. The Institute of Electrical and Electronics Engineers (IEEE) also supports vehicular communication with the IEEE 1609 family of standards for wireless access in vehicular environments (WAVE) [4]. Previous works present approaches that employ various technologies for the implementation of VCS. In this way, several car manufacturers support their vehicles through Internet access via cellular networks. However, using cellular networks is not the best way to build a VCS in terms of cost and latency. In many proposals, standard IEEE 802.11 is deployed for a VCS. However, this protocol has a limited radio range and needs numerous base stations to maintain the vehicles connected to the infrastructure. Using Vehicular Ad Hoc Network (VANET) with On-Board Units (OBUs) and Roadside Units (RSUs) appears to be the more effective method, but it also entails significant challenges. VANETs also enable multi-hop routing through vehicles to reach the infrastructure. Nevertheless, without securing these networks, damage to property and life can be done at a greater extent.

Simple and effective security mechanisms are the major problem of deploying VANET in public. Without security, a VANET is wide open to a number of attacks such as propagation of false warning messages as well as suppression of actual warning messages, thereby causing accidents. This makes security a factor of major concern in building such networks.

1.1 Research Motivation

VANETs deployment will not occur without assuring secure communications. As a special case of mobile ad hoc networks (MANETs), VANETs inherit all of MANETs security concerns while introducing additional security challenges specific to their characteristics. In VANETs, attackers could forge, inject, replay and drop messages in order to violate user privacy, information integrity, authenticity, and system performance.

In order to secure a VANET, the following security requirements should be met [5]:

- *Authentication*: Entity authentication is required to ensure that the communicating entities are legitimate. In addition, data authentication is also a concern to ensure that the contents of the received data is neither altered nor replayed.

- *Non-repudiation*: Non-repudiation is necessary to prevent legitimate users from denying the transmission or contents of their messages.

- *Privacy*: Preserving users' privacy is necessary to prevent the disclosure of their location information and real identities.

- *Access control*: Access control is required to delimitate the operations that any entity in the network is allowed to perform. Moreover, any misbehaving entity should be removed from the network to protect other legitimate entities. In addition, any action taken by those misbehaving entities should be repealed.

- *Availability*: Users may be frustrated if VANET services become temporarily unavailable due to attacks such as DoS attacks.

Without security, all users would be potentially vulnerable to the misbehavior of the services provided by the VANET. Hence, it is necessary to evict compromised, defective, and illegitimate nodes. The basic solution envisioned to achieve these requirements is to use digital certificates linked to a user by a trusted third party. These certificates can then be used to sign information. Most of the

existing solutions manage these certificates by means of a central Certification Authority (CA) [6, 7]. According to IEEE 1609.2 standard [8], vehicular networks will rely on the public key infrastructure (PKI). In PKI, a CA issues an authentic digital certificate for each node in the network. Therefore, an efficient certificate management is crucial for the robust and reliable operation of any PKI. A critical part of any certificate-management scheme is the revocation of certificates.

To revoke a vehicle in PKI, a certificate revocation list (CRL) has to be issued by the trusted authority (i.e., centralized revocation) and broadcast by the infrastructure RSUs. The centralized certificate status checking process in the traditional PKI may be unfeasible in the large scale VANETs due to the following reasons: (1) Each CA handles a large number of CRL requests that can create a bottle-neck; (2) The CRL downloading process is lengthy compare to the short V2I communication duration between the RSUs and the highly mobile vehicles during which the updated CRL has to be transmitted to the requesting vehicles. This long delay is due to the fact that any request sent by a vehicle to any RSU must be forwarded to the CA, and CA has to send the most updated CRL to that RSU which in turn forwards this CRL to the requesting vehicles. Consequently, the traditional PKI has to be optimized to satisfy the revocation service requirements for vehicular communication scenarios. To design a functional revocation service for VANETs, it is required for each OBU to efficiently check the status of any certificate in a timely manner.

Additionally, while wired networks can guarantee on demand connectivity between the CA and principals in the wired network for obtaining the current certificate status information, VANETS cannot guarantee such on demand contact at all times due to the sporadic connectivity. Such on demand connectivity with the centralized entity is essential for recipients to have confidence on the security infrastructure. Hence, the risk on trusting outdated certificate status information while being disconnected from the infrastructure needs to be quantified.

1.2 Objective of the Thesis

This thesis aims to mitigate the issues of conveying certificate status information over vehicular networks. Therefore, the objectives of this thesis are as follows:

1. Modeling the revocation process to perform analytic and empirical evaluations. This modeling should allow users and authorities to predict when a revocation is prone to occur. Moreover, the resulting model should also serve as evaluation tool to generate synthetic revocations.

2. Evaluating the performance of current certificate status mechanisms in vehicular networks. This involves analyzing the impacts of sporadic connectivity with the security infrastructure on the security performance of the proposed security mechanisms.

3. Proposing a metric to quantify the risk of operating while being disconnected from the infrastructure. This metric should take into account all the information about the revocation process available to the certification authority.

4. Designing a new certificate status validation mechanism for vehicular networks. This mechanism should take into account the knowledge derived from the revocation process modeling, and use the criterion to measure the risk of operating while disconnected from the infrastructure.

Chapter 2

Results

This section summarizes the main contributions of this thesis aligned with the objectives detailed in the previous section. Throughout the research process that involves the development of the thesis, several results have been obtained. These results have been validated by the international scientific community through the assessment of papers published in high-ranked journals and international conferences. Each contribution is briefly described in the following sections and appended at the end of this document.

2.1 Analysis and modeling of the revocation process

Most of the effort on analyzing certificate revocation has been mainly put on studying the trade-offs that can be achieved when dealing with different revocation mechanisms [9, 10, 11, 12]. These studies aim to compare the performance of different revocation mechanisms in different scenarios. Recently however, there have appeared some studies like [13, 14, 15] that can be considered a first step towards understanding the revocation process itself. These studies have mainly analyzed the probability distribution of certificate revocation requests. However, these later studies do not capture the time evolution of the revocation process or provide a means to efficiently forecast revocation events.

A revocation method is selected by an organization based on the cost, infrastructure, and volumes of transactions that are expected. To gauge these costs, different revocation mechanisms are tested under the assumption that the revocation events follow a specific probability distribution. Most theoretical frameworks and simulation studies for performance evaluation assume that the temporal distribution of queries follows a Poisson distribution and using this, organizations can estimate the infrastructure needed to deploy the PKI and the associated costs. However, in this thesis, we have demonstrated that revocation data is statistically *self-similar*, that none of the commonly used revocation models is able to capture this fractal behavior, and that such behavior has serious implications for the design, control, and analysis of revocation mechanisms such as CRLs.

We started by analyzing the validity of Poisson-like process assumption. We used publicly available CRLs from different certification authorities (containing more than 300,000 revoked certificates over a period of three years (see Table 2.1)). Our analysis demonstrated that the Poisson distribution fails to capture the statistical properties of the actual revocation process. We also saw that the Poisson distribution grossly under-estimates the bandwidth utilization of the revocation mechanism. At first glance, this might look like an obvious result, since after all as a memoryless process, Poisson distribution cannot be expected to model periodic trends like daily, weekly and monthly cycles in revocation rates. We showed however that the modeling inability transcends simple cycles. In particular, we showed self-similarity has a severe detrimental impact on the revocation service performance.

Issuer Name	Number of Revoked Certificates	Last Update	Next Update
GoDaddy	932,900	2012/02/01	2012/02/03
VeriSign	5,346	2012/02/02	2012/02/16
Comodo	2,727	2012/02/03	2012/02/06
GlobalSign	7,591	2012/02/02	2012/03/03
Thawte	8,061	2012/02/01	2012/02/16

Table 2.1: Description of the collected CRLs.

Results of our analysis, including burstiness at all scales, strongly indicate self-similar nature of revocation events. In Figure 2.1 we can observe different evident trends; (i) Burstiness in all time scales: the burstiness of the revocation process does not disappear when changing the time scales. (ii) Lack of natural length of bursts: The figure shows burstiness ranging from days to months. Note that the full duration of the figure with the largest time slot is 1,000 days, and some of the bursts have many hours of duration.

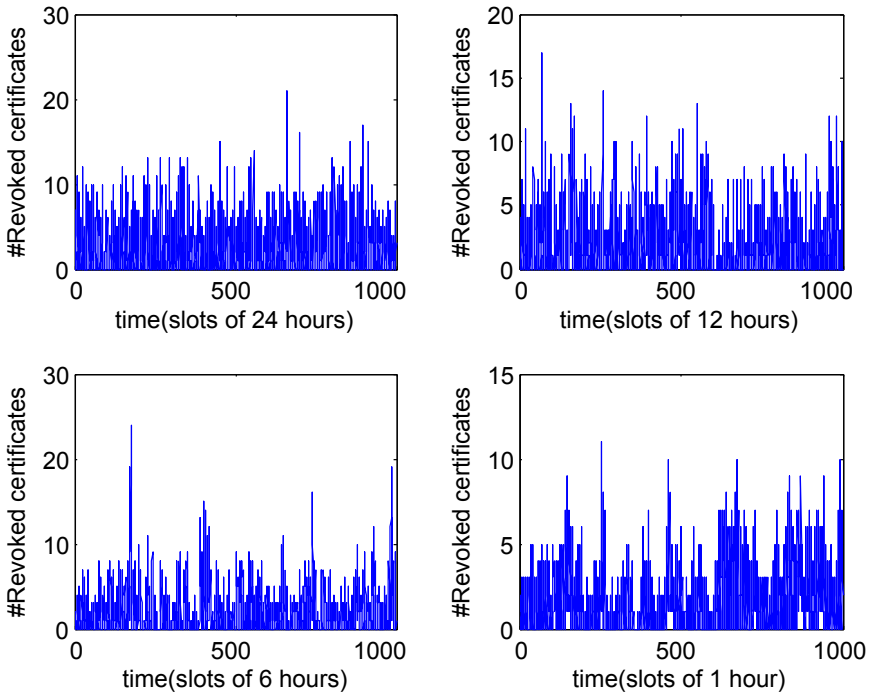


Figure 2.1: Revocation Bursts over Four Orders of Magnitude.

We confirmed this by analyzing the autocorrelation of the revocation process and estimating the Hurst parameter for the observed distribution and showing that the estimates validate self-similar nature of the revocation lists. First of all, we started analyzing the autocorrelation of the revocation data. Recall that in a self-similar process autocorrelations decay hyperbolically rather than exponentially fast, implying a nonsummable autocorrelation function $\sum_k r(k) = \infty$ (long-range dependence or LRD). For the frame data, the empirical autocorrelation functions $r(k)$ are shown in Fig. 2.2, with lag k ranging from 0 to 100. Notice

that $r(k)$ decreases slower than exponentially no matter the CA. The curve does decay toward zero, but it does so extremely slowly. The very slowly decaying autocorrelations are indicative of LRD.

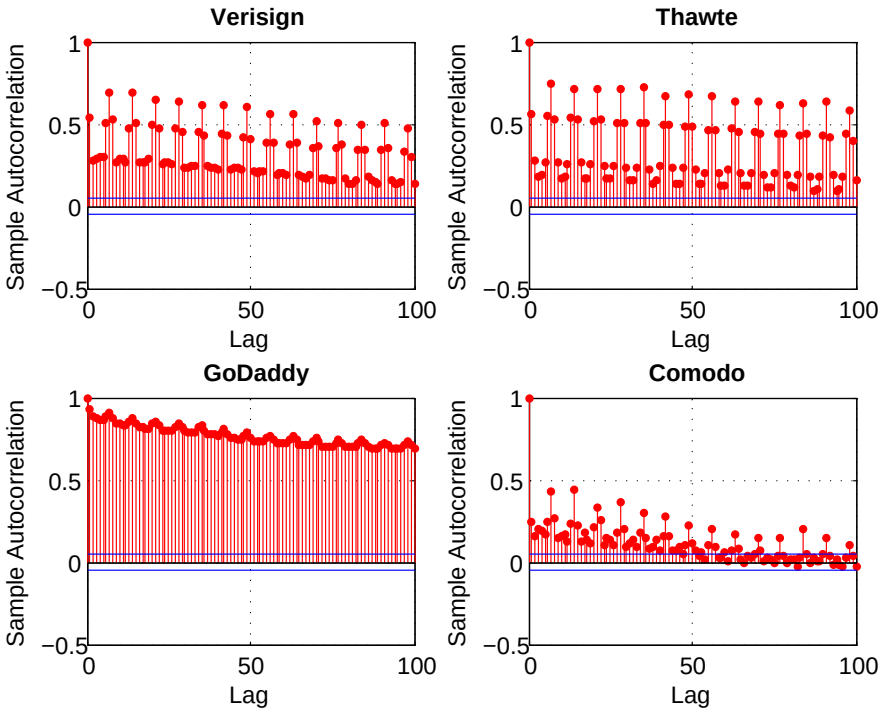


Figure 2.2: Autocorrelation function of the revocation process per CA.

Then, we used five different methods for assessing self-similarity: the variance-time plot, the rescaled range (or R/S) plot, the periodogram plot, the Detrended Fluctuation Analysis (DFA) plot and the Whittle estimator. We concentrated on individual months from our revocation time series, so as to provide as nearly a stationary dataset as possible. To provide an example of these approaches, analysis of a single month from GoDaddy revocation data is shown in Figure 2.3. The figure shows plots for the four graphical methods: variance-time (upper left), rescaled range (upper right), periodogram (lower left) and DFA (lower right). The variance-time plot is linear and shows a slope that is distinctly different from -1 (which is shown for comparison); the slope is estimated using regression as -0.077, yielding an estimate for H of 0.96. The R/S plot shows an asymptotic slope that is different from 0.5 and from 1.0 (shown for comparison); it is estimated

using regression as 0.95, which is also the corresponding estimate of H. The periodogram plot shows a slope of -0.14 (the regression line is shown), yielding an estimate of H as 0.83. Finally, the Whittle estimator for this revocation data (not a graphical method) yields an estimated Hurst value of 0,923 with a 95% confidence interval of $(0.87, 0.95)$.

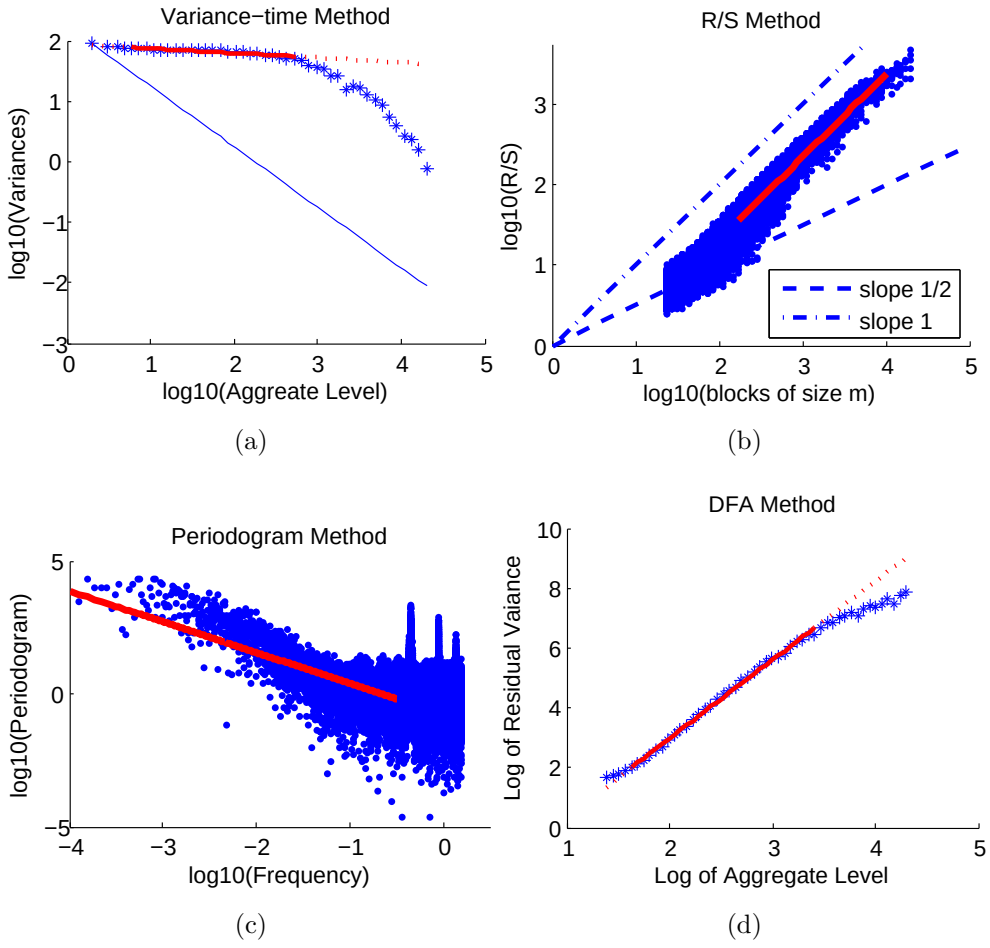


Figure 2.3: Graphical methods for checking for self-similarity of the revocation process from GoDaddy (a) variance-time plot, (b) pox plot of R/S, (c) periodogram plot, and (d) DFA plot.

Beyond invalidating Poisson-like distributions, this proof of self-similarity has important implications on CA utilization, throughput, and certificate stratus checking time. Intuitively, as the revocation process is bursty (non-uniformly

distributed) the CA will be partially idle during low burst periods and vice versa. Thus, the revocation lists will grow non-uniformly, and current updating policies will result bandwidth inefficient.

After proving the selfsimilar nature of the revocation process, we went a step further by developing an accurate and rigorous model for certificate revocation process. The proposed model is based on an autoregressive fractionally integrated moving average (ARFIMA) process [16], which provides an accurate and parsimonious model for revocation.

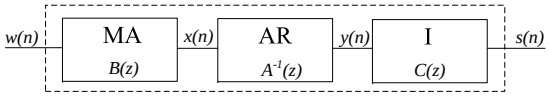


Figure 2.4: Components of an ARFIMA process.

Figure 2.4 shows a scheme of the ARFIMA model, where the components of each bloc are:

$$\begin{aligned} A(z) &= 1 - 0.6467z^{-1} + 0.02693z^{-2} + 0.09085z^{-3} + 0.09753z^{-4} + 0.1218z^{-5} + 0.1991z^{-6} \\ &\quad - 0.804z^{-7} + 0.6906z^{-8} + 0.03223z^{-9} - 0.04807z^{-10} - 0.007471z^{-11} - 0.0759z^{-12} \\ &\quad - 0.08934z^{-13} - 0.07605z^{-14} - 0.006487z^{-15} - 0.02565z^{-16} - 0.01994z^{-17} - 0.04003z^{-18} \\ &\quad - 0.05007z^{-19} - 0.01331z^{-20} - 0.07361z^{-21} - 0.001947z^{-22} - 0.02836z^{-23} - 0.01824z^{-24} \\ &\quad - 0.03693z^{-25} + 0.007019z^{-26} - 0.07691z^{-27} - 0.01872z^{-28} - 0.03821z^{-29}, \qquad (2.1) \\ B(z) &= 1 - 0.6454z^{-1} + 0.005554z^{-2} + 0.1113z^{-3} + 0.1317z^{-4} + 0.1032z^{-5} + 0.2802z^{-6} \\ &\quad - 0.6652z^{-7} + 0.6688z^{-8}, \qquad (2.2) \\ C(z) &= (1 - z^{-1})^{-0.3}. \qquad (2.3) \end{aligned}$$

Once we obtained the model, we described how to use it to build a synthetic revocation generator that can be used in simulations of resource assessment. To be able to construct the revocation trace generator, we needed to concatenate a zero-memory non-linear function (ZMNL) to the ARFIMA model Figure 2.5 shows the block diagram of the synthetic revocation generator, where the ZMNL function is placed at the output of the ARFIMA filter. The values of the white noise sequence $w(n)$ at the input of the ARFIMA filter are chosen such that $Var(w(n)) = 1$ and $E[w(n)] = 0$. In turn, the output of the ARFIMA filter $s(n)$ becomes the input

of the ZMNL function. In this way, the ARFIMA model transforms the $N(0,1)$ sequence in a colored $N(0,\sigma_s^2)$ sequence. Then, the ZMNL function transforms the colored $N(0,\sigma_s^2)$ sequence in an $Exponential(\mu_r)$ sequence, where μ_r is the measured average of daily revoked certificates.

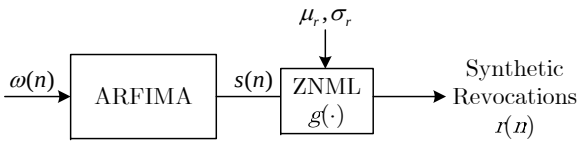


Figure 2.5: Synthetic Revocation trace generator.

Hence our model produces synthetic revocation traces that are indistinguishable for practical purposes from those corresponding to actual revocations.

2.2 PKI deployment in vehicular adhoc networks

Our previous results showed that when deploying revocation mechanisms the characteristics of the revocation process have to be taken into account. In the case of the vehicular networks, the IEEE 1609.2 standard [8] states that they will rely on the use of CRLs. In particular, OBUs must obtain the Certificate Status Information (CSI) from the revocation system. In the literature there are several mechanisms to distribute CSI in environments prone to disruptions [10, 17, 18, 19] though none of them takes into account the revocation process characteristics in its design. They are essentially based on retrieving the CSI from the infrastructure during connectivity intervals and using some caching strategy when the connection to the infrastructure is not possible. Then, OBUs may use their cached copy of the CSI (previously downloaded during a connectivity interval) or may try to discover more recent CSI among their neighbors. Once a copy of the CSI is obtained, OBUs have to face the problem of evaluating the freshness of this copy. Depending on the freshness of the CSI, the risk of trusting this information as comprehensive will vary. OBUs should be able to quantify

this risk to make an informed decision whether to operate or not with a specific certificate.

CRLs are expected to be quite large because the network scale of VANETs is expected to be very large and because to protect the privacy of users each vehicle is going to have many temporary certificates (or pseudonyms). Hence, the distribution of CRLs is prone to long delays. Moreover, during the early deployment of VANETs, RSUs may not be uniformly distributed in the network. Therefore, the way of distributing CRLs must be designed to ensure that revocation is can be correctly deployed in those delay-tolerant environments. There have been proposed several ways to improve the distribution of CRLs (e.g. [7, 19]). These proposals intended to make more efficient the distribution of the CRLs, by for example, reducing its size or using V2V communications. However, none of these proposals deals with the problem of the lack of information about certificates that are revoked during the validity interval of a CRL. In this thesis, we presented a metric that quantifies the risk that the recipients are facing when accepting messages signed with certificates that are not present in the CRLs at the OBU.

Using group theory and a probabilistic analysis, we calculated the probability of considering a certificate as a valid one when the real status known by the CA is revoked at time t as (see details in [20]):

$$\rho(t) = Prob(Cert \in \mathcal{U}) = \frac{p(t - t_0)}{(1 - p)T_c + p(t - t_0)}, \tag{2.4}$$

where T_c is the mean certificate lifetime, p is the percentage of revoked certificates and $\mathcal{U}$ is the set of revoked certificates that were not included in the previous CRL.

Figure 2.6 shows the theoretical evolution of $\rho(t)$ during three consecutive CRL updates. As expected, the probability is zero at instants of CRL update as there are no unknown revoked certificates. On the contrary, this probability is maximum just before publicizing a new CRL, as the number of unknown revoked certificates is maximum at this point. Note that this maximum (as well as the slope of the probability function) varies depending of the percentage of revoked

certificates (p_i). Thus, when this percentage is higher (note that $p_2 > p_3 > p_1$) the probability increases more rapidly.

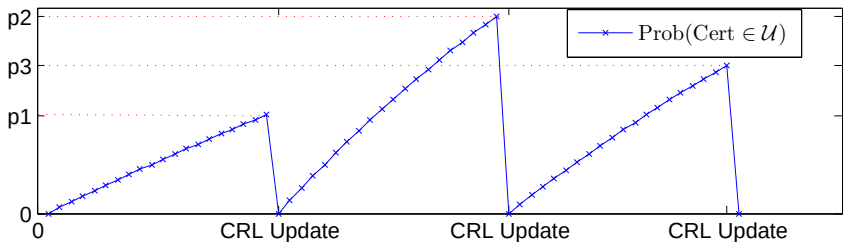


Figure 2.6: Time evolution of the probability of considering an unknown revoked certificate as valid.

Once we obtained the probability of operating with an unknown revoked certificates, we developed a new risk analysis method to identify and assess this risk. In addition, we must assure that risk information produced is processed and reliably applied to decision making. Previous works in the literature [21, 22, 23] acknowledged the existence of an operational risk when using a revocation mechanism such as CRLs. However, these works neither quantified this risk nor provided a means to deal with it. We modeled and characterized a risk-based decision making system based on fuzzy logic. To that end, taking into account the information that users can obtain from the CRLs, we design a fuzzy inference system that gives as output the risk of operating with a particular CRL. Using the proposed model, users get an idea of how risky is to operate with their current CRL and are able to make risk-based decisions.

Risk analysis by the trusting user in its potential interaction with a probable illegitimate user was done by:

1. Determining the possibility of operating with users that have their certificate revoked using $\rho(t)$;

2. Determining the possible consequences of operating with an illegitimate user, using the certificate revocation causes (see Table 2.2).

Then, we defined a fuzzy-risk based decision making system where the inputs of the inference system were:

Numerical Code	Revocation Code	w_i	Description
(1)	`keyCompromise`	9	Private key has been compromised.
(2)	`cACompromise`	10	Certificate authority has been compromised.
(3)	`affiliationChanged`	1	Subject's name or other information has changed.
(4)	`superseded`	0	Certificate has been superseded.
(5)	`cessationOfOperation`	1	Certificate is no longer needed.
(6)	`certificateHold`	3	Certificate has been put on hold.
(7)	`removeFromCRL`	0	Certificate was previously on hold and should be removed from the CRL.
(8)	`privilegeWithdrawn`	5	Privileges granted to the subject of the certificate have been withdrawn.
(9)	`aACompromise`	10	Attribute authority has been compromised.

Table 2.2: Revocation codes, weight values w_i and description.

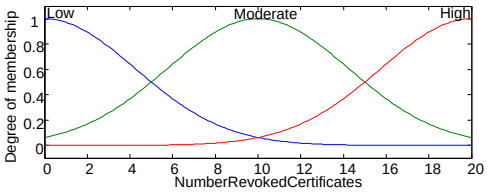
1. **Number of revoked certificates** ($NumRev$): as users have cached CRLs which include the list of revoked certificates and their revoked date, users can know the number of revoked certificates per day;

2. **Revocation categories** ($RevCat$): CRLs can also include the revocation cause of each certificate;

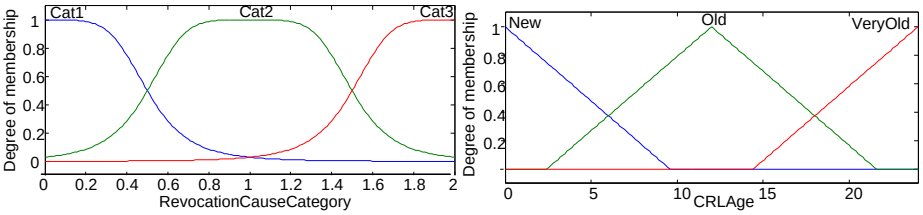
3. **Age of the CRL** (CRL_{age}): using also the information contained in the CRL; users can calculate the time elapsed since the issuance of the CRL.

For each of these inputs we defined a *membership* function (see Fig. 2.7).

Finally, a case study on risk analysis of a CRL issued by an actual CA was used to show the validity of the proposed model. The results of the risk assessment in the case study were represented as risk score, located in a defined range, and risk category with linguistic words, which indicates that by using the proposed methodology the risk associated with CRLs can be assessed effectively

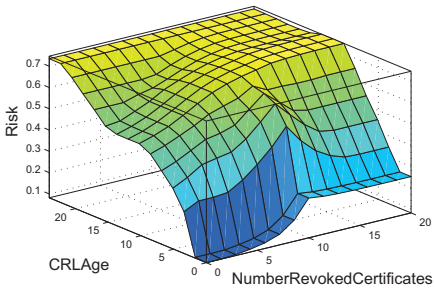


(a) Number of Revoked Certificates

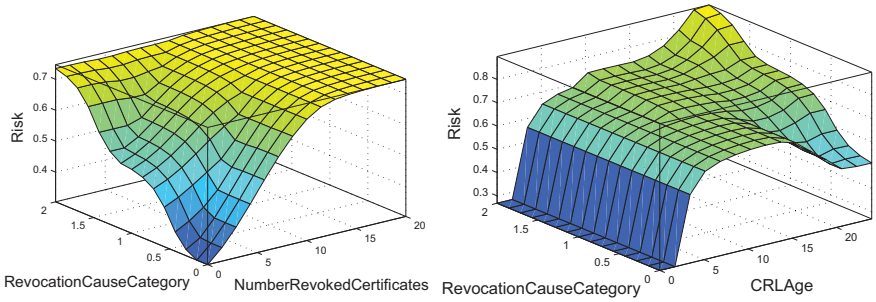


(b) Revocation Cause Categories (c) CRL Age

Figure 2.7: Membership functions.



(a)



(b) (c)

Figure 2.8: Risk Indicator as a combination of a) the CRL age and the number of revoked certificates, b) the revocation cause categories and the number of revoked certificates, c) the CRL age and the revocation cause categories.

and efficiently. These results showed that although this CA is issuing CRLs with a frequency of 1 day, there is still some inherent risk that our model achieves to measure (see Fig. 2.8). Based on this metric, any user that operates using certificates from this CA could make informed decisions.

2.3 Certificate Status Checking mechanism for VANETs

At this point of the thesis, we were aware that current revocation mechanisms will exhibit some issues when directly applied to a vehicular network, and they need to be improved in terms of efficiency and CSI freshness. First of all, we analyzed the drawbacks of applying the IEEE 1609.2 standard proposal which suggests the use of CRLs.

As mentioned before, for a CA to invalidate a vehicle's certificates, the CA includes the certificate serial number in the CRL. The CA then distributes the CRL so that vehicles can identify and distrust the newly revoked vehicle. The distribution should spread quickly to every vehicle in the system. However, the distribution itself poses a great challenge due to the size of the CRL. As a CRL is a list containing the serial numbers of all certificates issued by a given certification authority that have been revoked and have not yet expired, its distribution causes network overhead. Moreover, the CRL size increases dramatically even if only a small portion of the OBUs in the VANET is revoked. To have an idea of how big the CRL size can be, consider the case where 1% of the total number of the OBUs in the United States is revoked. Recall that in a VANET, each vehicle owes not only an identity certificate, but also several pseudonyms. The number of pseudonyms may vary depending on the degree of privacy and anonymity that it must be guaranteed. According to [24], OBUs must store enough pseudonyms to change pseudonyms about every minute while driving. This equates to about 43,800 pseudonyms per year for an average of two hours of driving per day. In the US, 255,917,664 "highway" registered vehicles were counted in 2008, of which 137,079,843 passenger cars [25]. In this case, the CRL would contain around 100 billion revoked certificates. Assuming that certificates can be identified by a 16

byte fingerprint (the size of one AES block), the CRL size is around 1,7 TB. Only the amount of memory necessary to storage this CRL makes it impossible its deployment.

The CRL size can be reduced by using regional CAs. However, it appears a trade-off between the region and the CRLs size. While using a single CA responsible for every certificate and pseudonym will give place to CRLs of several terabytes. Hence, it is necessary to divide the CRL information according to regional areas. In this sense, if we divide the entire United States by cities (i.e. ∼10,016 cities), the CRL size is reduced to around 170 Mbytes. Using the 802.11a protocol to communicate with RSUs in range, vehicles could have between 10-30 Mbps depending on the vehicle's speed and the road congestion [26]. Thus, in the best case (under non-congested conditions) a vehicle will need more than 45 seconds to download the whole CRL. In scenarios where vehicles are not able to keep a permanent link with the infrastructure for this amount of time, techniques such as Bloom filter or Digital Fountain Codes could be used to download the CRL. Therefore, though the problem of having a huge CRL is mitigated by the use of such techniques, the restraints imposed by the distribution affect the freshness of the revocation data.

A direct consequence of this significant time to download a CRL is that a new CRL cannot be issued very often, so its validity period has to be shortened. This validity period directly determines how often a vehicle has to update the revocation data. Therefore, the validity period of the CRL is critical to the bandwidth consumption. Moreover, it appears another trade-off between the freshness of revocation data and the bandwidth consumed by downloading CRLs. Large validity periods will decrease the network overhead at expenses of having outdated revocation data. Small validity periods will increase the network overhead but users will have fresh information about revoked certificates. As CRLs cannot be issued every time there is a new revoked certificate, vehicles will be operating with revocation data that are not comprehensive. In this thesis we developed a revocation mechanism to improve the performance of the revocation process in a vehicular network by taking advantage of authenticated data structures and V2V communications.

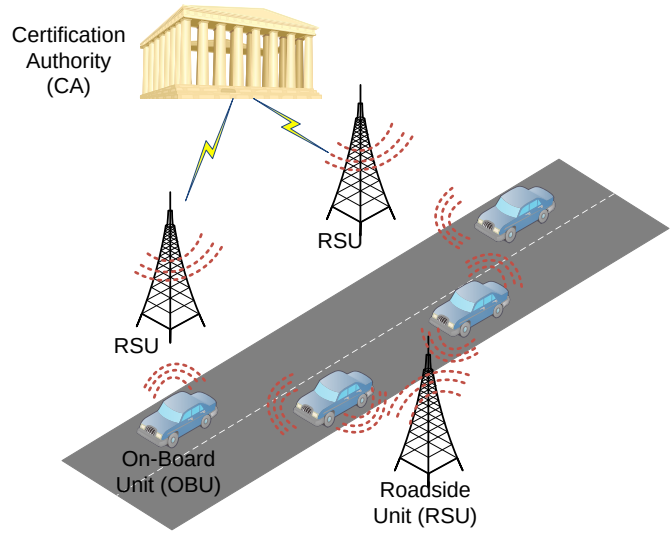


Figure 2.9: System Architecture.

Our proposal was called COACH (COllaborative certificate stAtus CHecking). COACH is an application-layer mechanism for distributing revocation data. The system architecture is an adaptation of a PKI system to the vehicular environment (see Fig. 2.9). The main idea behind COACH is to embed some little extra information into the CRL such that allows us to create an efficient and secure request/response protocol. For those nodes that just need to obtain status data of some certificates, our protocol avoids downloading a complete CRL. Specifically, we proposed a way of efficiently embedding a Merkle hash tree (MHT) [27] within the structure of the standard CRL to generate a so-called *extended-CRL*.

To create the *extended-CRL*, we used an extension, which is a standard way of adding extra information to the CRL. Our extension contains all the necessary information to allow any vehicle or VANET infrastructure element that possesses the *extended-CRL* to build the COACH tree, i.e., a hash tree with the CSI of the CRL. Using this COACH tree, any entity possessing the *extended-CRL* can act as repository and efficiently answer to certificate status checking requests of other vehicles or VANET elements (see Fig. 2.10). COACH responses are short since in general, their size is less than 1 Kbyte (see Table 2.3, where T_{hash} and T_{mul} denote the time required to perform a pairing operation and a point multiplication, respectively). This allows a COACH response to perfectly fit

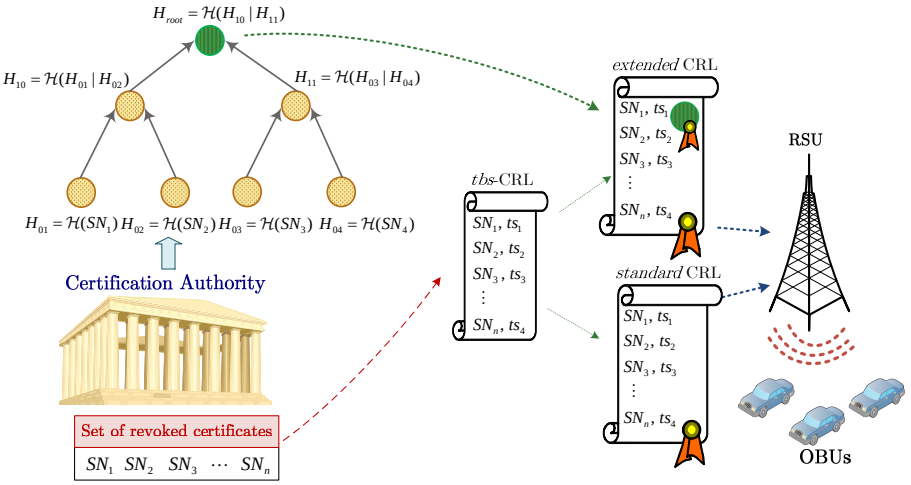


Figure 2.10: COACH bootstrapping.

within a single UDP message. We also proposed an enhancement of our basic mechanism called EvCOACH (Evergreen-COACH) to improve the performance of COACH in scenarios with relatively few revocations per CRL validity period.

Mechanism	Request size	Response size	Verification delay	Signing delay
CRL	73 bytes	145 Mbytes	$4T_{mul}$	T_{mul}
COACH	73 bytes	710 bytes	$k(T_{hash}(log_2 N + 1) + 4T_{mul})$	T_{mul}
EvCOACH	73 bytes	725 bytes	$k(T_{hash}(log_2 N + i + 2) + 4T_{mul})$	T_{mul}
ADOPT	66 bytes	586 bytes	$k(4T_{mul})$	$k(T_{mul})$

Table 2.3: COACH vs other certificate validation mechanisms

Note that ADOPT [10] (Ad-hoc Distributed OCSP for Trust) provides a revocation service based on the Online Status Checking Protocol (OCSP)[28] in a decentralized manner. ADOPT uses cached OCSP responses that are distributed and stored on intermediate nodes in the VANET. Thus, ADOPT's query cost is the lowest but not far from COACH. Moreover, we also showed by simulation, that COACH makes the distribution of CSI more efficient than distributing complete CRLs (even though they are compressed), reducing the data that have to be transmitted over the VANET.

Vehicle Speed	*Delay*								
	COACH			**CRL**			**ADOPT**		
	Tx	*Comp.*	*RTT*	*Tx*	*Comp.*	*RTT*	*Tx*	*Comp.*	*RTT*
20 m/s	75 ms	2,401 ms	78 ms	3,521 h	2,400 ms	3,521 h	72 ms	3,612 ms	101 ms
30 m/s	149 ms	2,401 ms	157 ms	8,213 h	2,400 ms	8,214 h	122 ms	3,600 ms	312ms
40 m/s	173 ms	2,401 ms	187 ms	9,811 h	2,400 ms	9,813 h	152 ms	3,600 ms	421ms

Table 2.4: Delays when querying for CSI.

Table 2.4 shows the mean delays incurred when querying for the status of a given certificate. With *transmission delay* we denote the time to send the CSI query and the corresponding response. If we compare the transmission delay of the different revocation mechanisms, we can observe that ADOPT is the fastest but not so far from COACH. On the other hand, by *computational delay* we denote the time required to compose and validate a CSI response. In this case, ADOPT has the worst computational delay because each CSI response has to be signed by the CA. CRL computational delay is minimal as the CRL is only signed once and to searching the serial number of a certificate in the list has a computational cost of $O(log_2 N)$. COACH only requires one CA signature but a $\mathcal{P}$ath has to be computed each time a CSI response is required, so the computational cost is similar to the CRL. Finally, we define Round-Trip Time (RTT) as the time that takes since a vehicle requests for CSI until the status of a given certificate is validated. Therefore, the RTT is affected by the transmission, computational and propagation delays. ADOPT has the worst RTT due to the multihop transmission of the cached CSI, while CRL and COACH download the CSI directly from the repository in range. In any case, the vehicles' speed affects transmission and RTT delays in all three revocation mechanisms. We must stress that a node possessing an *extended-CRL* can act as COACH repository but that a COACH repository is not a TTP. In other words, COACH is cryptographically offline, which means that no online trusted entity (like a CA) is needed for authenticating the responses produced by COACH repositories.

Though COACH improves the efficiency of the revocation mechanism, it does not enhance the freshness of the revocation data. To that end, we designed a new

mechanism to improve the distribution of CSI by transmitting the revocation information that is unknown to a particular user during the validity period of the cached CSI. The main idea behind that is to allow vehicles requesting for new CSI during the validity period of the current CRL. Thus, revocations that occur during the validity period of the CRL will not be unknown to the vehicles during this whole validity period, reducing the risk of operating with an unknown revoked certificate. We addressed the CRL distribution problem by exploiting the combination of three well-known mechanisms: (1) delta-CRL [29], (2) Merkle hash tree (MHT) [27], and (3) one-way hash chain [30]. By combining these three mechanisms, we designed a new tool which allows increasing the availability and freshness of the certificate status information and at the same time reduces the bandwidth necessary to check the validity of a given certificate. It takes advantage of V2V communication to create mobile repositories so that vehicles do not have to rely solely in the RSUs to obtain CSI. Therefore, it reduces the peak bandwidth load associated with the CSI requests as there are more entities in the network that can answer these requests. To achieve, we combine the issuance of delta-CRLs and MHT. (*For more details see [31] or the full version of this document available at Biblioteca Rector Gabriel Ferraté*).

By using the underlying concept of delta-CRLs, we implemented a more efficient way of distributing CSI inside the VANET. To help minimize frequent downloads of lengthy CRLs, delta-CRLs are published aperiodically. On the other hand, our mechanism codes the information included in the CRL and delta-CRLs in different MHTs. As in COACH using this tree, any entity possessing the *extended-CRL* can act as repository and efficiently answer to certificate status checking requests of other vehicles.

Using the NCTUns [32] simulator we analyzed the performance in a vehicular scenario. In VANETs, the most important issue in any revocation method is the delay of delivering the CSI to the vehicles to prevent that misbehaving vehicles from jeopardizing the safety of its neighbors. Consequently, we measured the revocation delay as delay from the moment a vehicle issues a CSI request until the moment the new CSI is received. (*For more details see [31] or the full version of this document available at Biblioteca Rector Gabriel Ferraté*).

It is worth noting that the worst mechanisms in terms of delay are the traditional CRL and delta-CRL as requesting entities are downloading all the available CSI. However, the delay of the conventional CRL compared with the proposed protocol decreases with the number of CSI requests. The variations in time to download the CRL are due to the number of intermediate RSUs existing in the connection between the CA and the vehicle sending the revocation request. The average time to validate the status of a certificate in ADOPT is lower because of the number of hops that are necessary to retrieve the cached CSI. Our proposal in its MHT mode of operation is the fastest in average when validating the status of a certificate. However, this mode of operation has also a notable deviation. While in ADOPT the high deviation is due to the number of hops, in our proposal this deviation is mainly due to the number of Δ-trees that a vehicle has to check when a certificate is not revoked.

Thus, the evaluation shows that it not only improves in terms of bandwidth but also in terms of scalability (increases the number of available repositories) and vulnerability (controlled WOV). In this way, our proposal becomes an offline certificate status validation mechanism as it does not need trusted responders to operate. Therefore, our proposal significantly achieves great efficiency and scalability, especially when deployed in heterogeneous vehicular networks.

2.4 Impact of the revocation service in PKI prices

As final part of this thesis, we analyzed the economic impact of the revocation service in the certificate prices. We noticed that with the appearance of novel network environments (e.g. VANET), the quantity of CAs in the SSL certificate market will become larger and the market concentration will diminish, but this will not simple eliminate the oligopoly in the short-term. During the 90s, the certification market, the competition among CAs appears mainly as price competition. In this situation, malignant price competition would be detrimental to the interests of the users and lead to the CA's pay crisis. Facing the situation, the main CAs began to change the competitive strategies from basic price competition to price and quality of services (QoS) competition. To provide better QoS, CAs have to improve their revocation service, and specifically the freshness of the

CRLs. Users will pay more for a service that issues certificate status information faster. Time-to-revocation metric is visible to costumers by checking the CA's repositories where they publicize the revocation information.

We proved that there exists an oligopoly of CAs which compete in certificate prices and QoS. We assume that the revocation probability is *ex-ante* uncertain which is quite logical and intuitive. The number of revoked certificates varies with time and in a manner that cannot be predicted with certainty. We showed that an uncertain revocation probability introduces a systematic risk that does not decrease by selling more certificates. If CAs are risk averse, this effect relaxes price competition. The equilibrium characteristic of the certification market was found by establishing a price competition model with different QoS.

Firstly, we defined a utility function. We maximized this utility, assuming that the total utility U which users can get after they purchase a certificate consists of two parts. The first part is wealth utility which represented by U_w the other part is QoS utility which the applicant can get after they obtained the CA's services, represented by U_{QoS}. The total utility U is defined as:

$$U = \alpha_1 U_w + \alpha_2 U_{QoS}, \forall \alpha_k \in [0,1] \text{ and } \alpha_1 + \alpha_2 = 1. \tag{2.5}$$

where α_i represents the significance level of U respectively. We assume that the certification market is covered in full. Users will intend to maximize their utility, i.e.:

$$\theta^* = \arg\max_{\theta} \quad U. \tag{2.6}$$

We obtained the certificate price and the coverage in the equilibrium, which allowed us to conclude that:

- In the equilibrium, two CAs with different revocation services achieve their maximum gain, the CA with better revocation service obtains a higher price for their certificates. This is mainly due to the fact that as both CAs have associated the same probability of being compromised, but the QoS of the first CA is better, this CA can set a higher price per certificate.

- In the equilibrium, the coverage that each CA should establish is the same and is inversely proportional to the risk-aversion and the probability of operating with a revoked certificate.

Finally, to corroborate the benefits of the presented model, we analyzed the case of current SSL providers that issue digital certificates. An SSL certificate can be obtained from amounts as low as $43 to as high as $3000 per year. Whilst the type of encryption can be the same, the cost is determined by the rigor of the certification process as well as the assurance and warranty that the vendor can provide. Table 2.5 shows the prices and QoS that the leading CAs operating in the SSL Certificate market are offering. The SSL Certificate market was traditionally dominated by a small number of players, namely VeriSign and Thawte. Whilst in a monopolistic position they had the capability of charging inflated prices for a commodity product. However new providers with no necessity to hold prices high were able to offer SSL certificates at far more reasonable prices.

SSL Provider	**Product Name**	**Price/Year($)**	**Warranty($)**	**Assurance**	**Issuing time**	**Mean CRL lifetime**
COMODO	EnterpriseSSL Platinum	311.80	1,000,000	High	<1 hour	4 days
COMODO	InstantSSL Pro	169.80	100,000	High	<1 hour	4 days
Verisign	Secure Site Pro Cert	826.67	2,500,000	High	2-3 days	15 days
Verisign	Managed PKI for SSL Std	234.00	100,000	High	2-3 days	15 days
GeoTrust	QuickSSL Premium	118.00	100,000	Low	Immediate	10 days
GeoTrust	True BusinessID	159.20	100,000	High	2 days	10 days
Go Daddy	Standard SSL	42.99	10,000	Low	Immediate	1 day
Go Daddy	Standard Wildcard	179.99	10,000	Low	Immediate	1 day
Entrust	Advantage SSL Certificates	167.00	10,000	High	2 days	1 week
Entrust	Standard SSL Certificates	132.00	10,000	High	2 days	1 week
Thawte	SSL 123	129.80	-	Low	Immediate	1 month
Thawte	SGC Super cert	599.80	-	High	2 days	1 month

Table 2.5: SSL Certificate Types and Services offered by main CAs [1].

To test whether these factors are determinant factors for the certificate prices, we perform a multivariate regression analysis explaining the yearly price of SSL certificates. General regression investigates and models the relationship between

a response (Certificate price) and predictors (Warranty, issuing interval and CRL lifetime). Note that the response of this model is continuous, but we have both continuous and categorical predictors. With this model we determine how the certificate price changes as a particular predictor variable changes. We use data from a survey of CAs performed in 2010 [1]. The obtained regression model is expressed in the following equations for high and low assurance certificates, respectively:

$$Price/Year(\$) = 98,4353 + 0,000220857\,W - 0,549141\,\overline{I_{time}} + 8,6116\,\frac{1}{\overline{CRL_{Lf}}},$$

$$Price/Year(\$) = 20,0405 + 0,000220857\,W - 0,5491411\,\overline{I_{time}} + 8,6116\,\frac{1}{\overline{CRL_{Lf}}},$$

where W denotes the warranty, $\overline{I_{time}}$ is the mean issuing time, and $\overline{CRL_{Lf}}$ is the mean lifetime of the CRLs issued by the CA.

Both regression equations show that the coefficient of the predictor associated to the CRL's mean lifetime is significant. Thus, it is demonstrated that the revocation service plays an important role when establishing the certificate prices.

Chapter 3

Quality Indexes

The research presented in this thesis has been validated internationally in various journals and conferences where experts have provided valuable comments and insights that have improved our research. Tables 3.1 and and 3.2 show the publications made during the development of the thesis In both tables, the displayed information gives evidence of the quality of each of them.

Year	Publication Title	Journal	Quality Index
2013	Efficient Certificate Status Information distribution mechanism	Mobile Information Systems	Impact Factor (ISI) = 2.432 h-index= 23
2012	Risk based decision making for Public Key Infrastructure using fuzzy logic	International Journal of Innovative Computing, Information and Control	Impact Factor (ISI) = 1.667 h-index= 32
2012	A Modeling of Certificate Revocation and Its Application to Synthesis of Revocation Traces	IEEE Transactions on Information Forensics and Security	Impact Factor (ISI) = 1.340 h-index= 41
2012	COACH: COllaborative certificate stAtus CHecking mechanism for VANETs	Journal of Network and Computer Applications	Impact Factor (ISI) = 1.065 h-index= 28

Table 3.1: Quality Indexes of the articles published in journals.

Year	Title	Conference	Quality Index
2012	Impact of the Revocation Service in PKI Prices	Information and Communications Security (ICICS 2012)	CORE Ranking B h-index= 20
2012	On the Self-similarity Nature of the Revocation Data	in Information Security Conference (ISC 2012)	CORE Ranking B h-index= 16
2012	Toward Revocation Data Handling Efficiency in VANETs	in Communication Technologies for Vehicles (Nets4Cars 2012)	No CORE ranking h-index= 2
2009	PKIX certificate status in hybrid MANETs	Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks (WISTP 2009)	CORE Ranking C h-index= 10

Table 3.2: Quality Indexes of the articles presented at international conferences.

Chapter 4

Conclusions

In this thesis, we have analyzed the revocation process and proposed a set of mechanisms to provide an efficient revocation service for VANETs. Our results have shown that the proposed mechanisms can achieve the targeted security requirements. In addition, the detailed performance evaluation and security analysis have indicated that the proposed protocols are secure and efficient. The achievements accomplished in this thesis can be summarized as follows:

- We have analyzed real empirical data collected from the leading CAs. We have shown that the revocation process is statistically self-similar (irrespective of when data were collected during the 3-year period 2008-2011 or from which CA). Moreover, we have demonstrated that the degree of self-similarity, which can be measured in terms of the Hurst parameter H, is a function of the overall utilization of the revocation service and can be used for measuring the "burstiness" of the revocation process (i.e. the more bursts in the revocation process the higher H). Hence, leading CAs share similar Hurst parameters even though they operate in different market segments.

- The intermittent connectivity between the entities of vehicular networks and security infrastructure results in incomplete or outdated revocation information at the recipients of signed messages. This incomplete/outdated information puts the recipients in a dilemma while accepting messages signed using certificates that are not present in the CRLs at the On Board Unit. To

this respect, we have presented a new metric that quantifies the confidence the recipients can have while accepting messages signed using certificates that are not present in the CRLs at the OBU. Moreover, we have developed a systematic methodology to build a fuzzy system that models risk and assists the user in the decision making process related to certificate revocation. Our system not only considers the possibility of taking as valid a certificate that has been revoked but also other key risk factors. In this respect, we have identified potential risk sources involved in the revocation system and we have characterized them using fuzzy logic. The inputs given to the fuzzy system can be inferred from a standard CRL and as CRLs are accessible to any PKI user, in practice, everybody can take advantage of our fuzzy system. The output of our system is a measure of the risk of operating with a particular CRL at a given instant. Based on this output, users can either decide whether to trust or not a given signed message.

- We have proposed two efficient revocation mechanism for VANETs, which substantially reduce the overhead of the certificate status checking. These checking mechanisms are based on an *extended-CRL*. The main advantage of this *extended-CRL* is that the road-side units and repository vehicles can build an efficient structure based on an authenticated hash tree to respond to status checking requests inside the VANET, saving time and bandwidth. Thus, we decrease the vulnerability window that a misbehaving vehicle has and this results in higher safety level for VANET. Both mechanisms are resistant to the most known revocation attacks. In addition, they can be efficiently integrated with any PKI and/or any misbehavior detection scheme for VANETs and they fulfill the IEEE 1609.2 Standard.

- Finally, we have studied the economic impact of the revocation service in the certificate price. We have shown that the market of certificate providers can be described as an oligopoly where oligarchs compete not only in price but also in quality of service. We have modeled this oligopoly using a game theoretic approach to find the prices in the equilibrium. We have been able to capture the QoS of the products offered by a CA, by means of the timeliness of the revocation mechanism and the security level. In

our model of the certification industry with profit-maximizing CAs and a continuum of individuals, we showed that although the undercutting process in certification prices seems similar to the price setting behavior of firms in Bertrand competition there exists a crucial difference depending on the QoS of the revocation service. The solution of the game for two CAs in the oligopoly that offer certificates with different QoS shows that the revenues of the CA which provides a better revocation mechanism and a higher security level are larger. Therefore, a CA has to take into account not only the probability of operating with a revoked certificate, but also the quality of the revocation mechanism and the security level when setting the prices of its certificates and the compensation expenses. Thus, any CA should comprehensively consider the difference in quality of its services compared with other CAs.

Appendix A

Publications

[1] C. Gañán, J. L. Muñoz, O. Esparza, J. Mata-Díaz, J. Hernández-Serrano, and J. Alins, "COACH: COllaborative certificate stAtus CHecking mechanism for VANETs," *Journal of Network and Computer Applications*, Mar. 2012. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1084804512000616

[2] C. Gañan, J. L. Muñoz, O. Esparza, J. Loo, J. Mata-Díaz, and J. Alins, "Efficient Certificate Status Information distribution mechanism," *Mobile Information Systems*, pp. 1–31, 2013, (in press). [Online]. Available: http://dx.doi.org/10.3233/MIS-130167

[3] C. Gañán, J. Mata-Diaz, J. L. Munoz, J. Hernandez-Serrano, O. Esparza, and J. Alins, "A Modeling of Certificate Revocation and Its Application to Synthesis of Revocation Traces," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1673–1686, Dec. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6247505

[4] C. Gañán, J. Muñoz, O. Esparza, J. Mata-Díaz, and J. Alins, "Risk-based decision making for Public Key Infrastructure using fuzzy logic," *International Journal of Innovative Computing, Information and Control (IJICIC)*, vol. 8, no. 11, pp. 7925–7942, 2012. [Online]. Available: http://www.ijicic.org/ijicic-ksi-07.pdf

[5] C. Gañán, J. L. Muñoz, O. Esparza, J. Mata-Díaz, and J. Alins, "Impact of the Revocation Service in PKI Prices," in *Information and Communications Security*, ser. Lecture Notes in Computer Science, T. Chim and T. Yuen, Eds., vol. 7618. Hong Kong: Springer Berlin Heidelberg, 2012, pp. 22–32. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-34129-8_3

[6] C. Gañán, J. Mata-Díaz, J. L. Muñoz, O. Esparza, and J. Alins, "On the Self-similarity Nature of the Revocation Data," in *Information Security*, ser. Lecture Notes in Computer Science, D. Gollmann and F. Freiling, Eds., vol. 7483. Passau: Springer Berlin Heidelberg, 2012, pp. 387–400. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33383-5_24

[7] J. Muñoz, O. Esparza, C. Gañán, and J. Parra-Arnau, "PKIX certificate status in hybrid MANETs," in *Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks*, ser. Lecture Notes in Computer Science, O. Markowitch, A. Bilas, J.-H. Hoepman, C. Mitchell, and J.-J. Quisquater, Eds. Springer Berlin Heidelberg, 2009, vol. 5746, pp. 153–166. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-03944-7_12

[8] C. Gañán, J. Muñoz, O. Esparza, J. Mata-Díaz, and J. Alins, "Toward Revocation Data Handling Efficiency in VANETs," in *Communication Technologies for Vehicles*, ser. Lecture Notes in Computer Science, A. Vinel, R. Mehmood, M. Berbineau, C. Garcia, C.-M. Huang, and N. Chilamkurti, Eds., vol. 7266. Vilnius: Springer Berlin Heidelberg, 2012, pp. 80–90. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-29667-3_7

A Modeling of Certificate Revocation and Its Application to Synthesis of Revocation Traces

Carlos Gañán, Jorge Mata-Díaz, Jose L. Muñoz, Juan Hernández-Serrano, Oscar Esparza, and Juanjo Alins

***Abstract*—One of the hardest tasks of a public key infrastructure (PKI) is to manage revocation. New communication paradigms push the revocation system to the limit and an accurate resource assessment is necessary before implementing a particular revocation distribution system. In this context, a precise modeling of certificate revocation is necessary. In this paper, we analyze empirical data from real certification authorities (CAs) to develop an accurate and rigorous model for certificate revocation. One of the key findings of our analysis is that the certificate revocation process is statistically self-similar. The proposed model is based on an autoregressive fractionally integrated moving average (ARFIMA) process. Then, using this model, we show how to build a synthetic revocation generator that can be used in simulations for resource assessment. Finally, we also show that our model produces synthetic revocation traces that are indistinguishable for practical purposes from those corresponding to actual revocations.**

***Index Terms*—Autoregressive fractionally integrated moving average (ARFIMA), public key infrastructure (PKI), revocation, self-similarity.**

I. Introduction

DIGITAL certificates are means of accurately and reliably distributing public keys to users needing to encrypt messages or verify digital signatures. Certificates are signed by certification authorities (CAs) and managed during their life-cycle by a Public Key Infrastructure (PKI). Various circumstances may cause a certificate to become invalid prior to the expiration of its planned validity period. Thus, the PKI has to collect and distribute information about revoked certificates. Currently deployed PKIs rely mostly on Certificate Revocation Lists (CRLs) for handling certificate revocation [1]. Although CRLs are the most widely used way of distributing certificate status information, much research effort has been put on studying other revocation distribution mechanisms in a variety of scenarios [2], [3].

In the past, little work has been done for analyzing the revocation process itself. For instance, many studies in the literature compare the performance of the different revocation distribution mechanisms considering very simplistic assumptions about the revocation process such that the percentage of revoked certificates remains always constant in the system. While these assumptions might have been enough in the early deployment of PKIs, we strongly believe that they have to be refined to face the challenges of currently deployed PKIs and also of emerging scenarios such as VANETs or WSNs. Only recently, we can find works like [40], [4]–[6] that carry out statistical studies about the revocation process using data available from real CAs. These studies can be considered a first step towards understanding revocation. They essentially analyze the probability distribution of revocation and conclude that this distribution roughly follows a Poisson distribution.

Following this direction, we also analyze empirical data from real CAs and we go a step further by developing an accurate and rigorous model for certificate revocation. One of the key findings of our analysis is that the certificate revocation process is statistically self-similar. As none of the currently common formal models for revocation is able to capture the self-similar nature of real revocation data, we develop a method for modeling this behavior. The proposed model is based on an autoregressive fractionally integrated moving average (ARFIMA) process [7], which provides an accurate and parsimonious model for revocation. Once we obtain the model, we show how to use it to build a synthetic revocation generator that can be used in simulations of resource assessment. To be able to construct the revocation trace generator, we will show that we need to concatenate a zero-memory nonlinear function (ZMNL) to the ARFIMA model. The final result is that our model produces synthetic revocation traces that are indistinguishable for practical purposes from those corresponding to actual revocations.

With synthetic revocation traces, current revocation schemes can be improved to define more accurate revocation data issuance policies. Neglecting the burstiness of the revocation process leads to inefficient revocation data release strategies. We show that traditional mechanisms that aim to scale, such as delta-CRL, can benefit from our traces to improve their updating strategies.

The rest of this paper is organized as follows. In Section II, we introduce the two stochastic processes that we use to model the revocation process. In Section III, we discuss the methodology we used to collect and analyze real-world revocation data. In Section IV, we identify the best ARFIMA model that fits the revocation events. Next, in Section V we present the generator of synthetic revocation traces using the obtained ARFIMA model. In Section VI we discuss the applications of this generator and in Section VII we present the impact of our finding on the related work. Finally, we conclude in Section VIII.

Manuscript received November 04, 2011; revised June 30, 2012; accepted July 04, 2012. Date of publication July 23, 2012; date of current version November 15, 2012. This work was supported by the Spanish Ministry of Science and Education under the projects CONSOLIDER-ARES (CSD2007-00004) and TEC2011-26452 "SERVET," and by the Government of Catalonia under Grant 2009 SGR 1362. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Robert H. Deng.

The authors are with the Telematics Department, Universitat Politècnica de Catalunya, Barcelona, 08034, Spain (e-mail: carlos.ganan@entel.upc.edu; jmata@entel.upc.edu; jose.munoz@entel.upc.edu; jserrano@entel.upc.edu; oesparza@entel.upc.edu; juanjo@entel.upc.edu).

Digital Object Identifier 10.1109/TIFS.2012.2209875

II. Background

A. Self-Similar Processes

A phenomenon which is self-similar looks the same or behaves the same when viewed at different degree of magnification. Self-similarity is the property of a series of data points to retain a pattern or appearance regardless of the level of granularity used and can be the result of long-range dependence (LRD) in the data. If a self-similar process is bursty at a wide range of timescales, it may often exhibit LRD.

LRD means that all the values at any time are correlated in a positive and non-negligible way with values at all future instants. A continuous time process $Y = Y(t),\ t \geq 0$ is self-similar if it satisfies the following condition [8]:

$$Y(t) \stackrel{D}{=} a^{-H} Y(t), \quad a > 0 \ \ t \geq 0, \quad for\ 0 < H < 1,$$

where $\stackrel{D}{=}$ means equally distributed and H is the index of self-similarity, called the Hurst parameter and the equality is in the sense of finite-dimensional distributions. The values of H are in the interval $(0.5, 1)$ if there exists LRD. A value of H equal to 0.5 indicates the absence of LRD.

B. Autoregressive Fractionally Integrated Moving Average (ARFIMA) Processes

One observed property of many data series is that they appear to have long memory, either in mean or in variance. This means that the effect of shocks on the time series takes a very long time to disappear. Traditional models describing short-term memory, such as autoregressive, integrated, moving average processes (ARIMA) defined in [7], cannot precisely describe long-term memory. A set of models has been established to overcome this difficulty, and the most famous one is the autoregressive fractionally integrated moving average model (ARFIMA). ARFIMA model was established by [9] and [10]. These processes are the natural generalization of the standard ARIMA by permitting the degree of differencing (d) to take fractional values.

Each ARFIMA process has three parts: the autoregressive (or AR) part; the integrated (or I) part; and the moving average (or MA) part. The models are often written in shorthand as $ARFIMA(p, d, q)$ where p describes the AR part, d describes the integrated part and q describes the MA part. Unlike common ARIMA, in ARFIMA processes the degree of differencing d is allowed to take non-integer values [10].

- **Auto Regressive**. This part of the model describes how each observation is a function of the previous p observations. For example, if $p = 1$, then each observation is a function of only one previous observation. That is, $y(n) = a_0 + a_1 y(n-1) + w(n)$ where $y(n)$ represents the observed value at n, $y(n-1)$ represents the previous observed value at $n-1$, $w(n)$ represents some random error and a_0 and a_1 are both constants. Other observed values of the series can be included in the right-hand side of the equation if $p > 1$:

$$y(n) = a_0 + a_1 y(n-1) + \cdots + a_p y(n-p) + w(n). \quad (1)$$

- **Integrated**. This part of the model determines whether the observed values are modeled directly, or whether the differences between consecutive observations are modeled instead. If $d = 0$, the observations are modeled directly. If $d = 1$, the differences between consecutive observations are modeled. If $d = 2$, the differences of the differences are modeled. In practice, d is rarely more than 2. That is a non-stationary process is integrated of order d if we need to difference it d times to induce stationary and it is denoted $I(d)$. Although the integrated component can be considered within the AR component by its formulation, its synthesis depends on different factors. Thus, the integrated component also shows the dependence with past values of the series but its synthesis depends on the nonstationary moments of the process. The order d of the integrated component is fixed by the order of the highest nonstationary moment of the stochastic process. In general, the integrated component can be expressed:

$$s(n) = c_1 s(n-1) + \cdots + c_d s(n-d) + w(n), \quad (2)$$

where:

$$c_i = \frac{\Gamma(i+d)}{\Gamma(d)\Gamma(i+1)}, \quad (3)$$

where Γ represents the gamma function, and

$$d < \frac{1}{2},\ d \neq 0, -1, -2, \ldots, \text{and } i \in \{1, 2, \ldots, \infty\}.$$

- **Moving Average**: This part of the model describes how each observation is a function of the previous q errors. For example, if $q = 1$, then each observation is a function of only one previous error. In general,

$$x(n) = b_0 w(n) + b_1 w(n-1) + \cdots + b_q w(n-q), \quad (4)$$

where the terms b_i are constant coefficients. Here $w(n)$ represents the random error at n and $w(n-q)$ represents the previous random error at $n-q$.

It is worth noting that ARFIMA processes are related to self-similarity. Beran showed that partial sums of an ARFIMA process have the same limiting distribution as a globally self-similar process [11]. Thus, an ARFIMA process is a self-similar process with the ability to capture both the short-range dependent (SRD) and LRD characteristics, that is, an ARFIMA process can be regarded as the increment process for a globally self-similar process. In this sense, we can relate the two parameters that define each one of these processes. The relation between the Hurst parameter (i.e., the index of self-similarity) and d (i.e., the index of fractionality) is [12]:

$$H = d + 0.5. \quad (5)$$

III. Data Collection and Preprocessing

The previous step to design our synthetic traces generator is to acquire information from real CAs about revoked certificates,

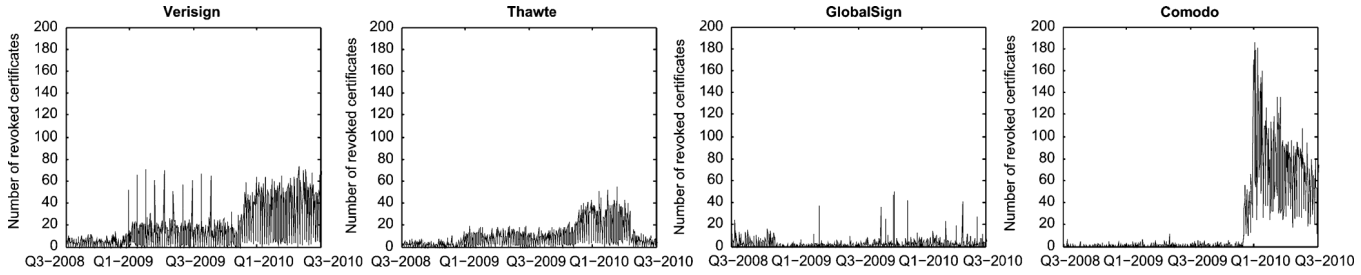


Fig. 1. Number of revoked certificates evolution for each CA.

TABLE I
DESCRIPTION OF THE COLLECTED CRLS

CRL Name	#Revoked Cert.	Issuer	Last Update	CRL Size	Cert. Type
Class3InternationalServer	16.584	VeriSign	2011/01/31	∼ 200 KB	SGC
UTN-USERFirst-Client	26.286	Comodo	2011/02/03	∼ 100 KB	SGC
PersonalSign Class 2 CA	6.695	GlobalSign	2011/02/07	∼ 300 KB	SGC
ThawteCodeSigningCA	10.418	Thawte	2011/01/31	∼ 200 KB	CSC

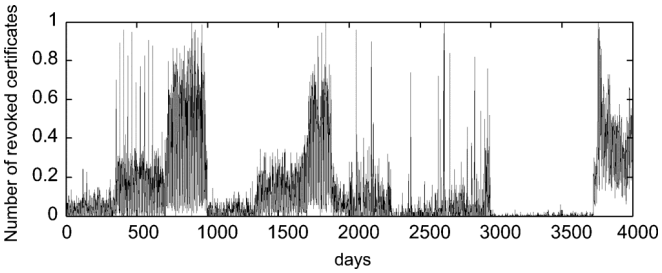


Fig. 2. Global time series.

so we can analyze the time evolution of these data. For this purpose, we collected revocation data from different certification authorities (CAs) using their available Certificate Revocation Lists (CRLs). In particular, we built some scripts to download and preprocess the CRLs from the following CAs[1]: VeriSign, Thawte, GlobalSign and Comodo.

Once downloaded the revocation data, we preprocess these data to remove duplicated information. For example, when a revoked certificate expires, it typically remains in the CRLs for one additional publication interval, so we preprocess the CRLs to remove these duplicates. On the other hand, Thawte's and GlobalSign's CRLs may contain duplicate entries for the same certificate because of their policy statements. These policy statements impose that a certificate that is revoked by several reasons must be included in the CRL as many times as the number of revocation reasons. Thus, we remove any duplicate entry from the composite dataset, and tally the number of revocations per day. Finally, we build a dataset that covers non-expired revoked certificates from 2008 to 2010. A summary about this dataset per CA is shown in Table I. Note that these CRLs cover two types of certificates: Server Gated Cryptography (SGC) and Code Signing Certificates (CSC, also known as a Software Publishing Certificate).

Then, from the dataset we could obtain the last update instant and the next update instant of the CRL, the serial number and the revocation date of each revoked certificate. With all this information, now we can analyze the time evolution of the number of revoked certificates per day. Fig. 1 shows the number of certificates revoked for the period of 2008/08/01 through 2010/08/21 for each CA. Analyzing all the collected data, we can conclude that:

- The number of revoked certificates bounces on a daily basis. Particularly, many revocations occur during weekdays, whereas few occur during weekends.
- There are some small peaks in the amount of certificates revoked per day. Moreover, there are also extraordinarily large spikes in certificate revocations at specific dates.
- Different CAs exhibit similar characteristics in terms of the revocation pattern. However, the mean number of revoked certificates depends on the CA's market share.
- Thawte and VeriSign exhibit a significant increment of the number of certificates revoked per day from 2008 to 2010. These changes might be mainly due to the changes in the total number of certificates being issued at different years. In this sense, as the number of certificates issued daily by Verisign increased approximately in 10 units from 2008 to 2009, the number of daily revoked certificates also increased in 1 unit in average. Thus, the percentage of revoked certificates remained fairly constant from one year to another. Similarly, this percentage also remains fairly constant for Thawte.
- GlobalSign market share is approximately five times lower than Verisign or Thawte. Therefore, the amount of revoked certificates that it manages is smaller. However, the time evolution of the number or revoked certificates follows a similar pattern, as there are also some small increases over the years.
- Comodo's market share has increased abruptly the last year, going from managing less than ten revoked certificates per day to more than a hundred. Despite this sudden increment, the autocorrelation function of the revocation process is still similar to the other three CAs.

As our goal is to obtain a generic model that fits the time evolution of the number of revoked certificates independently of the CA, we build a single time series by concatenating all four time series (see Fig. 2). By modeling the concatenation of all four time series, we will develop a generic model that will capture the revocation pattern independently of the CA. However, one of the difficulties in modeling revocation data is that the amount

[1]According to NetCraft's survey [13], using these CAs we cover most of the world market for SSL.

of revoked certificates that a CA manages depends on its market share. To analyze the global trend of the revocation process, we need to remove the influence of the volume of certificates that each CA manages, so that we can concentrate on the revocation pattern itself. One way to do this is to normalize the data. Therefore, we normalize each individual time series dividing by the maximum number of daily revoked certificates of each CA. This normalization will have to be undone once the model is obtained. Therefore, our synthetic revocation trace generator will need as input parameter the mean number of daily revoked certificates which is specific for each CA. Once all the data has been normalized, we create the global time series by concatenating the normalized data. In the following section we model this time series, obtaining a global model that fits well any CA.

IV. Modeling Revocation

In the previous section, we have analyzed the time evolution of the number of revoked certificates for each CA. So, we want to obtain a model from the revocation time-series to obtain a suitable model for generating synthetic revocation traces. For this purpose, one of the simplest techniques is to use a Multiple Linear Regression (MLR). Linear regression is useful for exploring the relationship of an independent variable to a dependent variable when the relationship is linear. However, MLR has drawbacks when the time-series exhibits high correlation. Correlation means that the value of the considered parameter at one time is influenced by values of the parameter at previous times. This happens when the values of the dependent variable over time are not randomly distributed. In our case, we will find that for the time-series of the number of daily revoked certificates, the error residuals are correlated with their own lagged values. This serial correlation violates the standard assumption of regression theory that disturbances are not correlated with other disturbances. The primary problems associated with serial correlation are:

- Regression analysis and basic time-series analysis are no longer efficient among the different linear estimators.
- Standard errors computed using the regression and time-series formula are not correct and are generally understated. If there are lagged dependent variables set as the regressors, regression estimates are biased and inconsistent but can be fixed using ARFIMA.

Another problem of MLR is that it fails to capture seasonal, cyclical, and countercyclical trends in time series. If there are lagged dependent variables set as the regressors, regression estimates are biased and inconsistent. Fortunately, this can be fixed using an autoregressive fractional integrated moving average Model (ARFIMA). Thus, as we have found that the simplest analysis, the MLR, is not suitable to develop our synthetic revocation trace generator, we will use ARFIMA.

The aim of this section is to show that the revocation process is a self-similar process that can be modeled as an ARFIMA process. To that end, we carry out the following steps:

1) Description of each component of the ARFIMA process in the Z-domain.
2) Formulation of the I component as a Laurent's series.
3) Characterization of the components of the ARFIMA process. This characterization consists of three steps:

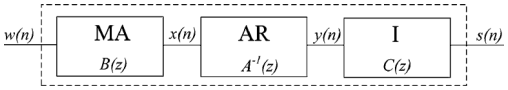


Fig. 3. Components of an ARFIMA process.

a) *Model identification*:
In this step, first we test for stationarity of the time series. Then, we determine the likely values of the order of the model, i.e., the p, d and q parameters of the ARFIMA model. The order to determine the likely values of the parameters is:
   a) d, the level of differencing. In our case, d is fractional, so we need to measure the intensity of self-similarity (H) and then we calculate d.
   b) p, the autoregression.
   c) q, the moving average.
   d) *Estimation of the ARMA components*:
   Once the order of the ARFIMA model has been determined, the values of the autoregressive component parameters and of the moving average component parameters are estimated to fit the global revocation time series.
   e) *Model Diagnostic checking*:
   Once we have identified and estimated the ARFIMA model, we assess the adequacy of the models to the revocation data. This model diagnostic checking step involves both parameter and residual analysis.

A. ARFIMA Processes in the Z-Domain

In Section II-B we have described ARFIMA processes in the time-domain. However, as any linear system, an ARFIMA process can be expressed by a difference equation involving the input series and the output series. If we Z-transform the difference equation and reorganize it, we can compute what is called the transfer function of the system.

For this purpose, we use the delay operator z^{-1} [14] to Z-transform the time-domain expression of an $ARFIMA(p,d,q)$ process. Fig. 3 shows a scheme of the ARFIMA model. Note that, as it is shown in the figure, we can express the transfer function of the $ARFIMA(p,d,q)$ process as a cascade of all three components.

The first step is to Z-transform the moving average component. A $MA(q)$ stochastic process is one that is generated using the difference equation expressed in (4). Applying the Z-transform to (4), we can express the MA process in the z-domain as:

$$B(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} \cdots + b_q z^{-q}. \tag{6}$$

Note that in the previous expression, we only use previous samples of the input signal. The main features of the associated generating system are that it is Linear time-invariant (LTI), causal and stable. The MA system is Finite Impulse Response (FIR) and, therefore, an all-zero system. Fig. 4 represents the $MA(q)$ as a FIR filter whose transfer function is $B(z)$.

An $AR(p)$ stochastic process is one that is generated using the difference equation expressed in (1). This is a quite general situation, in which it is reasonable to think that a given sample

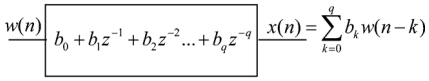


Fig. 4. MA filter.

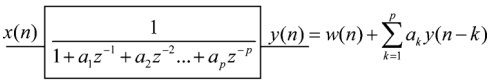


Fig. 5. AR filter.

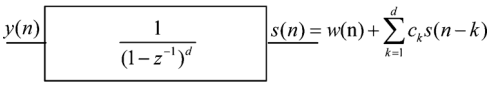


Fig. 6. Integrated filter.

of a time-series depends linearly on previous samples plus some random error. In this context, the transfer function of $AR(p)$ process in the z-domain can be expressed as:

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} \cdots + a_p z^{-p}. \tag{7}$$

The impulse response of the associated system is Infinite Impulse Response (IIR). Note that, this time, the autocorrelation is not limited and it tends to 0 when the lag tends to infinity, only if the module of all its poles is strictly smaller than 1. That means that if this condition is met, then the $AR(p)$ process is ergodic. Fig. 5 represents the $AR(q)$ as an IIR filter whose transfer function is $1/A(z)$.

Finally, we can also express the integrated component in the z-domain from (2):

$$C(z) = (1 - z^{-1})^{-d}. \tag{8}$$

In the same way as the autoregressive and moving average components of the ARFIMA process, we can represent the integrated component as a linear filter. Fig. 6 represents the $I(d)$ as a linear filter whose transfer function is $C(z)$.

Finally, the general expression of an $ARFIMA(p, d, q)$ process can be expressed by its Z-transform as:

$$S(z) = [B(z)A^{-1}(z)C(z)] \cdot W(z). \tag{9}$$

Understanding expression (9) as the relationship between the input $w(n)$ and the output $s(n)$ of a digital filter in a given instant n, the transfer function of the filter $H(z)$ could be defined as:

$$H(z) = \frac{S(z)}{W(z)} = B(z)A^{-1}(z)C(z). \tag{10}$$

It is worth noting that the factors of the transfer function follow the reverse order of the synthesis of the model. However, the order of the system in the cascade can be rearranged without affecting the characteristics of the overall combination. Hence, it is equivalent to changing the order by the commutative property of linear systems. Fig. 7 represents the ARFIMA filter with transfer function $H(z)$.

Note also that the roots of the polynomial $B(z)$ correspond to the zeros of the filter and the zeros of $A^{-1}(z)$ and $C(z)$ to

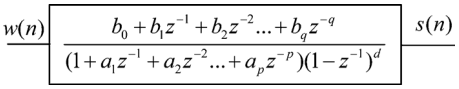


Fig. 7. ARFIMA filter.

the poles. According to the definition of the c_i values expressed in (3), the integrated order defines the multiplicity of the pole in $z = 1$. This pole generates the instability of impulsional response. The rest of obtained poles (z_k) will be found in the unit circle $(|z_k| < 1)$ of the Z plane.

B. Integrated Component as an Infinite Series

Once we have derived the expression of each one of the components of the ARFIMA process, we simplify the calculation of this integrated part to make an easier characterization of these components. This simplification will allow us building a much simpler generator as we will use sums and multiplications instead of gamma functions.

Thus, we show that we can use the Laurent's series to simplify the calculation of the integrated part expressed in (8). To that end, we show that the autocorrelation function of the coefficients c_i using gamma functions as in (3) is quite close to the autocorrelation function using (12) when limiting the number of coefficients.

A pure integrated process has the following transfer function defined in 8. According to Newton's generalized binomial theorem [15], $\left(1 - z^{-1}\right)^d$ is analytic in the open disk $\{z \in \mathbb{C} \mid |z| < 1\}$ for every $d \in \mathbb{C}$ and converges at $|z| = 1$ only for $d < 0$. Thus, the filter $C(z)$ can be expanded to the infinite series and we get the $MA(\infty)$ representation:

$$C(z) = \sum_{i=0}^{\infty} c_i z^{-i}, \tag{11}$$

where c_i is derived from the Laurent's series:

$$c_i = (-1)^i \frac{(-d)(-d-1)\ldots(-d-i-1)}{i!}, \tag{12}$$

and they can be related recursively as follows:

$$\begin{aligned} c_0 &= 1, \\ c_i &= c_{i-1} \frac{d+i-1}{i}. \end{aligned} \tag{13}$$

Moreover, these coefficients constitute the impulsional response $h(n)$ of the $ARFIMA(0, d, 0)$ filter. Fig. 8 shows the equivalence of the filters.

Let $y(n)$ be an uncorrelated process at the input of the filter, and $s(n)$ the output of the same filter. Then:

$$s(n) = y(n) * h(n). \tag{14}$$

Thus, the autocovariance functions satisfy:

$$K_{ss}(n) = K_{yy}(n) * r_{hh}(n), \tag{15}$$

where $K_{xx}(n)$ represents the autocovariance of the process $x(n)$. The input is an uncorrelated signal, so:

$$K_{yy}(n) = \delta(n), \tag{16}$$

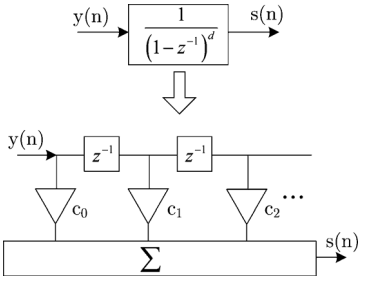


Fig. 8. Equivalent filters.

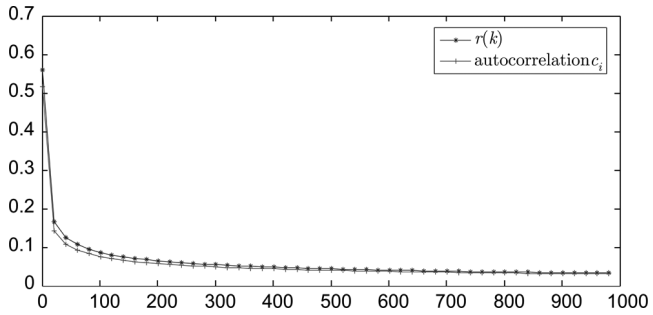


Fig. 9. Autocorrelation comparison.

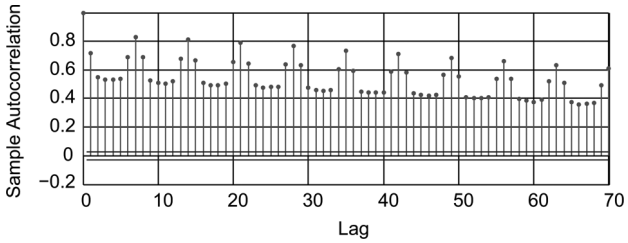


Fig. 10. ACF of the composite time series.

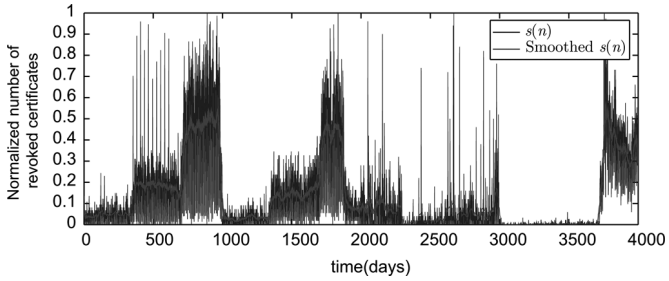


Fig. 11. Time evolution of the mean number of revoked certificates per day.

where $\delta(n)$ is the Dirac's delta function. Therefore:

$$K_{ss}(n) = K_{hh}(n). \tag{17}$$

Thus, the autocovariance of the output is the same as the autocovariance of the coefficients in expression (12). Note that as the autocorrelation is just a normalization of the autocovariance, the autocorrelation of the output is also the same as the autocorrelation of the coefficients c_i. Moreover, the output of the filter is a second order self-similar process.

Knowing that the ACF of an asymptotic second order self-similar process with self-similar parameter H is [8]:

$$r(k) = \frac{1}{2}\left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right] \quad \forall k = 1, 2, 3, \ldots. \tag{18}$$

Therefore, the autocorrelation of the coefficients c_i must fit also the expression given in (18). In order to validate this study, the autocorrelation of 50000 coefficients is compared with the expression (18) with $H = 0.8$ $(d = 0.3)$ in Fig. 9. As expected, the autocorrelation of the coefficients using the Laurent's series is quite close to the autocorrelation of the coefficients using the exact expression as in (3). One of the main goals of Section V will be to obtain a valid bound for the minimum number of coefficients needed to obtain the desired LRD at the output of the ARFIMA filter.

C. ARFIMA Model Identification

In this section, we identify all the necessary components and parameters of the ARFIMA model. To that end, we need to follow three steps.

1) Testing for Stationarity: Broadly speaking, a time-series is said to be stationary if there does not exist systematic change in the mean (no trend), if there does not exist systematic change in the variance, and if strict periodic variations have been removed. To test stationarity, we analyzed the Autocorrelation Function (ACF) of the global revocation time series of our dataset. The ACF plot is a plot of the partial correlation coefficients between the series and lags of itself.

This ACF function is shown in Fig. 10. As observed, the ACF of the time series decreases slowly, which is an indication that the mean is not stationary. Notice also that there exists certain seasonality in the ACF. This seasonality is mainly due to the fact that the number of revoked certificates decreases during the weekends. In addition, the temporal series of the number of revoked certificates (see Fig. 2) presents great variations. Thus, the first visual analysis suggests that the time series of the number of daily revoked certificates is non-stationary. To confirm this, we perform a unit root test. Those tests are based on the idea that a linear stochastic process has a unit root if 1 is a root of the process's characteristic equation. In such a case, a process is non-stationary. If the other roots of the characteristic equation lie inside the unit circle—that is, have a modulus (absolute value) less than one—then the first difference of the process will be stationary. We choose the KPSS test [16] at the 99% confidence level which rejects the null hypothesis of stationarity.

2) Finding the Integrated Component: Once we have confirmed that the time series is non-stationary, we have to find the integrated component, i.e., find d. The long range dependence complicates the characterization of the model because the temporal series shows an apparent non-stationary mean. This non-stationary mean can be observed in Fig. 11. To show the evolution of the mean number of revoked certificates, we have used a moving average filter with a 30-days span to smooth all of the data at once (by linear index). The long-range dependence produces that the mean varies. In order to characterize the ARFIMA model, it is necessary to capture this long range effect. Using the relationship between the index of self-similarity and the degree of differencing (see (5)), we can capture the LRD by means of the Hurst parameter.

TABLE II
HURST PARAMETER ESTIMATION FOR EACH CA USING DIFFERENT ESTIMATION METHOD

CA	Hurst Value			95% Confidence Interval		
	Whittle	R/S	Agg. Var.	Whittle	R/S	Agg. Var.
VeriSign	0.83	0.81	0.80	[0,79; 0,88]	[0,73; 0,84]	[0,72; 0,85]
Thawte	0.88	0.79	0.83	[0,83; 0,92]	[0,77; 0,81]	[0,79; 0,90]
GlobalSign	0.61	0.71	0.69	[0,56; 0,65]	[0,68; 0,75]	[0,67; 0,71]
Comodo	0.89	0.82	0.78	[0,80; 0,98]	[0,76; 0,88]	[0,73; 0,81]

However, while the Hurst parameter is perfectly well defined mathematically, its estimation is problematic. The accuracy and robustness of the H estimators can be influenced by processes such as periodicity, trend, length of the time series and short-range correlations which have different effects on different estimators [17]–[19] and can lead to erroneous estimation of the LRD intensity or even to reporting LRD on non-LRD series. This causes important problems when finding the most appropriate "H estimation" for the current time series. At present several methods to identify self-similar processes are known [20]. The most popular approaches are the following: analysis of R/S (rescaled adjusted range) statistics, analysis of the variance-time plot, analysis based on specific properties of $\mathrm{S}(\omega)$, Whittle estimation and analysis based on aggregated variance (see [41] for a comparison analysis). No single LRD estimator has been proved to produce more accurate estimates than the rest. We choose to use the Whittle estimation because it is based in the calculation of the FFT that has a numerical complexity of order $O[nlog_2(n)]$, so that it produces very fast algorithms for computing parameter estimations.

The Whittle method proposed in [21] calculates the Hurst index of self-similarity using the asymptotic properties of the spectral density. This method is based on a frequency domain maximum likelihood estimation of a fractionally integrated process for determining d. We estimate the Hurst parameter for all four CAs. The results are shown in Table II. It is worth noting that there exist notable differences in the Hurst value depending on the estimation method. As recently studied in [41], in general, different aspects must be taken into account before choosing the algorithm. For short inputs, only the R/S algorithm is suitable. As it is prone to noise, this algorithm must be used with care. For large input the Whittle method and wavelet-based algorithms will show better overall performance. As our dataset is quite large, we chose the Whittle method because its accuracy and simplicity. These results show that the Hurst parameter shows small variations depending on the CA. Despite these small variations, it is always above 0.5 (in the 0.61–0.89 range) which indicates the existence of the long-time correlation in the number of revoked certificates.

To build our general model, we will set the H parameter equal to the mean value of the estimated parameter $\hat{H}$ for each CA, i.e., $\hat{H} = 0.8$. The Hurst parameter is used to describe the degree of LRD and the burstiness of the traffic. Therefore, accurate characterization of LRD is very important in order to predict performance of the revocation service and to allocate network resources to provide a secure and reliable revocation mechanism. Our measured data show dramatically different statistical properties than those predicted by the stochastic models currently considered in the literature like Poisson or MMP. Almost

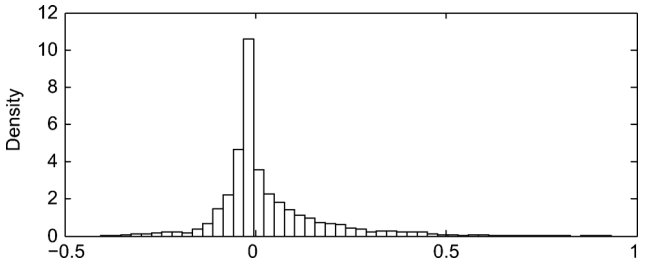


Fig. 12. PDF of the extracted integrated component time series.

all these models are characterized by an exponentially decaying autocorrelation function. As a result, they give rise to a Hurst parameter estimate of $H = .50$, producing variance-time curves, R/S plots, and frequency domain behavior strongly disagreeing with the self-similar behavior of actual revocation data. It is worth mentioning that the model derived will be rather insensitive to H (for the range of values that takes the analyzed CAs). So, the analysis of the Hurst parameter allows us to conclude that the integrated component of the model should be of order $\hat{d} = \hat{H} - 0.5 = 0.3$. Therefore, we can express its associated transfer function as $C(z) = (1 - z^{-1})^{-0.3}$.

3) Selecting the Parameters of the ARMA Process: Once we have obtained the value of d, we have to calculate the order of the autoregressive p and the order of the moving average q. To do so, it is necessary to extract the integrated component of the actual process $s(n)$. According to the scheme presented in Fig. 3, the residual ARMA series $y(n)$ and the real series $s(n)$ are related as follows:

$$Y(z) = C(z)^{-1}S(z) = \left(1 - z^{-1}\right)^{d} S(z). \tag{19}$$

Hence, the temporal series $y(n)$ can be obtained by means of a deconvolution filter. Deconvolution is, therefore, the reverse process in which an unknown input $Y(z)$ is calculated from the measured output $S(z)$ and a known transfer function $C(z)$ (approximated by the Laurent's series in expression (12) with 1,000 coefficients) . Fig. 12 shows the probability density function (PDF) of the obtained time-series $y(n)$. Note that this PDF does not fit exactly a normal distribution as there is a high peak around zero.

At this point, we have to check whether the time series without the integrated component is white noise or not. We check (at a confidence level of 99% and 70 lags) that $y(n)$ is not white noise by means of Ljung-Box Q-test [7]. The visual analysis of the autocorrelation of $y(n)$ confirms the result of this test (see Fig. 13). The fact that $y(n)$ is not white noise allows us to model it by means of an ARMA process.

Finally, we need to determine the best ARMA model that fits $y(n)$. We use the Rissanen's Minimum Description Length (MDL) criterion [22] for model selection among the different set of ARMA models with different numbers of parameters. It must be noted that information criteria penalize models with additional parameters. Therefore, the MDL model order selection criteria are based on parsimony, i.e., we adopt the simplest model that capture the self-similar pattern in accordance with the rule of Ockham's razor.

Varying the order of the AR component, we show in Fig. 14 the goodness of fit of each AR model. Note that each bar in the

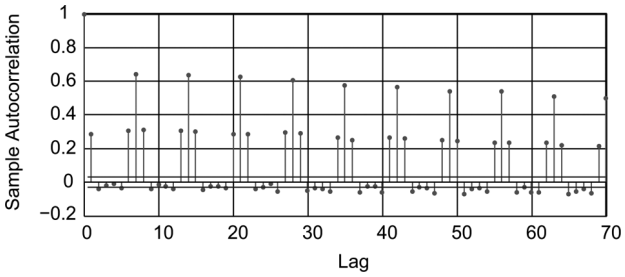


Fig. 13. ACF of the extracted integrated component time series.

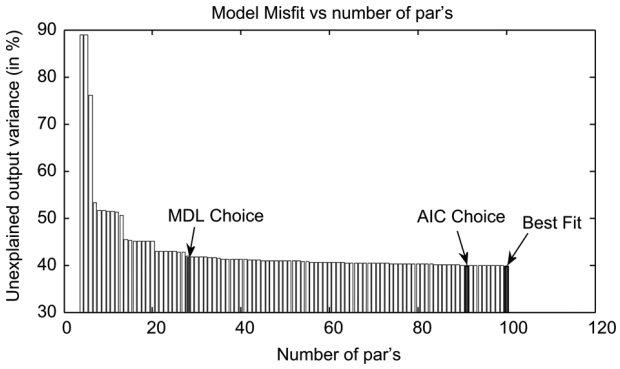


Fig. 14. AR parameter order estimation.

figure corresponds to an AR model with certain orders and delays. The x-axis shows the number of parameters in the respective models. The y-axis shows the part of the output variance, which is not explained by the model. That is, the ratio between the prediction error variance and the output variance in percent. Only that model that has the best fit for the given number of parameters is displayed. Therefore, using the MDL criteria, the order of the AR process found is $p = 29$ (see Fig. 14). In the next section, we calculate the value of the 29 coefficients of the AR component as well as the order and values of the MA coefficients.

D. Estimation of the ARMA Coefficients

Once we have identified the orders p and d of the ARFIMA model, we use the least square estimation to calculate the coefficients of the AR. After this calculation, we obtain the following coefficients:

$$\begin{aligned} A(z) = 1 &- 0.6467z^{-1} + 0.02693z^{-2} + 0.09085z^{-3} \\ &+ 0.09753z^{-4} + 0.1218z^{-5} + 0.1991z^{-6} \\ &- 0.804z^{-7} + 0.6906z^{-8} + 0.03223z^{-9} \\ &- 0.04807z^{-10} - 0.007471z^{-11} - 0.0759z^{-12} \\ &- 0.08934z^{-13} - 0.07605z^{-14} - 0.006487z^{-15} \\ &- 0.02565z^{-16} - 0.01994z^{-17} - 0.04003z^{-18} \\ &- 0.05007z^{-19} - 0.01331z^{-20} - 0.07361z^{-21} \\ &- 0.001947z^{-22} - 0.02836z^{-23} - 0.01824z^{-24} \\ &- 0.03693z^{-25} + 0.007019z^{-26} - 0.07691z^{-27} \\ &- 0.01872z^{-28} - 0.03821z^{-29}. \end{aligned} \tag{20}$$

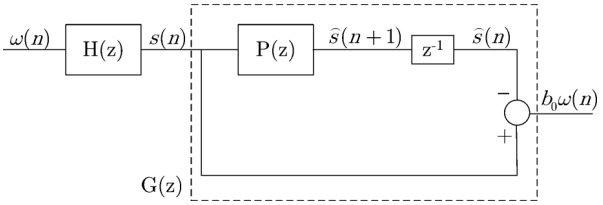


Fig. 15. Block diagram of the residual series estimation.

To obtain the MA component, the AR component of $y(n)$ is withdrawn. As shown in Fig. 3, the $x(n)$ series can be derived using the relation between the $y(n)$ and $x(n)$ series:

$$X(z) = A^{-1}(z)Y(z). \tag{21}$$

Again, we use a deconvolution to obtain $x(n)$ from the known inputs $y(n)$ and $a(n)$. To estimate the parameters of the MA process, the least square estimation is applied to fit the partial autocovariance function of $x(n)$ [7]. We use the same criteria as in the AR process for selecting the parameters of the MA process and the best adjustment is obtained with a MA process with order $q = 8$. Then, we calculate the MA coefficients, which yield the following result:

$$\begin{aligned} B(z) = 1 &- 0.6454z^{-1} + 0.005554z^{-2} \\ &+ 0.1113z^{-3} + 0.1317z^{-4} \\ &+ 0.1032z^{-5} + 0.2802z^{-6} \\ &- 0.6652z^{-7} + 0.6688z^{-8}. \end{aligned} \tag{22}$$

At this point, we have completely characterized the global revocation process as an $ARFIMA(29, 0.3, 8)$ process.

E. ARFIMA Model Diagnostic Checking

Now, we must check that the $ARFIMA(29, 0.3, 8)$ model fits the global revocation process. For that purpose, we analyze the residual series $w(n)$. The residual series can be calculated from the autoregressive component, the moving average component and the integrated component. In the following, we derive an expression of this residual series in the z-domain and analyze its statistical characteristics.

First, we express together the integrated and the moving average components in the following way:

$$B'(z) = \frac{B(z)}{C(z)} = B(z)\left(1 - z^{-1}\right)^d. \tag{23}$$

From [23] we know that:

$$s(n) - \hat{s}(n) = b_0 w(n). \tag{24}$$

Using (24), we define the transfer function of the diagnostic checker $G(z)$:

$$G(z) \triangleq b_0 \frac{W(z)}{S(z)}. \tag{25}$$

The relationship between the ARFIMA components and the residual series are shown in the scheme in Fig. 15.

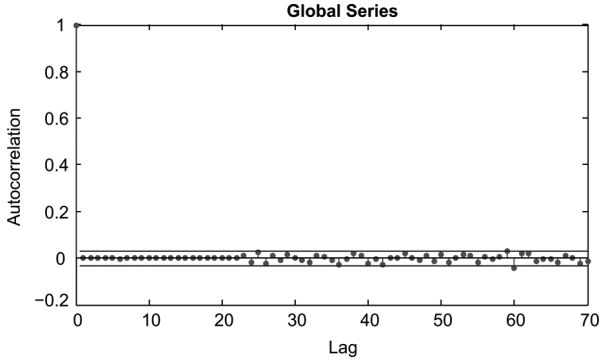


Fig. 16. ACF of the residual series for the global revocation process.

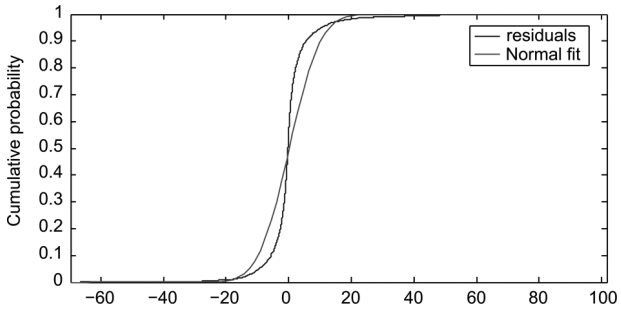


Fig. 17. CDF of the residual series versus Normal distribution.

From Fig. 15, we can express the residual series in the z-domain as:

$$W(z) = \frac{S(z)G(z)}{b_0}, \tag{26}$$

where $G(z)$ is

$$G(z) = 1 - P(z)z^{-1}, \tag{27}$$

and where $P(z)$ is the transfer function of the predictor.

Moreover, as it is shown in Fig. 15, the input of the system is $w(n)$ and the output is $b_0 w(n)$. Therefore:

$$H(z)G(z) = b_0. \tag{28}$$

Then, replacing (23) in (10), we can rewrite the transfer function of the ARFIMA filter as:

$$H(z) = \frac{B'(z)}{A(z)}. \tag{29}$$

Using (28), (29) and (27), we can express $P(z)$ as:

$$P(z) = \frac{B'(z) - b_0 A(z)}{z^{-1}B'(z)}. \tag{30}$$

Finally, replacing (30) and (27) in (26), we can express the residual series as a function of the ARFIMA components and the revocation series:

$$W(z) = \frac{S(z)(\frac{B'(z)}{b_0} - A(z))}{z^{-1}B'(z)}. \tag{31}$$

Once we have obtained the relationship between the residual series $w(n)$ and the ARFIMA components, we analyze the autocorrelation of the residuals. Fig. 16 presents the residuals' autocorrelation and the 99% confidence intervals. The residual diagnostic determines that the residuals are highly uncorrelated. Fig. 17 presents the CDF of the residuals' autocorrelation and the CDF of standard Gaussian distribution with its 99% confidence intervals. Notice that the residuals differ from the expected Gaussian distribution. The fact that the marginal distribution differs from the normal distribution will be taken into account during the design of the synthetic trace generator in the next section.

So far we have seen that the model fits quite accurately the global revocation process, next we check its suitability for each CA. For this purpose, we analyze the residuals for the individual revocation processes using the ARFIMA model obtained from the global series. Fig. 18 presents the residuals autocorrelation and the 99% confidence intervals for each CA. Although some residuals exceed the confidence intervals, the ARFIMA model still remains valid for each CA. Therefore, we can conclude that the proposed model is quite insensitive to Hurst variations, as it is able to fit well different revocation processes with different Hurst parameters.

Finally, in order to check that the model is not unspecified, we run again the Ljung-Box Q-test at a confidence level of 99% and 70 lags. The test results in the acceptance of the null hypothesis that the model fit is adequate (no serial correlation at the corresponding element of lags).

V. Revocation Traces Generation

Remember that the main goal of this work is to be able to generate synthetic revocation traces that mimic the self-similar behavior of real revocation data. To that end, we need to set a minimum bound for the number of coefficients required in the integrated component in order to achieve the desired LRD. For this purpose, at the input of the ARFIMA model, a Gaussian noise with zero mean and variance equal to 1, is applied. The number of coefficients in the integrated component is increased from 30 to 50,000 coefficients. The chosen value for the Hurst parameter is 0.8 ($d = 0.3$). The LRD level obtained in each case was measured using the Whittle method mentioned above. Table III shows the results of these tests. With 1,000 coefficients, the obtained value for $\hat{H}$ is 0.815, which is quite close to the expected value. Therefore, with only 1,000 coefficients of the equivalent filter, the generated synthetic series achieves a degree of self-similarity quite close to the expected one. Again, note that the model is rather insensitive to the H value.

Finally, to generate revocation traces the marginal distribution of our model must be adjusted to the probability distribution of the number of revoked certificates. In this step we find a problem. It is known that if the input of an ARFIMA model is Gaussian, the output will be also Gaussian. Furthermore, if the input is not Gaussian, then the output will not be Gaussian. As it was seen in the previous section, the marginal distribution of the number of revoked certificates is far from behaving as a normal distribution, rather it can be approximated as an exponential distribution (see [4]–[6] and Fig. 19). Thus, we will need to transform the normal probability distribution of our synthetic

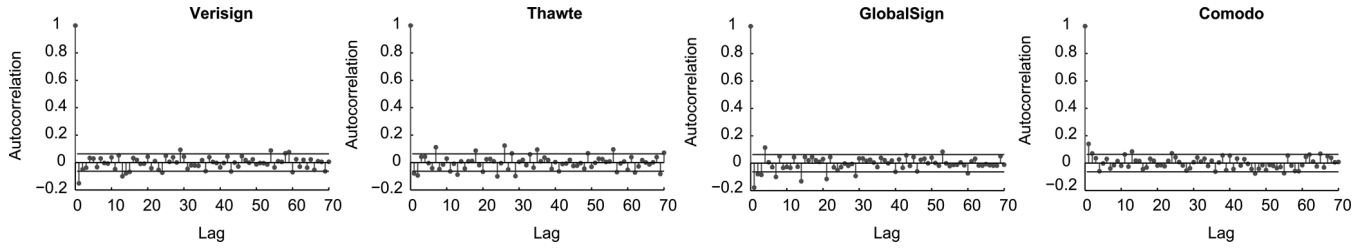


Fig. 18. ACF of the residual series for each CA.

TABLE III
HURST PARAMETER VERSUS NUMBER OF COEFFICIENTS

Number of coefficients	30	50	100	2,000	50,000
Hurst Parameter Value	0,899	0.852	0.827	0.809	0.806

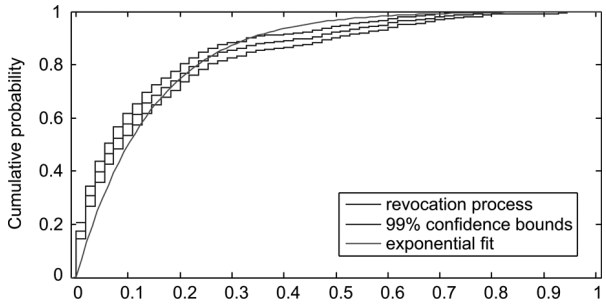


Fig. 19. CDF of the revocation process versus an exponential fit.

traces generator into an exponential distribution. For this purpose, we will use a zero-memory nonlinear (ZMNL) function as in [24], [25].

In more detail, we are going to use a monotonic ZMNL function $g(\cdot)$ relating the Gaussian distribution to the desired realization. This ZMNL is based on the cumulative distribution function of the revocation process and the known CDF of a Gaussian distribution. The established function is used to transform the Gaussian distribution into a realization of the exponential distribution. Since the transformation preserves the times of the zero-crossings and peaks of the original Gaussian distribution, and does not introduce any substantial discontinuities, the power spectral density is not substantially changed [26]. As the PDF of the revocation process can be approximated by an exponential distribution, the resulting CDF can be used with the known distribution of a Gaussian distribution to establish a ZMNL transformation function. This function relates a random variable with a Gaussian distribution to a random variable with exponential distribution.

Fig. 20 shows the block diagram of the synthetic revocation generator, where the ZMNL function is placed at the output of the ARFIMA filter. The values of the white noise sequence $w(n)$ at the input of the ARFIMA filter are chosen such that $Var(w(n)) = 1$ and $E[w(n)] = 0$. In turn, the output of the ARFIMA filter $s(n)$ becomes the input of the ZMNL function. In this way, the ARFIMA model transforms the colored $N(0,1)$ sequence in a colored $N(0,\sigma_s^2)$ sequence. Then, the ZMNL function transforms the colored $N(0,\sigma_s^2)$ sequence in

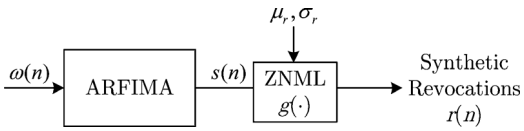


Fig. 20. Synthetic Revocation trace generator.

an $Exponential(\mu_r)$ sequence, where μ_r is the measured average of daily revoked certificates.

Hence, the ARFIMA filter output $s(n)$ is followed by a ZMNL $g(\cdot)$, which is chosen so that:

1) All the random samples $g(s(n))$ are positive.
2) The expected value $E[g(s(n))]$ matches well with the average of daily revoked certificates μ_r.
3) The influence on the autocovariance function of the synthetic sequence is low.

The conditions i) and ii) are obviously necessary in order to convert the Gaussian sequence (output of the ARFIMA filter) to a sequence which represents revocations. Fulfilling condition iii) allows to fit the measured autocovariance function directly to the ARFIMA filter. Thus, with ARFIMA and ZMNL together, it becomes possible to recreate the self-similar nature of the revocation process.

In order to fulfill the aforementioned conditions, we use a series of algebraic manipulations to obtain the desired ZMNL function that transforms $s(n)$ to an exponential distribution with specific parameters. First, we use the probability integral transformation that allows to convert the Gaussian distribution $(s(n))$ to a uniform distribution $(u(n))$:

$$u(n) \triangleq \int_{-\infty}^{s(n)} e^{-x/2\,\sigma_s^2} dx. \tag{32}$$

Once we have a uniform distribution, we convert it to an exponential distribution $(e(n))$ applying a logarithmic transformation (see Inverse Transformation Techniques in [27]).

$$e(n) \triangleq -\mu_r ln(u(n)). \tag{33}$$

Then, we modify the mean and variance of the exponential distribution in order to fit the desired parameters of the synthetic revocation process. Finally, we truncate the output allowing only nonnegative integers. Thus, the ZMNL function can be expressed as:

$$g(s(n)) = \max\left(0, \left\lceil a - b \cdot ln\left(\frac{1}{\sqrt{2\pi\,\sigma_s^2}} \int_{-\infty}^{s(n)} e^{-x/2\,\sigma_s^2} dx\right)\right\rceil\right). \tag{34}$$

TABLE IV
MEAN AND STANDARD DEVIATION OF THE NUMBER OF REVOKED CERTIFICATES FOR EACH CA

Certification Authority	μ_r	σ_r
VeriSign	15.7077	18.1256
Thawte	7.9970	10.1251
GlobalSign	2.7658	5.4362
Comodo	47.8001	42.6027

where the parameters a and b have to be chosen so that the measured mean and variance match well with that of the revocation process. In this manner,

$$a = \mu_r, \quad b = \mu_r \frac{\sigma_r}{\sigma_s},$$

where μ_r and σ_r are the mean and standard deviation of the synthetic revocation process to generate respectively.

Note that the variance of the $s(n)$ only depends on the value of the coefficients of the ARFIMA filter. In short, the amplitude of the autocovariance at zero displacement provides a measure of the signal magnitude [14]. This variance can be expressed as:

$$\sigma_s^2 = K_{hh}(0).$$

The expression of the $g(\cdot)$ in (34) can be simplified using the Q-function [28] to facilitate the implementation of the ZMNL function:

$$g(s(n)) = \max\left(0, \left\lceil a - b \cdot ln\left(1 - Q\left(\frac{s(n)}{\sigma_s}\right)\right)\right\rceil\right). \quad (35)$$

Using the Chernoff bound [29] of the Q-function, we can express the $g(\cdot)$ as:

$$g(s(n)) \simeq \max\left(0, \left\lceil a - b\left(ln(2) + \frac{s(n)^2}{2\sigma_s^2}\right)\right\rceil\right). \quad (36)$$

The ZMNL function allows to amplify the output of the fractional ARIMA model to tailor the standard deviation of the synthetic revocation process. In the same way, the mean of the revocation process is increased allowing to model different revocation rates. Note that the mean number of revoked certificates per day (μ_r) varies depending on the CA. Therefore this parameter must be set according to Table IV.

It is worth noting that the mean number of revoked certificates per day (μ_r) follows a specific pattern. Mainly, this parameter depends on the market share of the certification authority. Using the market share of each CA [30] as exogenous variable, we can perform a simple lineal regression analysis to estimate the endogenous variable μ_r. The result of the regression analysis using least squares is:

$$\mu_r = -10.2 + 3.22\ Market\ Share(\%).$$

The p-value of the regression analysis ($p = 0.099$) indicates that the relationship between μ_r and $MarketShare(\%)$ is statistically significant at an α-level of 0.1. This is also shown by the p-value for the estimated coefficient of $MarketShare(\%)$, which is 0.102. Analysis the coefficient of determination R^2 value, it shows that the $MarketShare(\%)$ explains 59,8% of the variance in μ_r, indicating that the model fits the data well. Because the model is significant and explains a large part of the variance in μ_r, the generator could have as input the $MarketShare(\%)$ of the desired CA instead of μ_r.

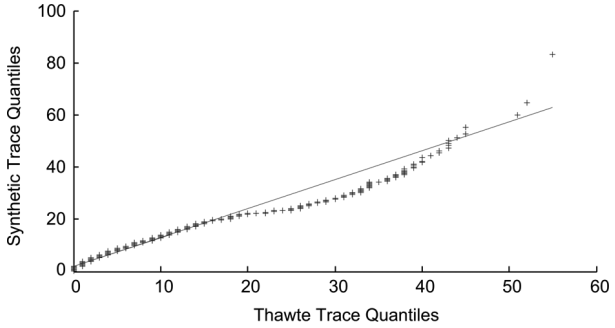


Fig. 21. Quantile-to-quantile plot of the generated synthetic trace and Thawte's revocation trace.

Summing up, using as ZMNL function (36), and concatenating it to the ARFIMA filter as in Fig. 20, we are able to generate synthetic revocation traces that mimic the actual behavior of the revocation process.

A. Quality of the Traces

In the following, we present a summary of a comprehensive study that evaluates the quality of the synthetic revocation traces. The goal of this study is to prove that the method really synthesizes revocations corresponding to an $ARFIMA(29, 0.3, 8)$ process. We show that the proposed generator produces synthetic revocations that are indistinguishable for practical purposes from those corresponding to actual revocations. Therefore, its marginal distribution is exponential and its autocorrelation function adequately fits the actual ACF, the estimation of H is close to the true value, and its spectral density is consistent with ARFIMA.

Exponentiality. The synthesized revocation traces follow an exponential distribution for practical purposes. In order to analyze the marginal distribution we have carried out the χ^2, Kolmogorov-Smirnov, and Anderson-Darling goodness-of-fit tests with satisfactory results. Moreover, visual tests such as the QQ-plot show that the traces follow quite well an exponential distribution (see Fig. 21).

Correlation Structure. To see how the autocorrelation function (ACF) of the revocation traces fits the correlations of an actual revocation trace, we generate a sequence based on Thawte's revocation statistics. Fig. 22 compares the ACF of the generated trace and the ACF of Thawte's revocation trace. The slope of both functions for large lags, which is related with the LRD level, is very similar. This allows to confirm the goodness of fit.

Spectral Density. Our synthesizing method easily passes the strict Beran goodness-of-fit test for the spectral density [11]. In this sense, the percentage of rejections is always lower than or equal to the level of significance.

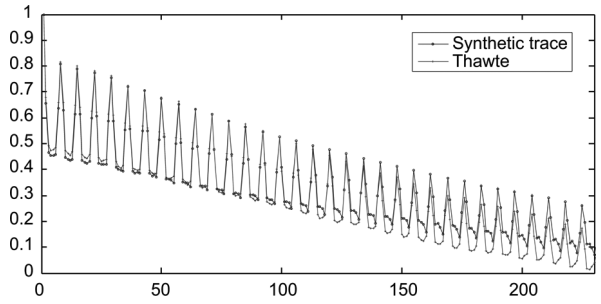


Fig. 22. ACF of a generated trace versus ACF of the revocation time series from Thawte.

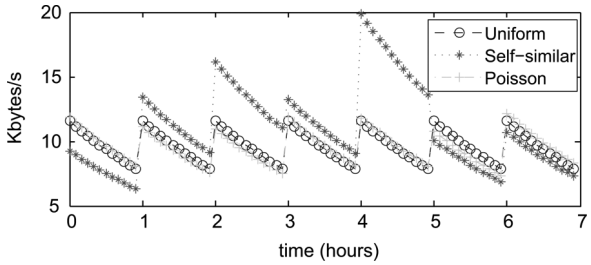


Fig. 23. BW consumption depending on the revocation process distribution.

VI. Synthetic Revocation Traces

In this section we describe the potential applications of synthetic revocation traces, because they are desirable for many reasons. Firstly, thorough revocation data are difficult to obtain, mostly because of security or privacy concerns. Only a few CAs (e.g., Verisign) allow to obtain additional information about the revoked certificates such as the issued time, the country or the issuer. However, revocation traces are necessary to evaluate the performance of certificate status validation mechanisms. To compare the performance of different mechanisms, researchers run simulations based on theoretical assumptions. For example, Naor and Nissim calculate the communication cost by assuming a fixed length CRL [31]. Cooper [32] and Arnes [33] model the distribution of revocation information by assuming an exponential inter-arrival probability for the requests for CRLs. Their theoretical assumptions turn their results into qualitative information, as they do not use neither real nor synthetic revocation traces to develop their models.

Secondly, there do not exist CRLs large enough to test the revocation mechanism proposed for new environments (e.g., VANETs) where these lists are expected to contain millions of revoked certificates [34]. In this sense, revocation mechanisms proposals for VANETs (e.g., [35]–[37]) that are tested without taking into account the self-similarity pattern of the revocation process will not be completely accurate.

Finally, synthetic trace generation is appealing because is simple, fast, and controllable. The key issue is the fidelity of a synthetic revocation trace, i.e., how well it mimics the relevant statistical properties of real revocation data. The synthetic revocation traces must fit not only the probability distribution but also the temporal correlation of the revocation process.

VII. Related Work

Most of previous works are not based on any empirical analysis of real-world data; instead, they focus on theoretical aspects of certificate revocation including the cause of revocation [38], the model of revocation [32] and communication cost of revocation [31]. However, these theoretical models are not able to capture the actual behavior of the revocation data. Most recently, authors have studied the statistical characteristics of real revocation data [4]–[6]. However, the bursty pattern of the revocation process has been neglected. In the previous sections we have shown that revocation data is statistically self-similar.

Regarding the traditional way of issuing CRLs, X.509 defines one method to release CRLs. This method involves each CA periodically issuing CRLs. Using this method the issued CRLs will contain a number of revoked certificates that will differ significantly from one CRL to another CRL. Thus, each CRL will have a different size, and the transmission of these lists will be bandwidth inefficient. Ma *et al.* in [5] already pointed out the inefficiencies of the traditional method, and proposed releasing CRLs based on a series of economic and liability costs. However, when they defined the function that calculates the number of new certificate revocations between two days, they assumed a Poisson process neglecting the burst pattern. Therefore, the CRL release policies they obtained could be highly improved by taking into account the self-similarity of the revocation process. Specifically, the liability costs could be reduced by issuing CRLs in order to palliate the burst pattern of the revocation data. To give an idea of the impact of using synthetic revocation traces, we analyze the work of Cooper in [32] and in [39]. In these works, Cooper analyzed the best way to issue CRLs, segmented CRLs and delta-CRLs in order to decrease the request peak bandwidth. The author assumed that an average of 1,000 certificates are revoked each day and that the CRLs have a fixed validity time. By doing these assumptions, the self-similar behavior of the revocation process is neglected and the results need to be adapted to the reality. From [39], we can calculate the bandwidth for a delta-CRL system can be computed as:

$$B = \frac{Nve^{-vt}((51 + 4.5rL_c)e^{-(w+l/O-l)v} + (51 + 9rw))}{(O-1)1 - e^{vl/O} + 1},$$

where N is the number of valid certificates, v is the validation rate, l is the amount of time that a delta-CRL is valid, L_c is the certificate lifetime, r is the number of certificates revoked per day, w is the window size of the delta-CRL and O is the number of delta-CRLs that are valid at any given time.

Using the bandwidth as comparison metric, we can evaluate the impact of the self-similarity. Fig. 23 shows the bandwidth necessary to download the revocation data using a sliding window delta-CRL scheme. We have assumed that there are 300,000 relying parties each validating an average of 10 certificates per day; delta-CRLs are issued once an hour, are valid for 4 hours, and have a window size of 9 hours. We have also assumed that an average of 10 certificates are revoked each day and that certificates are valid for 365 days. Note that depending on the distribution of the revocation process, the required bandwidth presents significant variations. Uniform and poisson distributions present a similar behavior. On the opposite, a self-similar process makes the delta-CRL's size to vary. Thus, the optimal window to issue delta-CRLs should be calculated taking into account the bursty pattern of the self-similar process. If this pattern is neglected, the peak bandwidth will vary with each delta-CRL issuance making the revocation service bandwidth-inefficient. When with a poisson or uniform

process the maximum peak bandwidth is of $\sim 12\ \mathrm{Kb/s}$, a burst of revocation events causes that during the 15th delta-CRL issuance there are required $\sim\ 20\ \mathrm{Kb/s}$. Therefore, ignoring the self-similar pattern of the revocation process will led to inaccurate network planning.

Authors in [5], [6] suggest a functional form for the probability density function of certificate revocation requests. They choose a exponential distribution function because it adequately approximates the data they collected from a single CA. Based on this assumption, they provide an economic model based on which a CA can choose what they state to be the optimal CRL release interval. However, they do not take into account the self-similar behavior of the revocation data. Intuitively, the critical characteristic of this self-similar pattern is that there is no natural length of a "burst" of revoked certificates: at every time scale ranging from a few days to weeks and months, similar-looking revocation bursts are evident. This bursty pattern could be taken into account by CAs to derive better strategies to release CRLs.

Walleck *et al.* in [4] carried out a deeper empirical analysis of the revocation data not only taking into account the number of revoked certificates, but also other variables such as geographical factors. They also conclude that their collected CRLs exhibit exponential distribution patterns. Though they acknowledge the existence of revocation burst, they do not capture this fractal behavior.

This self-similar or apparently fractal-like behavior of revocation data is very different both from currently considered formal models (e.g., pure Poisson or Poisson-related models). These differences require a new look at modeling the data and performance of the revocation service. For example, our analysis of the revocation data shows that the generally accepted argument for the "Poisson-like" nature of revocation events, namely, that the number of revoked certificates becomes smoother (less bursty) as the number of certificate sources increases, has very little to do with reality. In fact, the burstiness (degree of self-similarity) is expected to be intensified as the number of active certificate sources increases, contrary to commonly held views. Thus, in novel environments such as VANETs, where digital certificates will be used to provide anonymity, increasing the number of valid certificates, the self-similar behavior of the revocation data cannot be neglected. Thus, self-similarity is both ubiquitous in our data and unavoidable in future, and more in highly populated networks. However, none of the currently common formal models for revocation data is able to capture it.

VIII. Conclusions

In this paper, we have analyzed real empirical data collected from the leading CAs. The main findings of our analysis about the revocation process are that (I) this process is statistically self-similar (irrespective of when data were collected during the 3-year period 2008–2011 or from which CA), (II) the degree of self-similarity, which can be measured in terms of the Hurst parameter H, is a function of the overall utilization of the revocation service and can be used for measuring the "burstiness" of the revocation process (i.e., the more bursts in the revocation process the higher H), and (III) the leading CAs share similar Hurst parameters even though they operate in different market segments.

Moreover, as none of the currently common formal models for revocation is able to capture the self-similar nature of real revocation data, we have developed a method for modeling this behavior. The proposed model is based on an ARFIMA process, that provides an accurate and parsimonious model. In this context, this research represents a step towards linking empirical observations to mathematical models in description of the complex process of certificate revocation. We believe that this is going to be necessary in traditional scenarios with a high number of users as well as in incipient certification scenarios such as vehicular communications, in which CAs will have to deal potentially billions of issued certificates.

Finally, for practical purposes, we have shown how the developed model can be easily used as a synthetic revocation generator. We have also shown that our model produces synthetic revocations that are indistinguishable for practical purposes from those corresponding to actual revocations.

References

[1] R. Housley, W. Polk, W. Ford, and D. Solo, Internet X.509 Public key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Internet Engineering Task Force RFC 3280 Apr. 2002 [Online]. Available: http://www.rfc-editor.org/rfc/rfc3280.txt

[2] T. P. Hormann, K. Wrona, and S. Holtmanns, "Evaluation of certificate validation mechanisms," *Comput. Commun.* vol. 29, pp. 291–305, Feb. 2006 [Online]. Available: http://portal.acm.org/citation.cfm?id=1646655.1646878

[3] M. Lippert, V. Karatsiolis, A. Wiesmaier, and J. Buchmann, "Lifecycle management of x.509 certificates based on ldap directories," *J. Comput. Secur.* vol. 14, pp. 419–439, Sep. 2006 [Online]. Available: http://portal.acm.org/citation.cfm?id=1239313.1239316

[4] D. Walleck, Y. Li, and S. Xu, "Empirical analysis of certificate revocation lists," in *Proc. 22nd Ann. IFIP WG 11.3 Working Conf. Data and Applications Security*, 2008, pp. 159–174.

[5] C. Ma, N. Hu, and Y. Li, "On the release of crls in public key infrastructure," in *Proc. 15th Conf. USENIX Security Symp.*, Berkeley, CA, 2006, vol. 15.

[6] N. Hu, G. K. Tayi, C. Ma, and Y. Li, "Certificate revocation release policies," *J. Comput. Secur.* vol. 17, pp. 127–157, Apr. 2009 [Online]. Available: http://portal.acm.org/citation.cfm?id=1544133.1544134

[7] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day: Incorporated, 1990.

[8] W. Willinger, V. Paxson, and M. S. Taqqu, "Self-similarity and heavy tails: Structural modeling of network traffic," in *A Practical Guide to Heavy Tails*, R. J. Adler, R. E. Feldman, and M. S. Taqqu, Eds. Cambridge, MA: Birkhauser Boston Inc., 1998, pp. 27–53.

[9] C. W. J. Granger and R. Joyeux, "An introduction to long-memory time series models and fractional differencing," *J. Time Series Anal.*, vol. 1, no. 1, pp. 15–29, 1980.

[10] J. R. M. Hosking, "Fractional differencing," *Biometrika*, vol. 68, no. 1, pp. 165–176, Apr. 1981.

[11] J. Beran, *Statistics for long-memory processes, ser. Monographs on statistics and applied probability*. London, U.K.: Chapman & Hall, Oct. 1994, vol. 61.

[12] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[13] Netcraft, Market Share of Certification Authorities 2009 [Online]. Available: https://ssl.netcraft.com/ssl-sample-report/CMatch/certs

[14] J. G. Proakis, *Digital communications/John G. Proakis*. New York: McGraw-Hill, 1983.

[15] C. Liu, "The essence of the generalized newton binomial theorem," *Commun. Nonlinear Sci. Numerical Simulation*, vol. 15, no. 10, pp. 2766–2768, 2010.

[16] D. Kwiatkowski, P. C. Phillips, and P. Schmidt, Testing the Null Hypothesis of Stationarity Against the Alternative of a Unit Root, Cowles Foundation for Research in Economics, Yale University, May 1991.

[17] A. Montanari, M. S. Taqqu, and V. Teverovsky, "Estimating long-range dependence in the presence of periodicity: An empirical study," *Mathemat. Comput. Modelling*, vol. 29, no. 10–12, pp. 217–228, 1999.

[18] M. J. Cannon, D. B. Percival, D. C. Caccia, G. M. Raymond, and J. B. Bassingthwaighte, "Evaluating scaled windowed variance methods for estimating the hurst coefficient of time series," *Physica A, Statist. Theoret. Phys.*, vol. 241, no. 3–4, pp. 606–626, 1997.

[19] T. Karagiannis, M. Molle, and M. Faloutsos, Understanding the Limitations of Estimation Methods for Long-Range Dependence, UC Riverside, 2006.

[20] R. G. Clegg, A Practical Guide to Measuring the Hurst Parameter, 21st UK Performance Engineering Workshop, School of Computing Science, University of Newcastle, Tech. Rep. Series, CSTR-916, 2006, pp. 43–55.

[21] P. Whittle, "Estimation and information in stationary time series," *Arkiv für Matematik*, vol. 2, pp. 423–434, 1953.

[22] J. Rissanen, *Information and Complexity in Statistical Modeling*, 1st ed. New York: Springer, 2007.

[23] L. de la Cruz, E. Pallarès, J. Alins, and J. Mata, "Self-similar traffic generation using a fractional arima model—Application to vbr video traffic," *J. Brazilian Telecommun. Soc.*, no. 14, p. 1, 1999.

[24] P.-R. Chang and J.-T. Hu, "Optimal nonlinear adaptive prediction and modeling of mpeg video in atm networks using pipelined recurrent neural networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 6, pp. 1087–1100, Aug. 1997.

[25] R. Gruenenfelder, "Stochastic Modelling of the Traffic and its Properties in an ATM Network," Ph.D. thesis, Lausanne, 1991.

[26] G. Wise, A. Traganitis, and J. Thomas, "The effect of a memoryless nonlinearity on the spectrum of a random process," *IEEE Trans. Inf. Theory*, vol. 23, no. 1, pp. 84–89, Jan. 1977.

[27] V. Krishnan, *Probability and Random Processes, ser. Wiley Survival Guides in Engineering and Science*. New York: Wiley-Interscience, 2006.

[28] S. Stein, The q Function and Related Integrals Applied Research Laboratory, Sylvania Electronic Systems, Res. Rep. 467, 1965.

[29] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Ann. Math. Stat.*, vol. 23, pp. 493–507, 1952.

[30] WhichSSL, SSL Market Share, Tech. Rep. 979, 2010 [Online]. Available: http://www.whichssl.com/ssl-market-share.html

[31] M. Naor and K. Nissim, "Certificate revocation and certificate update," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 561–560, Apr. 2000.

[32] D. Cooper, "A model of certificate revocation," in *Proc. 15th Ann. Computer Security Applications Conf.*, 1999, pp. 256–264.

[33] A. Arnes, M. Just, S. J. Knapskog, S. Lloyd, and H. Meijer, "Selecting revocation solutions for PKI," in *Proc. NORDSEC '95*, 1995.

[34] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proc. 3rd ACM Workshop on Security of ad hoc and sensor networks, ser. SASN '05*, 2005, pp. 11–21.

[35] G. F. Marias, K. Papapanagiotou, and P. Georgiadis, "ADOPT. A distributed OCSP for trust establishment in MANETs," in *Proc. 11th Eur. Wireless Conf. 2005—Next Generation Wireless and Mobile Communications and Services (European Wireless)*, Apr. 10–13, 2005, pp. 1–7 [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5755348&isnumber=5755255

[36] A. Wasef and X. Shen, "EDR: Efficient decentralized revocation protocol for vehicular Ad HOC networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 9, pp. 5214–5224, Nov. 2009.

[37] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: Design and architecture," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 100–109, Nov. 2008.

[38] B. Fox and B. LaMacchia, "Certificate revocation: Mechanics and meaning," in *Proc. Int. Conf. Financial Cryptography (FC98)*, Feb. 1998, pp. 158–164.

[39] D. Cooper, "A more efficient use of delta-CRLs," in *Proc. 2000 IEEE Symposium on Security and Privacy. Computer Security Division of NIST*, 2000, pp. 190–202.

[40] M. Ofigsbø, S. Mjølsnes, P. Heegaard, and L. Nilsen, "Reducing the cost of certificate revocation: A case study," in *Proc. Public Key Infrastructures, Services and Applications*, 2010, vol. 6391, pp. 51–66, ser. Lecture Notes in Computer Science.

[41] R. Racine, "Estimating the hurst exponent," Master's thesis, ETH, Zurich, Apr. 2011.

Carlos Gañán was born in Barcelona, Spain, in 1984. He received the B.S. degree in electrical engineering and the M.S. degree in telematics from the Universitat Politècnica de Catalunya (UPC) in 2008 and 2009, respectively. In 2008, he joined the Information Security Group, with the Department of Telematics Engineering at UPC, Barcelona. He is currently pursuing the Ph.D. degree, carrying out research on security for vehicular communications. His academic interests span multimedia communications, network security, and vehicular ad-hoc networks.

Jorge Mata-Díaz received the M.S. degree in telecommunications engineering in 1991, and the Ph.D. degree in 1996, both from the Universitat Politècnica de Catalunya (UPC). The Spanish Association of Telecommunications Engineers rewarded him for his doctoral thesis with the Telefonica Award in Networks and Telecommunications Services. He is currently a research staff member of the Telematics Engineering Department. His research interests include network services, audiovisual appliance, streaming QoS, traffic modeling, and statistical performance analysis.

Jose L. Muñoz received the M.S. degree in telecommunication engineering from the Technical University of Catalonia (UPC) in 1999. In the same year, he joined the AUNA Switching Engineering Department. Since 2000, he works in the Department of Telematics Engineering of the UPC, currently as Associate Professor. In 2003, he received the Ph.D. degree in network security.

Juan Hernández-Serrano was born in Salamanca, Spain, in 1979. He received the M.S. degree in electrical engineering in 2002, and the Ph.D. degree in 2008, both from the Universitat Politècnica de Catalunya (UPC). In 2002, he joined the Information Security Group (ISG) within the Telematics Services Research Group at the Department of Telematics Engineering of the UPC. He currently works as assistant professor at the UPC. His research interests include security for large deployment of sensor networks, autonomous cognitive networks, smart grids, digital forensics, and e-voting.

Oscar Esparza received the M.S. degree in telecommunication engineering from the Technical University of Catalonia (UPC) in 1999. In the same year, he joined the AUNA Switching Engineering Department. Since 2001, he works in the Department of Telematics Engineering of the UPC, currently as Associate Professor. In 2004, he received the Ph.D. degree in mobile agent security and network security.

Juanjo Alins received the M.S. degree in telecommunications engineering in 1994, and the Ph.D. degree in 2004, both from the Polytechnic University of Catalonia, Spain. He is currently a research staff member with the Telematics Services Research Group in the Telematics Engineering Department. His research interests include network services to the home, audiovisual appliance, streaming QoS, secure multimedia transmission, traffic modeling, and statistical performance analysis. He works as a Professor in the Polytechnic University of Catalonia.

COACH: COllaborative certificate stAtus CHecking mechanism for VANETs

Carlos Gañán Jose L. Muñoz Oscar Esparza
Jorge Mata-Díaz Juan Hernández-Serrano Juanjo Alins

Universitat Politècnica de Catalunya (UPC)
{carlos.ganan,jose.munoz,oscar.esparza,jmata,jserrano,juanjo} @entel.upc.edu

Abstract

Vehicular Ad Hoc Networks (VANETs) require mechanisms to authenticate messages, identify valid vehicles, and remove misbehaving vehicles. A Public Key Infrastructure (PKI) can be used to provide these functionalities using digital certificates. However, if a vehicle is no longer trusted, its certificates have to be revoked and this status information has to be made available to other vehicles as soon as possible. In this paper, we propose a collaborative certificate status checking mechanism called COACH to efficiently distribute certificate revocation information in VANETs. In COACH, we embed a hash tree in each standard certificate revocation list (CRL). This dual structure is called *extended-CRL*. A node possessing an *extended-CRL* can respond to certificate status requests without having to send the complete CRL. Instead, the node can send a short response (less than 1Kbyte) that fits in a single UDP message. Obviously, the substructures included in the short responses are authenticated. This means that any node possessing an *extended-CRL* can produce short responses that can be authenticated (including Road Side Units or intermediate vehicles). We also propose an extension to the COACH mechanism called EvCOACH that is more efficient than COACH in scenarios with relatively low revocation rates per CRL validity period. To build EvCOACH, we embed an additional hash chain in the *extended-CRL*. Finally, by conducting a detailed performance evaluation, COACH and EvCOACH are proved to be reliable, efficient, and scalable.

Keywords: VANET, Authentication, Certificate Validation, PKI, CRL, Merkle hash tree.

1. Introduction

In the last years, wireless communications between vehicles have attracted extensive attention for their promise to contribute to a safer, more efficient, and more comfortable driving experience in the foreseeable future. This type of communications has induced the emergence of Vehicular ad hoc networks (VANETs), which consist of mobile nodes capable of communicating with each other (i.e. Vehicle to Vehicle Communication -V2V communication) and with infrastructure (i.e. Vehicle to Infrastructure Communication -V2I communication). To make these communications feasible, vehicles are equipped with *On-Board Units* (OBUs), and fixed communication units called *Road Side Units* (RSUs) are placed along the road. Finally, multihop communication based on IEEE 802.11 is used to facilitate information exchange among network elements that are not in direct communication range [4, 14].

The open-medium nature of these networks makes it necessary to integrate in VANET security mechanisms such as authentication, message integrity, non-repudiation, confidentiality and privacy [35]. The solution envisioned to achieve these functionalities is to use digital certificates provided by a centralized Certification Authority (CA) [11, 33].

In this context, according to the IEEE 1609.2 standard [12], certificates will be used for digitally signing messages and also for encryption (using the ECIES algorithm). Finally, vehicular networks will rely on a Public Key Infrastructure (PKI) to manage certificates. A critical part of the PKI is how to manage certificate revocation. In general, revocation systems for VANETs can be roughly classified as global or local depending on the extent of the revocation mechanism.

- *Local revocation approaches* enable a group of neighboring vehicles to revoke a nearby misbehaving node. In such approaches, revocation is possible without the intervention of external infrastructure at the expense of trusting other vehicles criteria.
- *Global revocation approaches* are based on the existence of centralized infrastructure such as the PKI, which is in charge of managing revocation.

According to the IEEE 1609.2 standard [12], vehicular networks will rely on PKI and Certificate Revocation Lists (CRLs) will be used to distribute the status (revoked or valid) of certificates. CRLs are black lists that enumerate revoked certificates along with the date of revocation and, optionally, the reasons for revocation. CRLs in VANET

are expected to be quite large because this type of network is expected to have many nodes (vehicles) and also because each vehicle will probably have many temporary certificates (also called pseudonyms) to protect the users' privacy. As a result, a VANET CRL might have a size of hundreds of Megabytes [31, 9, 42]. The distribution of such a huge structure within a VANET is a challenging issue and it has attracted the attention of many researchers [32, 17, 35, 9]. A general conclusion about these works is that most of the research efforts have been put into trying to reduce the size of the CRL, either trying to split it or trying to compress it (see Section 2).

In this paper, we take a novel approach because our primary goal is not reducing the CRL size[1] but we aim to design a more efficient way of using the CRL information to distribute certificate status information (CSI) inside the VANET. Our proposal is called COACH (COllaborative certificate stAtus CHecking). COACH is an application-layer mechanism for distributing revocation data. The main idea behind COACH is to embed some little extra information into the CRL such that allows us to create an efficient and secure request/response protocol. For those nodes that just want to obtain status data of some certificates, our protocol replaces downloading a complete CRL. In more detail, we propose a way of efficiently embedding a Merkle hash tree (MHT) [28] within the structure of the standard CRL to generate a so-called *extended-CRL*. To create the *extended-CRL*, we use an extension, which is a standard way of adding extra information to the CRL. Our extension contains all the necessary information to allow any vehicle or VANET infrastructure element that possesses the *extended-CRL* to build the COACH tree, i.e., a hash tree with the CSI of the CRL. Using this COACH tree, any entity possessing the *extended-CRL* can act as repository and efficiently answer to certificate status checking requests of other vehicles or VANET elements. COACH responses are short since in general, their size is less than 1 Kbyte. This allows a COACH response to perfectly fit within a single UDP message. As we will demonstrate by simulation, this makes the distribution of CSI more efficient than distributing complete CRLs (even though they are compressed), reducing the data that have to be transmitted over the VANET. We must stress that a node possessing an *extended-CRL* can act as COACH repository but that a COACH repository is not a TTP. In other words, COACH is offline, which means that no online trusted entity (like a CA) is needed for authenticating the responses produced by COACH repositories.

Finally, we also propose an enhancement of our basic mechanism called EvCOACH (Evergreen-COACH) to improve the performance of COACH in scenarios with relatively few revocations per CRL validity period. Notice that low revocation rates or small CRL validity periods give rise to such scenarios. In these scenarios, it is plausible to have the same revocation information in several consecutive CRLs. In this case, EvCOACH prevents end-entities from downloading a new CRL whose information is already known. To build EvCOACH, we additionally embed a hash chain in the *extended-CRL*. With this structure, now we can extend the validity of a previous CRL by periodically disclosing successive values of the hash chain. As we will show by simulation, EvCOACH overcomes COACH in terms of bandwidth efficiency in scenarios with relatively few revocations per CRL validity period.

The rest of this paper is organized as follows. In Section 2, we present the background related to our mechanism. In Section 3 we describe in depth COACH. Section 4 depicts EvCOACH, the variant of our proposed mechanism. Section 5 provides a security analysis of the proposals. In Section 6, we evaluate the proposed mechanisms. Finally, Section 7 concludes this paper.

2. Background

In this section, first we start describing the existing global revocation proposals for VANET. Then, we provide a brief overview of Merkle Hash Trees (MHT) [27], which is one of the foundations of the proposed certificate validation mechanism.

2.1. Global VANET revocation mechanisms

Global revocation approaches assume the existence of a Trusted Third Party (TTP), which manages the revocation service. The IEEE P1609.2 standard [12] proposes an architecture based on CAs. In this architecture, each vehicle possesses several pseudonyms, which are made publicly available by means of short-lived certificates. However, the revocation mechanism for VANET cannot rely uniquely on the use of short-lived certificates (e.g. as proposed in [19]) because compromised or faulty certificates could still cause damage until the end of their lifetimes.

Raya *et al.* [35] have proposed the use of short-lived certificates that are preloaded in a tamper-proof device (TPD). The TPD is a trusted component that forms part of the OBU. The TPD stores the valid certificates for a vehicle, signs messages, and performs encryption and decryption functions. Raya *et al.* introduced two centralized revocation protocols. The first one is based on the revocation of the TPD, which is necessary when all the certificates of a vehicle are to be revoked. This method assumes the presence of the (on-line) infrastructure to send these messages to the trusted component. To ensure that messages from this OBU are not considered valid once the certificates have been revoked, revocation information must also be distributed via CRLs. The second protocol proposed in [35] is based on the use of compressed CRLs. To compress the CRL, they propose to use Bloom filters. Their method reduces the size of a CRL by using about half the number of bytes to specify the certificate serial number for revocation. Storing CRL information in this manner

[1] Indeed, our proposal can work together with these other approaches that try to reduce the size of the CRL.

compresses the size of the CRL considerably since a fixed-length Bloom filter is distributed instead of distributing 8 to 14 bytes for every certificate that is revoked.

The distribution of CRLs to all vehicles is not trivial. Some authors [33, 32] have proposed the use of regional certification authorities instead of using a single central authority. Papadimitratos *et al.* [34] suggest restricting the scope of the CRL within a region. The authors also propose breaking the Certificate Revocation List (CRL) into different pieces and transmitting these pieces using Fountain or Erasure codes. In this way, a vehicle can reconstruct the CRL after receiving a certain number of pieces. Similarly, in [41], each CA distributes the CRL to the RSUs in its domain through Ethernet. Then, the RSUs broadcast the new CRL to all the vehicles in that domain. In the case RSUs do not completely cover the domain of a CA, V2V communications are used to distribute the CRL to all the vehicles [17]. This mechanism is also used in [7], where it is detailed a public key infrastructure mechanism based on bilinear mapping. Revocation is accomplished through the distribution of CRL that is stored by each user.

Another adaptation of classic public key infrastructure to VANETs is proposed in [3]. This architecture is based on elliptic curves, where each user gets a master key and a master certificate from the CA. Users can then generate their key pairs or certificates using the masker key, the master certificate and their own secret key. In this mechanism the revocation is also centralized. Whenever a key has to be revoked, the CA publishes some data depending on which the nodes have to update their keys. Lin *et al.* in [18] present another centralized revocation mechanism, which minimizes the storage at the CA for later liability establishment, but the revocation is aided by the RSU.

Finally, some proposals in the literature divert from the IEEE P1609.2 standard and use the Online Status Checking Protocol (OCSP)[30]. OCSP is a request/response protocol between clients and responders. An OCSP responder is a trusted intermediate authority for revocation data distribution. Requests may or may not be signed by the client but all the responses must be signed so that clients can ensure that they are communicating with an authorized OCSP responder. In VANET, there is a proposal called ADOPT (Ad-hoc Distributed OCSP for Trust) [25] that provides a revocation service based on OCSP in a decentralized manner. ADOPT uses cached OCSP responses that are distributed and stored on intermediate nodes in the VANET.

2.2. The Merkle Hash Tree

A Merkle hash tree (MHT) [27] is essentially a tree structure that is built with a One Way Hash Function (OWHF). The leaf nodes contain the hash values of the data of interest (`data1`,`data2`,...) and the internal nodes contain the hash values that result from applying the OWHF to the concatenation of the hash values of its children nodes. In this way, a large number of separate data can be tied to a single hash value: the hash at the root node of the tree. MHTs can be used to provide an efficient and highly-scalable way to distribute revocation information, as it is described in [8] for MANETs (Mobile Ad Hoc Networks). A sample MHT is presented in Figure 1. This hash tree is binary because each node has at most two children or equivalently, two sibling nodes are combined to form a parent node in the next level. We will denote these siblings as "left" and "right" and a detailed explanation of how to build the hash tree for COACH is given in Section 3.3.

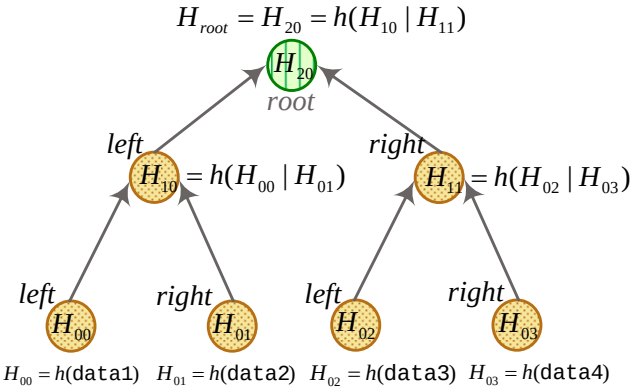


Figure 1: Sample binary Merkle Hash Tree.

3. COACH: Collaborative Certificate Status Checking Based on Merkle Hash Trees

3.1. System Requirements

These are the requirements that a status checking mechanism for VANETs must fulfill:

1. *Reliability*: the certificate status checking service must be available at all times, even if RSUs or mobile repositories fail (or are attacked). Reliability is essential to provide security services, and the status checking mechanism must be resilient to attacks. For instance, a denial of service attack should not cause vehicles to consider revoked certificates as valid.

2. *Scalability*: the cost of validating the status of certificates must be low enough. This is due to VANETs may be very large networks, involving from hundreds of CAs to millions of users conducting billions of transactions.

3. *Bandwidth*: the cost in terms of bandwidth consumption should be small. Status checking protocols must minimize the amount of data that has to be exchanged. As a consequence, users will experience shorter validation delays, less burden will be placed on the network infrastructure, and the cost of providing the revocation service will be reduced.

4. *Performance*: the entire system must be efficient, without requiring excessive computational complexity or network overhead for any participant.

5. *Auditability*: while the requirements for the actual revocation decision itself vary widely between implementations, the cryptographic operations should be auditable.

6. *Manageability*: it must be possible to operate the system in a secure manner. Whenever possible, critical keys should be stored in a tamper-proof device where they are less probable to be compromised

7. *Practicality*: the system must be easy to integrate into existing applications and future infrastructure.

8. *Authentication*: each entity in the network should have an authentic identity. Any received message should first be authenticated before performing further processing.

9. *Unforgeability*: no entity should be capable of generating fake revocation information.

10. *Resistance to replay attacks*: it should be impossible to cheat a vehicle by recording and replaying an old revocation message, for instance trying to persuade this vehicle that a certain certificate is valid when in fact it has been revoked.

3.2. System Architecture

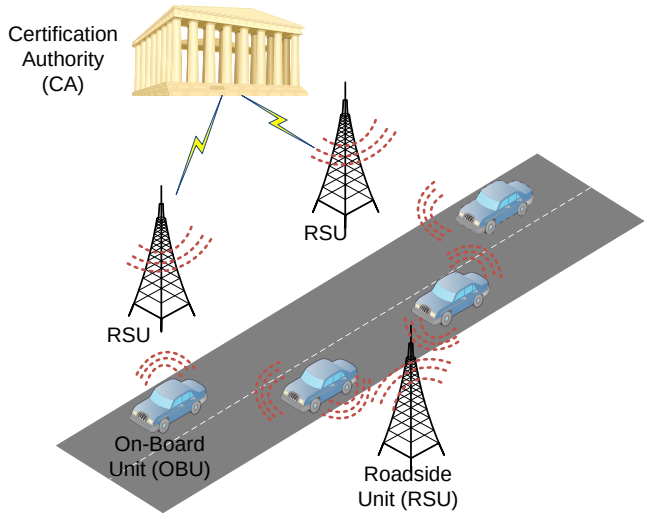


Figure 2: System Architecture.

The system architecture is an adaptation of a PKI system to the vehicular environment. We present a hierarchical architecture (see Fig. 2) which consists of three levels: the certification authority (CA) is located at level 1, as it is the top of the system. The road side units (RSUs) are located at level 2. Finally, the on-board units (OBUs) are located at the bottom of the hierarchy.

The main tasks of each entity are:

1. The CA is responsible for generating the set of certificates that are stored in each OBU. It is also responsible for managing the revocation information and making it accessible to the rest of the entities. By definition of TTP, the CA should be considered fully trusted by all the network entities, so it should be assumed that it cannot be compromised by any attacker. In fact, in our proposal the CA is the only trusted entity within the network.

2. RSUs are fixed entities that are fully controlled by the CA. They can access the CA anytime because they are located in the infrastructure-side, which does not suffer from disconnections. If the CA considers that an RSU has been compromised, the CA can revoke it.

3. OBUs are in charge of storing all the certificates that a vehicle possesses. An OBU has abundant resources in computation and storage and allows any vehicle to communicate with the infrastructure and with any other vehicle in its neighborhood.

3.3. COACH Tree

In this section, we introduce the data structure that COACH uses to handle the revocation service. In this sense, we define the COACH tree as a particular case of Merkle Hash Tree explained in section 2. The COACH tree is a binary hash tree where each node represents a revoked certificate.

We denote by $N_{i,j}$ the nodes within the COACH tree, where $i, j \in \{0, 1, 2...\}$ represent respectively the i-th level and the j-th node in the i-th level. We denote by $H_{i,j}$ the cryptographic (hash) value stored by node $N_{i,j}$.

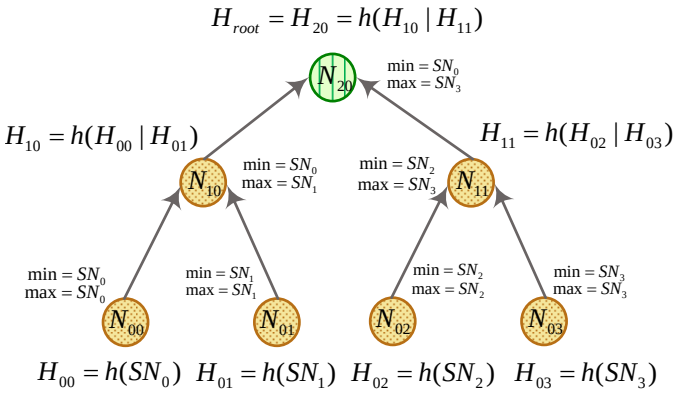


Figure 3: Sample COACH Tree.

Nodes at level 0 are called "*leaves*" and they represent the data stored in the tree. In the case of revocation, leaves represent the set Φ of certificates that are revoked at a given instant t,

$$\Phi_t = \{SN_0, SN_1, \ldots, SN_j, \ldots, SN_n\}\,. \qquad (1)$$

If SN_j is the certificate serial number stored by leaf $N_{0,j}$, then $H_{0,j}$ is computed as:

$$H_{0,j} = h(SN_j)\,, \tag{2}$$

where $h()$ corresponds to the OWHF function.

Leaves are ordered in the following way: leaves on the left represent smaller serial numbers than leaves on the right. Each node also stores the minimum and maximum serial number of its children. If a leaf has no children, then it uses its own serial number for the maximum and minimum values.

To build the COACH tree, two adjacent nodes at a given level i ($N_{i,j}$, $N_{i,j+1}$) are combined into one node in the upper level, which we denote by $N_{i+1,k}$. Then, $H_{i+1,k}$ is obtained by applying h to the concatenation of the two cryptographic values:

$$H_{i+1,k} = h(H_{i,j}|\, H_{i,j+1}). \tag{3}$$

At the top level there is only one node called the "root". H_{root} is a digest for all the data stored in the COACH tree. Figure 3 shows a sample COACH tree.

Definition 1. Let the $\mathcal{D}$igest be the concatenation of the certification authority distinguished number DN_{CA}, the root hash H_{root} and the validity period of the CRL. Once created, the $\mathcal{D}$igest is signed by the CA.

$$\mathcal{D}igest = \{DN_{CA}, H_{root}, ValidityPeriod\}_{SIG_{CA}}. \tag{4}$$

Definition 2. Let the $\mathcal{P}\text{ath}_{SN_j}$ be the set of cryptographic values necessary to compute H_{root} from the leaf SN_j.

Remark 1. Note that the $\mathcal{D}$igest is trusted because it is signed by the CA and it is unique within the tree. Meanwhile, $\mathcal{P}$aths are different for each leaf.

Claim. An End Entity can verify whether $SN_j \in \Phi$ if the MHT provides a response with the proper $\mathcal{P}\text{ath}_{SN_j}$ and the $\mathcal{D}$igest of the MHT.

Example. Let us suppose that a certain user wants to find out whether SN_1 belongs to the sample COACH tree of Figure 1. Then,

$\mathcal{P}\text{ath}_{SN_1} = \{H_{0,0}, H_{1,1}\}$,

$\mathcal{D}igest = \{DN_{CA}, H_{2,0}, ValidityPeriod\}_{SIG_{CA}}$.

The response verification consists in checking that $H_{2,0}$ computed from the $\mathcal{P}\text{ath}_{SN_1}$, $h(h(h(SN_1)|H_{0,0})|H_{1,1})$ matches the cryptographic value $H_{2,0} = H_{root}$ included in the $\mathcal{D}$igest:

$$H_{root} = H_{2,0} = h(h(h(SN_1)|H_{0,0})|H_{1,1})\,. \tag{5}$$

Remark 2. Note that the COACH tree can be built by a Trusted Third Party (e.g. a CA) and freely distributed to untrusted repositories. The COACH tree cannot be forged, that is, any change in the tree made by a non-TTP will be detected. This is due to any modification in the COACH tree (for instance, the addition or deletion of a leaf node) causes H_{root} to change. As H_{root} is included in the $\mathcal{D}$igest, which is signed by the CA, this modification will cause the signature to be invalid. To perform a successful attack, the attacker would need to find a pre-image of an OWHF, which is computationally infeasible by definition.

3.4. COACH modus operandi

COACH consists in three stages. During the first stage of *System Initialization*, the CA creates the "extended-CRL", that is, a CRL in which a signed extension is appended. This extension will allow third non-trusted parties to answer status checking requests in an off-line way when required. Once this *extended-CRL* has been constructed, it is distributed to the RSUs. In the second stage of *Repository Creation*, a non-trusted entity (i.e. a RSU or a vehicle) gets the *extended-CRL* and becomes a certificate status checking repository for other VANET entities. Finally, in the third stage of *Certificate Status Checking*, vehicles can use an efficient protocol to obtain the CSI from an available VANET repository.

The main advantage of COACH over CRL is that the entire CRL is not needed for verifying a specific certificate and that a user may hold a succinct proof of the validity of her certificate. With COACH only the repositories must store the whole CRL, while users only need to query for the status of a particular certificate. Thus, the bandwidth that non-repository vehicles need to check the status of a certificate is much lower with COACH than with CRL. On the other hand, COACH allows any vehicle to act as repository. This is unfeasible with standard CRLs as the time and the bandwidth necessary to transmit a CRL would not be available with V2V communications. In this way, COACH becomes an offline certificate status validation mechanism as it does not need trusted responders to operate. Henceforward, we give a more detailed description of COACH modus operandi.

3.4.1. System Initialization

In this first stage, the CA creates the *extended-CRL* and delivers it to the RSUs. An *extended-CRL* is basically a standard CRL with an appended extension. This extension can be used by non-trusted entities (RSUs and vehicles inside the VANET) to act as repositories and answer to certificate status requests. All the tasks of this system initialization are performed in the CA locally (see Figure 4).

These are the steps that the CA must carry out:

1. The CA creates a *tbs-CRL*[2] (to be signed CRL), that is, a list that contains the serial numbers of the certificates that have been revoked (along with the date of revocation), the identity of the CA, some timestamps to establish the validity period, etc.

[2]The CA can generate the standard CRL by simply signing this tbs-CRL.

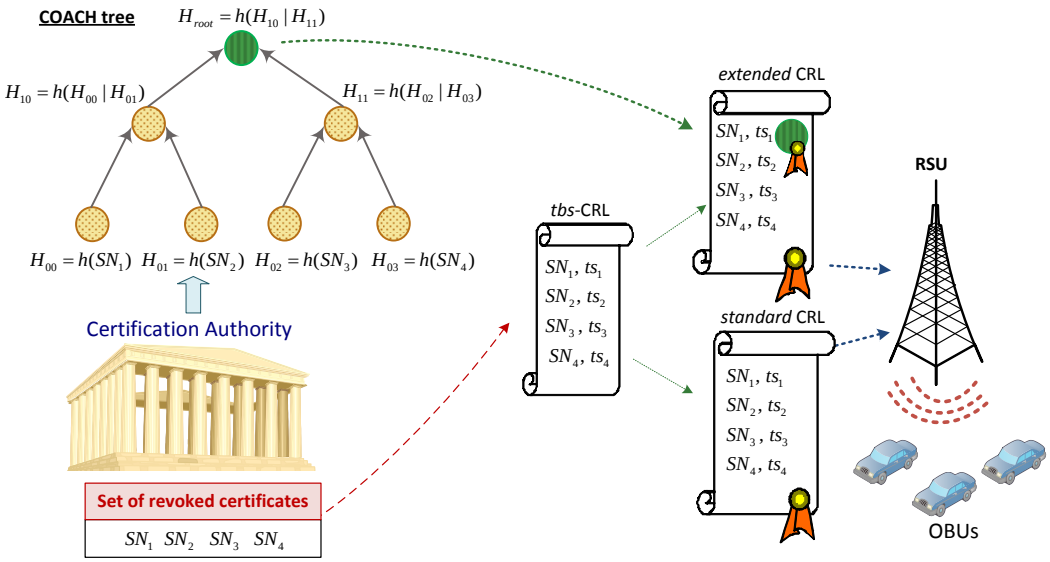


Figure 4: COACH System Initialization Example.

2. The CA creates the COACH tree, that is, a MHT that is constructed by using the serial numbers within the previous *tbs-CRL* as leaves of the tree. The COACH tree has been designed as a binary tree, and it should be constructed following the methodology explained in Section 3.3. The order of these leaves within the COACH tree is the same than the order of appearance in the tbs-CRL. We assume the *tbs-CRL* to be a sequence of revoked certificate ordered by serial number. Therefore, the leaves of the COACH tree also follow the same order, that is, the bottom left leaf stores the revoked certificate with lowest serial number. Note that if the COACH tree is formed by an odd number n of leaves, there is a leaf $N_{0,n-1}$ that does not have a pair. Then the single node is simply carried forward to the upper level by hashing its $H_{0,n-1}$ value. We proceed in the same way if any i-th level is formed by an odd number n of nodes. Once created the MHT, the CA obtains the root hash.

3. The CA calculates the extension, which consists basically of the $\mathcal{D}$igest. Just recall that this $\mathcal{D}$igest was calculated in Equation 4 as the concatenation of the certification authority distinguished number, the root hash and the validity period of the CSI, and after that signed by the CA. Obviously, the distinguished number and the validity period should be the same than the ones contained in the tbs-CRL. In fact, the COACH tree is just a different way of representing the CSI, but the hash tree will be valid during the same time and will provide the same information than the CRL. Once calculated, this $\mathcal{D}$igest is appended to the tbs-CRL, generating the *tbs-extended-CRL*.

4. The CA signs the tbs-extended-CRL, generating the *extended-CRL*. Notice that this second overall signature not only authenticates all the CSI, but also binds this CSI to the $\mathcal{D}$igest. The *extended-CRL* is only slightly larger than the standard CRL, as we will show later in Section 6.

5. Finally, the CA distributes copies of the *extended-CRL* to the designated RSUs, which will act as the typical PKI repositories, in the same manner as they would do with a standard CRL.

After this first stage of System Initialization, the RSUs have a copy of the *extended-CRL*, which contains exactly the same CSI than a standard CRL and it is valid for the same time. The advantage of an *extended-CRL* is that any non-trusted entity in possession of it can generate again the COACH tree locally, and obtain the root hash. As the *extended-CRL* also includes the $\mathcal{D}$igest, which is signed by the CA, this entity has an authenticated version of the COACH tree and can answer to CSI requests in an off-line way.

3.4.2. Repositories creation

In this stage, RSUs become new repositories of the VANET. Vehicles can also become mobile repositories allowing to distribute the extended-CRL information in areas with poor coverage. To become a repository an entity must follow the next steps:

1. The entity obtains the *extended-CRL* either from the CA or from another entity that has an up-to-date copy of the *extended-CRL* in its cache. Notice that the CA uses a secure wireline to communicate with the RSUs, while the RSUs use a wireless link to communicate with the vehicles.

2. Once the *extended-CRL* has been downloaded, the entity verifies that the signature of the *extended-CRL*

is valid and corresponds to the CA. If so, the entity generates locally the COACH tree using the serial numbers within the *extended-CRL* and following the same algorithm than the CA (as explained in Section 3.3). The root hash of the tree created from the *extended-CRL* entries must match the signed root value contained in the $\mathcal{D}$igest.

3. At this moment, the entity can respond to any status checking request from any vehicle until the *extended-CRL* expires.

3.4.3. Certificate status checking

After the second stage, RSUs and some vehicles will be able to act as repositories. The last stage of the mechanism consists in providing the certificate status information to any vehicle that needs to validate the status of a certificate. Firstly, a vehicle that needs to check the status of a certificate must locate a valid repository. To do so, the vehicle uses a Service Discovery Protocol (SDP) to find a RSU or a vehicle that is acting as repository. A secure service discovery and communication protocol is mandatory in order to prevent from many attacks and malicious processes in VANETs. We assume that the VANET is using an efficient service discovery protocol that guarantees a secure discovery and communication in the vehicular system while maintaining the network scalability and a low communication delay. There have recently appeared some new protocols for ad hoc environments that intend to provide these features [29, 15, 2, 1]. For instance, authors in [1] design an advertisement and discovery mechanism for VANETs which permits vehicles to discover nearby RSUs and their services securely and preserving their privacy. Any of these mechanism could work with COACH to facilitate the discovery of the repositories.

Once the repository has been located, the vehicles start the status checking protocol. The protocol for status information exchange is based on the hash tree structure and it allows checking the integrity of a single *extended-CRL* entry with only some hash material plus the $\mathcal{D}$igest (included in the extension). On the one hand, this is much more efficient than broadcasting the entire *extended-CRL*. On the other hand, the mechanism is fully offline (the only trusted authority is the CA), which is a very good feature because sometimes it may be impossible for vehicles to reach the CA due to lack of coverage.

Hence, a vehicle that needs to check the status of a certificate must follow the next steps:

1. The vehicle uses a service discovery protocol to find either a RSU or a mobile repository inside its coverage range for status checking.

2. The vehicle sends the serial number of the certificate that is going to be verified to the repository. The repository searches the target certificate in the hash tree. In the case the certificate is found, the repository sends the $\mathcal{P}$ath, i.e., the hash values of the nodes of the tree which are needed to calculate the signed root.

3. The vehicle verifies that the H_{root} calculated from the $\mathcal{P}$ath matches the H_{root} contained in the $\mathcal{D}$igest.

Notice that as the H_{root} is signed by the CA, it is just as impractical to create falsified values of the $\mathcal{P}$ath as it is to break a strong hash function. In case the certificate is not revoked, the repository sends the adjacent leaves to the requested certificate. To this respect, the repository has to prove that a certain certificate (SN_{target}) does not belong to the set of revoked certificates (Φ). To prove that $SN_{target} \notin \Phi$, as the leaves are ordered, it is enough to demonstrate the existence of two leaves, a minor adjacent (SN_{minor}) and a major adjacent (SN_{major}) that fulfill:

1. $SN_{major} \in \Phi$.
2. $SN_{minor} \in \Phi$.
3. $SN_{minor} < SN_{target} < SN_{major}$.
4. SN_{minor} and SN_{major} are adjacent nodes.

In any case, the data that the repository needs to send to a node to perform the status checking can be placed in a single UDP datagram using 802.11p link-layer.

It is worth noting that under low-equipped vehicle scenarios, communication disruptions may occur frequently due to high mobility, network congestion, or potential attacks. In such scenarios, requesting nodes that do not get an answer from a local repository have to assume that either its request/respond has not arrived (e.g due to possible interferences), the repository has been compromised or a malicious node is acting as prankster. Note that in any case, the requesting node assumes the absence of the repository because the physical layer in VANETS (based on the Dedicated Short Range Communication (DSRC) protocol [14]) defines a control channel where every node broadcasts a beacon that provides trajectory and other information about the vehicle. Thus, any node is aware of its (legitimate) neighbors at any instant and which are acting as mobile repositories. At this point, reaching a local repository becomes a routing problem which has been well studied in the literature (e.g. [13, 6]).

4. Evergreen COACH (EvCOACH)

In this section, we present a variant of the COACH mechanism specially designed to enhance performance when the revocation rate is low. As explained in the previous section, COACH computes a new hash tree each time the *extended-CRL* expires. However, it could be the case where there have been no revocations during the lifetime of the previous *extended-CRL*. That is to say, the list of revoked certificates has not changed during successive update periods, and the only parameter that has changed is the validity period of the *extended-CRL*. Thus the set of revoked

certificates at $t_0 = thisUpdate$ is the same as the set of revoked certificates at $t_1 = nextUpdate$, i.e., $\Phi_{t_0} = \Phi_{t_1}$. In this context, the *extended-CRL* has a lifetime equal to $nextUpdate - thisUpdate$.

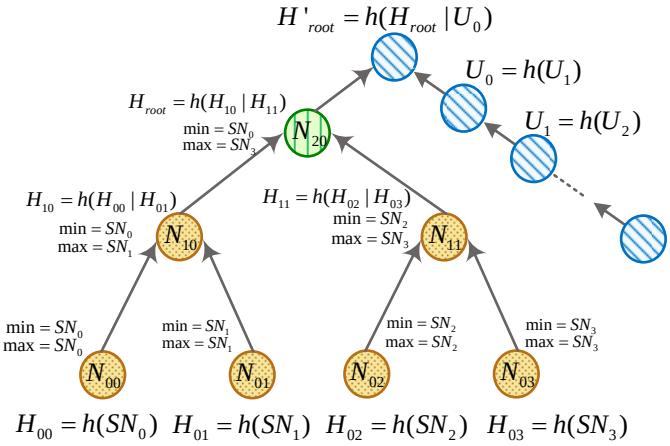


Figure 5: Example of EvCOACH tree.

In this case, vehicles that already have downloaded the whole *extended-CRL* do not obtain new information by downloading a new CRL that has no new revoked certificates. Thus, it would be highly inefficient to publicize a whole new *extended-CRL* when only the timestamps are different from the previous *extended-CRL*. For this reason, we propose a new mechanism that reutilizes the COACH tree calculated previously, minimizing the amount of information that has to be transmitted in such situation. For this purpose, we have designed the EvCOACH tree, which is a COACH tree that is constructed by adding a new branch. This new branch is calculated by hashing a new secret nonce generated by the CA (see Figure 5). In this context, the new EvCOACH tree can be considered perennial as it can live for more than one *extended-CRL* lifetime. This "perennial" property means that the validity of the hash tree can be reassured as many times as the length of the hash chain.

Before describing EvCOACH, we introduce some new definitions.

Definition 3. Let *primaryUpdateValue* (U_d) be a secret nonce that the CA generates.

Definition 4. Let *maximumUpdateValue* (U_0) be the parameter that is concatenated with the old hash root to compute the new root of the EvCOACH tree.

Definition 5. Let *currentUpdateIndex* (i) be the number of time-periods (Δt) elapsed since the last *extended-CRL* publication.

Definition 6. Let *maximumUpdateIndex* (d) be the maximum number of time-periods that the *extended-CRL* can be revalidated.

Next equation shows the relations among these definitions. Note that each value U_i can be calculated applying a hash function h to the previous value, and the first value of the hash chain is the secret nonce U_d.

$$U_d \xrightarrow{h} U_{d-1} \xrightarrow{h} \ldots \xrightarrow{h} U_i \xrightarrow{h} U_2 \xrightarrow{h} U_1 \xrightarrow{h} U_0$$

Definition 7. Let *validityInterval* (Δt) be the additional amount of time during which an invariant EvCOACH tree is still valid.

The aim of this new mechanism is to minimize the amount of information that has to be sent when there are no new revoked certificates with respect to the previous *extended-CRL* publication. For that purpose, the mechanism revalidates the previous *extended-CRL* during a certain amount of time (Δt) without having to recalculate the EvCOACH tree again. Note that the duration of Δt is critical to the performance of the mechanism. The overestimation of Δt could allow to operate with certificates even though they have been revoked. On the contrary, the underestimation of Δt could lead to over-issuing CRLs. We give some hints on how to estimate Δt accurately in Section 4.4. We now give details for the operations of the three parties in the system (CA, RSUs and vehicles) when using EvCOACH.

Note that EvCOACH is just an extension of COACH that allows reassuring the validity of the MHT when there are no new revocations. This is achieved at expenses of an additional communication overhead and computational cost. However, the benefits of not updating the *extended-CRL* overcome these costs.

4.1. CA operations

- **System Initialization**:

  1. *Creation of the EvCOACH tree*: the CA generates one part of the EvCOACH tree by using the set of initially revoked certificates, as explained in Section 3.3. Notice that at the top level of the COACH tree, there is only one node whose value is H_{root}. In fact, this part of the EvCOACH tree has been constructed in the same way than a COACH tree (see Section 3.4.1). The CA also has to generate the other part of the EvCOACH tree, the new branch. To do so, the CA generates a nonce (U_d) that must be kept secret during the whole validity period of the hash tree. Then, it calculates U_0 applying d times the hash function to U_d:

     $$U_0 = h^d(U_d).$$

     Then, the EvCOACH tree is finally the union of both parts, the COACH tree and the new branch, as it is shown in Figure 5. The new EvCOACH root hash can be calculated as:

     $$H'_{root} = h(H_{root}|U_0).$$

2. *Creation of the $\mathcal{D}igest$*: the way in which EvCOACH calculates the $\mathcal{D}$igest is slightly different from COACH, because it includes the first node of the new branch. First, the CA estimates the amount of time (Δt) during which the CRL is not expected to change. That is to say, it estimates the amount of time Δt during which no new certificates are expected to be revoked. Finally, the EvCOACH $\mathcal{D}$igest is calculated as:

$$\mathcal{D}igest = \{DN_{CA}, H'_{root}, U_0, \Delta t\}_{SIG_{CA}}.$$

Again, this $\mathcal{D}$igest is included within the *extended-CRL*, as it was in COACH.

3. *Distribution of the revocation material*: Finally, the CA sends to the RSUs the *extended-CRL*, which contains the (sorted) list of revoked certificates serial numbers along with the EvCOACH $\mathcal{D}$igest.

- **Revalidation of the revocation material**: Once the *extended-CRL* expires, the CA must check whether it can re-use the revocation material already distributed. To that end, the CA has to check if there are new revoked certificates. If there are new certificates the CA must initialize the system again, recalculating the EvCOACH tree (using a new $U_{d'}$) and the $\mathcal{D}$igest, and distributing all the revocation material. On the contrary, if the list of revoked certificates is the same, so that:

$$\Phi_{t_0} = \Phi_{t_1} = \Phi_{t_1+i\Delta t},\ i \geq 0,$$

then, the CA can revalidate the revocation material performing the following steps (just remember that $t_0 = thisUpdate$ and $t_1 = nextUpdate$):

1. *Calculation of the new U_i*: The CA calculates the new value U_i where i is the number of time-periods (Δt) elapsed since the first CRL publication:

$$U_i = h^{d-i}(U_d).$$

Note that at a given instant t_i the CA is the only one capable of calculating the U_i. However, any node that receives a nonce U_i can check its authenticity by hashing i times U_0.

2. *Distribution of the freshest U_i*: The corresponding U_i is distributed to the RSUs and mobile repositories. Notice that the size of the U_i is lower than the size of the $\mathcal{D}$igest distributed in COACH and much lower than the CRL. In this sense, the CA revalidates the information of the EvCOACH tree just by issuing a nonce.

4.2. RSU operations

- **Retrieving the revocation material**: At CRL update instants, RSUs obtain the *extended-CRL* from the CA through a secure wire-line. If instead of an update there is a revalidation of the CRL, the RSUs just download the corresponding U_i.

- **Responding to vehicles requests**: In order to be able to respond to the users' certificate status queries, RSUs must manage the EvCOACH tree. This hash tree can be created directly from the *extended-CRL*, or revalidated by checking the validity of the corresponding U_i. When a user's request arrives at the RSU, it has to process whether it is a request for the whole CRL, or just a certificate status checking. In the former case, the RSU has to forward the *extended-CRL* that it has previously downloaded from the CA. In the latter case, it just has to calculate the $\mathcal{P}$ath of the certificate to prove that a certificate is revoked, or calculate SN_{minor} and SN_{major} to prove the contrary.

4.3. Vehicle operations

Regarding the operations that a vehicle can perform, we can distinguish two types of vehicles, those who want to contribute to the revocation mechanism and become repositories, and those who just want to check the status of the certificates. Repository vehicles can perform the same actions as normal vehicles but, in addition, they can also perform RSU operations. It must be noted that when repository vehicles perform RSU operations they do not communicate with the CA but with the RSUs in range.

- **Checking certificate status**: When a vehicle needs to communicate with another entity, firstly it must check the validity of the entity's certificate. For that purpose, if there is a reachable RSU, a non-repository vehicle queries it for the $\mathcal{D}$igest or the corresponding U_i and the $\mathcal{P}$ath of the certificate. Then, it checks the validity of the $\mathcal{D}$igest. To do so, they verify that the corresponding U_i is valid as this value can only be generated by the CA due to the properties of a OWHF. Therefore, to check validity of U_i at an instant $t' \in [nextUpdate + (i-1)\Delta t, nextUpdate + i\Delta t]$ a vehicle has to verify that the following equality is satisfied:

$$U_0 = h^i(U_i) \ \ with \ \ i \leq d.$$

Once, it has been checked the validity of the U_i, the vehicle has to check whether the $\mathcal{P}$ath is valid or not. To do so, the vehicle verifies that the H'_{root} calculated from the $\mathcal{P}$ath matches the H'_{root} contained in the $\mathcal{D}$igest.

- **Responding to vehicle requests**: Repository vehicles can also respond to other vehicle requests, following the same steps than a RSU.

Note that the main differences between COACH and EvCOACH are: 1) EvCOACH has an additional branch in the MHT that allows the revalidation of the MHT, 2) the $\mathcal{D}$igest in EvCOACH contains one extra parameter to allow the verification of the revalidation. With these slight modifications, EvCOACH highly improves its performance in those VANET where the CRL does not grow between consecutive updates.

4.4. Estimating Δt

In order to estimate the duration of the validity of the certificate status information, the CA needs to balance the liability cost of not releasing a new CRL on time and the costs of releasing CRLs too often. We show a technique to calculate the revalidation interval (Δt) of the EvCOACH tree to minimize that cost.

Authors in [21] carry out an analytical study dealing with optimization of the release of CRLs. Although the data that they collected to perform the analysis do not belong to a CA deployed in a VANET, we can extrapolate some of their results to our scenario. According to [21] and [40], the probability density function (PDF) of the revocation process fits an exponential distribution:

$$f(x) = \frac{1}{\mu} e^{-\frac{x}{\mu}}.$$

The μ parameter represents the mean lifetime of a revoked certificate, and according to those papers it equals 30 days approximately. With that information and the lifetime of the operative certificates in the VANET, the CA can estimate the probability of having a certificate revoked, and thus estimate Δt.

5. Security Analysis

In this section we provide a brief security analysis of COACH and EvCOACH. As a general remark, note that it is relatively straightforward to see that these structures are secure because they are essentially a type of Merkle hash tree, which has been proven to be secure.

In first place, assuming that h is a one-way collision-resistant hash function and in second place, assuming that the signature scheme (i.e. ECDSA) is secure, neither a proof of revocation nor a proof of validity can be forged. Thus, any integrity violation of a node in a tree can be detected thanks to the properties of the hash function h, which is one-way and collision-resistant. Next, we show that COACH and EvCOACH are secure against fabrication attacks, denial of service attacks and simple replay attacks.

5.1. Fabrication Attacks

In COACH, a $\mathcal{P}$ath is legitimate only if it contains the signed proof (i.e. the root of the merkle hash tree) and it has not expired. The one-way property of the hash function prevents any adversary from fabricating a $\mathcal{P}$ath that yields the correct value for the COACH-tree root. Note that the same applies to EvCOACH as the proof included in the $\mathcal{P}$ath is calculated by hashing a random value (U_i) and the root of the COACH-tree. As U_i is calculated by hashing i times a nonce only known by the CA, no entity can forge this value. Therefore, both mechanisms are resistant against fabrication attacks.

5.2. Replay Attacks

COACH and EvCOACH are also safe against replay attacks in which a malicious entity is replying with an old $\mathcal{P}$ath. This is because the $\mathcal{D}$igest includes a validity period, so any entity can check if a given $\mathcal{P}$ath that has associated a specific the $\mathcal{D}$igest is outdated. This validity period cannot be forged or altered thanks to the one-way property of the hash function. Therefore, expired $\mathcal{P}$aths can be dismissed, effectively avoiding this kind of replay attacks.

5.3. Denial of Service Attacks

Finally, malicious and selfish behaviors should be considered when providing revocation data in a VANET. For example, a malicious vehicle in a VANET may start flooding the network with requests. As a result, mobile repositories and RSUs receiving these bogus requests will calculate the corresponding $\mathcal{P}$ath to build the responses. This process consumes resources and might potentially deny the service to other vehicles. On the other hand, some VANET nodes may exhibit a selfish behavior. In terms of COACH and EvCOACH, selfish vehicles may, for instance, decide not becoming a repository despite having enough resources for doing so. These behaviors on a large scale VANET can result in network congestion and potential denial of service. To prevent and deal with these attacks, a trust establishment framework could be used to support COACH's operation (e.g. [24]). This type of frameworks, incorporate self-evidences, recommendations, subjective judgment and historical evidences to continuously evaluate the trust level of vehicles, which can benefit COACH (and EvCOACH) in terms of availability and robustness.

6. Performance Evaluation

6.1. Analytical Evaluation

In this section, we analytically compare the performance of COACH with other mechanisms in terms of overhead and computational cost.

6.1.1. Communication Overhead

Let's start estimating the size of a CRL in a vehicular environment. The size of a CRL is proportional to the number of revoked certificates. Let N_{veh} be the total number of vehicles in the region that the CRL needs to cover, ρ the average percentage of certificates revoked, L_f the lifetime (or validity period) of a certificate, and $\bar{s}$

the mean number of pseudonyms of a vehicle. Additionally, let N_{rev} be the number of non-expired certificates that were revoked, i.e., the number of certificates that the CRL contains. According to [40], the probability density function of certificate revocation approximately follows an exponential distribution:

$$f(t; L_f) = L_f \cdot e^{-t \cdot L_f}, \forall t \geq 0$$

Therefore, if certificate is revoked at instant t of its lifetime, it stays in the CRL for $L_f - t$. Thus, the expected time a revoked certificate stays in the CRL can be estimated as:

$$E(L_f - t) = E(L_f) - E(t) = L_f - \frac{1}{L_f} = \frac{L_f^2 - 1}{L_f} \simeq L_f$$

Then, we can estimate the mean number of revoked certificates in a CRL as:

$$\overline{N_{rev}} = N_{veh} \cdot \rho \cdot \bar{s} \cdot L_f.$$

Finally, we estimate the size of a CRL in a VANET. The CRL contains some header information and a CRL entry for each certificate on the list. CRL entries will have varying sizes, but according to 1609.2 standard [12], 14 bytes per entry is a realistic figure, i.e, $s_e = 14$ bytes. The size of the CRL header is negligible compared to the total size of the CRL. According to NIST statistics [5], 10% of the certificates need to be revoked during a year, i.e., $\rho = 0.1$. Recall that in a VANET, each vehicle owes not only an identity certificate, but also several pseudonyms. The number of pseudonyms may vary depending on the degree of privacy and anonymity that it must be guaranteed. According to Raya, Papadimitratos, and Hubaux in [10] the OBU must store enough pseudonyms to change pseudonyms about every minute while driving. This equates to about 43,800 pseudonyms per year for an average of two hours of driving per day. Haas, Hu, and Laberteaux in [32] recommend changing pseudonyms every 10 minutes, and driving 15 hours per week. This equates to 4,660 pseudonyms per year, but they recommend storing five years of pseudonyms for a total of about 25,000 pseudonyms per OBU. Therefore, we set $\bar{s} = 25,000$. Regarding to the certificate lifetime, according to [40], it ranges from 26 to 37 days. In this manner, we set the lifetime to 1 month. Therefore, the expected CRL size is:

$$CRL_{size} = \overline{N_{rev}} \cdot s_e = N_{veh} \cdot \rho \cdot \bar{s} \cdot L_f \cdot s_e$$

Assuming that a regional certification authority could manage around 50,000 vehicles, the expected CRL size is $CRL_{size} \simeq 145$ Mbytes. On the other hand, the response size of COACH is much smaller than a CRL as it consists only of the $\mathcal{D}$igest and the $\mathcal{P}$ath for a given certificate. Using the SHA-1 algorithm (hash size of 160 bits), and ECDSA-256 the size of the response of COACH for 10,000,000 revoked certificates (including pseudonyms) is of approximately 710 bytes. In the same way, EvCOACH has the same response length of standard COACH but adding one nonce to the response (typically 15 bytes).

Mechanism	Request size	Response size
CRL	73 bytes	145 Mbytes
COACH	73 bytes	710 bytes
EvCOACH	73 bytes	725 bytes
ADOPT	66 bytes	586 bytes

Table 1: Comparison of the overhead introduced by COACH and other certificate validation mechanisms.

Table 1 shows the size of the response and the request for the different certificate validation mechanisms. Note that the request size is very similar for all the mechanisms. However, the size of the response varies significantly, e.g., COACH and EvCOACH response sizes are six orders of magnitude smaller than conventional CRL. Figure 6 shows the size of the response for CRL, ADOPT [25] and COACH depending on the number of revoked vehicles in the network. While ADOPT response size is constant, the size of the response when using CRL or COACH increments with the number of revoked certificates. As you can observe, the CRL size grows linearly with the number of revoked certificates, while COACH and EvCOACH response sizes describe a logarithmic growth.

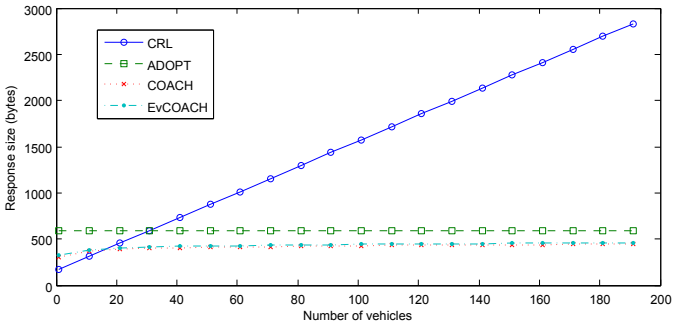


Figure 6: Response size vs number of vehicles.

Notice that downloading a CRL is more efficient than performing COACH requests/responses only when the vehicle makes more than 200,000 requests (see Figure 7), which is quite unfeasible in a typical VANET environment where a normal vehicle is not expected to establish contact with such amount of vehicles. Regarding ADOPT, its response size is slightly smaller than in COACH, but it lacks the benefits that COACH provides to operate during disconnections. Essentially, ADOPT is a caching mechanism for OCSP responses. Like COACH responses, OCSP responses are much smaller than VANET CRLs and thus, under a low request rate ADOPT can feasibly provide timely revocation status information without burdening the network. The main problem with ADOPT is related

to how to locate cached responses. Vehicles have to broadcast their queries in the hope of finding another vehicle that has previously asked for the desired CSI and that has this cached data. This broadcast incurs a significant overhead in a VANET, where vehicles have to validate several certificates per second. Moreover, even if a vehicle is able to locate an intermediate node capable of serving the desired CSI, this CSI could be outdated. Therefore, ADOPT cannot assure that a response has been previously cached by a neighbor vehicle in such dynamic scenarios, in other words, it is quite probable that vehicles will end up querying an OCSP responder or the CA for the CSI. In vehicular scenarios the number of cached responses could be huge, and therefore, also a huge storage capacity is required in the vehicle. In addition, ADOPT does not guarantee that a vehicle obtains the status of a given certificate when needed. So, ADOPT has smaller responses, but it does not provide as fresh information as COACH and it forces VANET nodes to store a large amount of CSI data. Finally, ADOPT makes the network more vulnerable than when using COACH because more trusted connections with OCSP responders or the CA are needed.

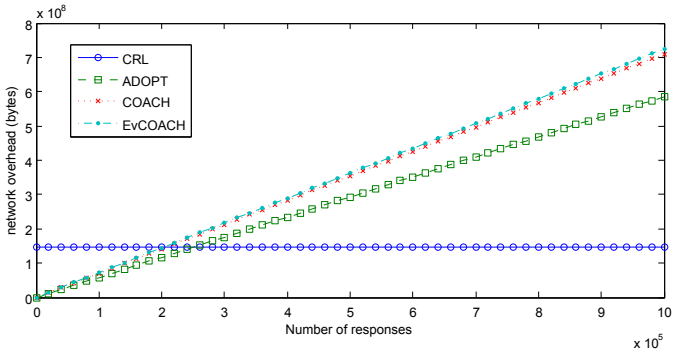


Figure 7: Network overhead vs number of responses.

6.1.2. Computational Cost

As an initial remark, we would like to mention that in the following evaluation, we consider the cryptographic delay only due to hashing and point multiplication on the elliptic curve, as they are the most time-consuming operations in the proposed protocol. Let T_{hash} and T_{mul} denote the time required to perform a pairing operation and a point multiplication, respectively. The elliptic curve digital signature algorithm is the digital signature method chosen by the VANET standard IEEE1609.2, where a signature generation takes T_{mul} and a signature verification takes $4\,T_{mul}$. In COACH, to verify a credential, a verifier must perform a hash operation to compute the current contents of the leaf node corresponding to the target serial number (SN_i). Finally, the verifier needs to perform $log\,N$ hash operations to compute the root of the tree using the $\mathcal{P}$ath. Therefore, the total computation overhead when checking the status of a certificate is $T_{hash}(logN + 1) + 4\,T_{mul}$. In [43], T_{mul} are found for an MNT curve with embedding degree $k = 6$ that is equal to 0.6 ms. In our simulation, we use an Intel Core i7 950 (at 3.07GHz) which is able to perform 1015952 SHA-1 Hashes per second, i.e, $T_{hash} = 0,98\mu$s.

On the other hand, EvCOACH introduces an additional delay due to the validation of the nonce transmitted along with the $\mathcal{P}$ath. In this case, the verifier must hash the nonce i times (where i represents the *currentUpdateIndex*) to obtain U_0. Once U_0 is calculated, an additional hash operation is needed to obtain the new root hash (i.e. H'_{root}). Therefore the total computation overhead due to a certificate status checking in EvCOACH is $T_{hash}(log\,N + i + 2) + 4\,T_{mul}$.

Table 2 shows the verification and signing delays of verifying k certificates for each revocation system. CRL is the best mechanism in terms of computational cost, as the CA only needs to sign once the CRL and vehicles only have to check this signature to validate the status of any certificate. ADOPT is based on OCSP in which typically each response is digitally signed, which imposes a high computational cost on responders. Furthermore, since the ADOPT responder is a trusted online server, its security requirements are stricter compared with a CRL distribution server. Finally, COACH and EvCOACH have a little more computational cost than CRL but much lower than ADOPT. This is mainly due to fact that the CA only has to sign the H_{root}, and vehicles only need to check this signature to validate each certificate. However, for each certificate this root has to be calculated which induces a verification delay that does not exist with CRL.

Mechanism	Verification delay	Signing delay
CRL	$4T_{mul}$	T_{mul}
COACH	$k(T_{hash}(log_2N + 1) + 4T_{mul})$	T_{mul}
EvCOACH	$k(T_{hash}(log_2N + i + 2) + 4T_{mul})$	T_{mul}
ADOPT	$k(4T_{mul})$	$k(T_{mul})$

Table 2: Computational cost of validating k certificates per revocation mechanism.

6.2. Evaluation Setup

In the previous section, we have shown analytically that COACH and EvCOACH mechanisms outperform CRL in terms of bandwidth efficiency and time to get fresh CSI. Moreover, COACH also improves other revocation mechanisms such as ADOPT when analyzing the availability of fresh CSI. In this section, we evaluate the proposed mechanism in a VANET scenario taking into account the specific characteristics of these networks.

A novel emulator tool [36] has been used to carry out the evaluation of COACH. Its accuracy and reliability is shown in [37]. This emulation platform allows to execute real-time applications over VANET, but for our purpose, we only use the simulation capabilities of this platform. The network emulation that this platform implements is

based on the Network Simulator ns-2 [26] with some add-ons to enhance its functionality. To this respect, we use an open sourcec micro-traffic simulator called SUMO [16] to generate a realistic mobility model so that results from the simulation correctly reflect the real-world performance of a VANET (see Fig. 8). Thus, the results obtained with this platform are equivalent to the ones that would result from simulating the same scenario using SUMO and ns-2.

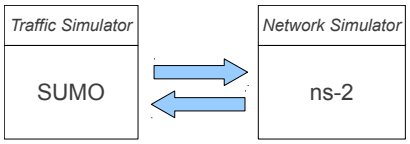


Figure 8: Simulation Architecture.

Using this emulation platform for VANET, we have evaluated the performance of our mechanism under three different environments. The reference scenario for all three environments is shown in Figure 9. This scenario consists of 4 two-lane roads forming a 1000x500m rectangle. RSUs are placed every 300 meters along the highway in order to achieve full coverage.

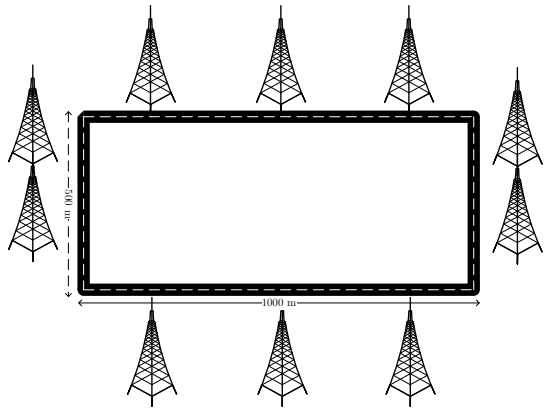


Figure 9: Reference Scenario.

Table 3 summarizes the values of the configuration parameters used in the reference scenario. Note that we have configured our simulation to use the Nakagami propagation model. We choose this propagation model because empirical research studies have shown that a fading radio propagation model, such as the Nakagami model, is best for simulation of a vehicular environment [39].

Using this scenario as reference, we vary the number of vehicles and their speed in order to test the performance of COACH. Firstly, we show that COACH performs better than CRL in these scenarios. Then, we estimate the overhead that the usage of COACH introduces compared to using standard CRLs.

In this way, we start by running a simulation for 8 hours, placing a single vehicle in the reference scenario. In this case, as there is full coverage and there are no other vehicles (i.e, channel contention is non-existent), the link to the infrastructure is quite stable. Figure 10 shows the

Parameter	Value
Area	1000x500m
Number of lanes	2
Number of RSUs	10
RSU Transmission range	300m
MAC	IEEE 802.11p
Bandwidth of a channel	10 MHz
Propagation model	Nakagami
Transport protocol	UDP

Table 3: Parameter values for the reference scenario.

obtained throughput for a single car at 72km/h, which remains fairly constant at 250 Kbps. Note that the VANETs are regarded as extreme environments that are opportunistically connected. Even if the average vehicle density is high, unbalanced traffic is inevitable and often leads to disconnection. In this simple scenario, the disconnection period is just of 15 minutes out of 8 hours (3%) which can be considered as short disconnection period in such challenged networks. These 15 minutes of disconnection are due to 48 disconnection events, so that the mean time of disconnection is about 30 seconds per event. In our case this disconnection events are mainly due to handoffs between different RSUs. As the time required to query and retrieve a COACH response is less than 200 ms, vehicles have enough time to check the validity of a certificate under these circumstances.

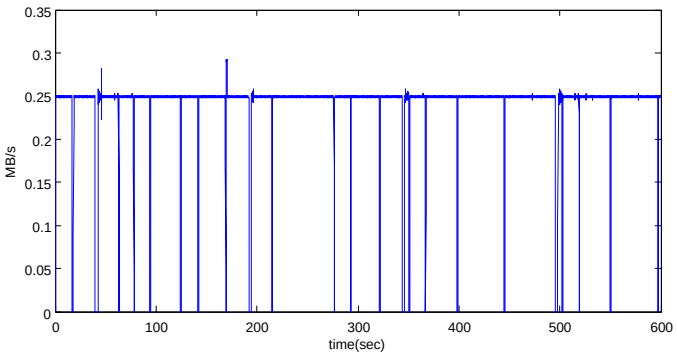


Figure 10: Throughput for one single car at 72 km/h.

Assuming that the CRL is encoded using some kind of Digital Fountain codes [22, 23, 20] (e.g. Raptor codes [38]), the total number of pieces (N) that a vehicle must download to complete the CRL is $N = (1 + \epsilon)M$ where M is the number of pieces in which the CRL is divided. Therefore, the size of the total number of pieces necessary to recover the CRL is slightly higher than the size of the original CRL. For instance, setting ϵ to 0.005, the amount of data necessary to recover a CRL for 50,000 vehicles is of approximately 146 Mbytes.

Figure 11 shows the time that a single vehicle needs to download a CRL depending on the size of the CRL and its speed. In our scenario, where the CRL is expected to

be around 145 Mbytes, the approximate time to download is 9 minutes. In contrast, the time required to retrieved fresh certificate status information using COACH is less than 3 ms. So COACH is five orders of magnitude faster than a standard CRL when checking the status of a given certificate. Note that, in this case scenario, the speed of the vehicle is not really decisive, i.e., depending on the speed the time to download a CRL does not vary significantly.

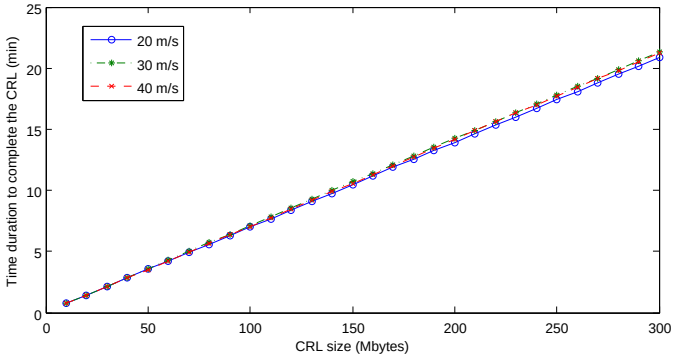


Figure 11: Time to download a CRL for a single vehicle at different speeds.

Once we have checked that COACH works better than the standard CRL in the case of a single vehicle no matter its speed, we test COACH performance when there is more traffic in the road. To do so, we placed 12 vehicles in the reference scenario moving at different speeds.

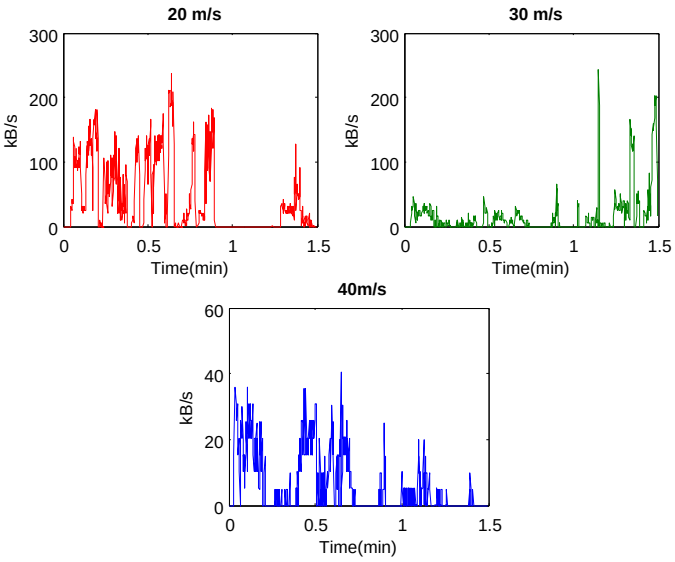


Figure 12: Mean throughput of 12 vehicles at three different speeds.

Figure 12 shows the mean throughput obtained for vehicles at three different speeds. Contrary to the single vehicle scenario, in this case the throughput varies with time significantly. Notice that there are large periods of time during which the vehicles are not able to establish a link with the infrastructure. This is mainly due to the medium contention. Moreover, the throughput also shows different patterns depending on the vehicle's speed. Thus, faster vehicles are more prone to suffer from network disruptions than slower vehicles.

As a consequence to the decrement of the throughput at high speeds, the time to download a CRL increases. In this manner, vehicles at low speeds are able to download the CRL faster. This pattern is shown in Figure 13. Moreover, it shows that slower vehicles are able to download bigger CRLs. In VANETS, where CRLs are expected to be of the order of hundreds of Mbytes, only vehicles at 20 m/s are able to download the whole CRL. On the other hand, vehicles at 40 m/s are only able to download a piece of the CRL of 120 Mbytes.

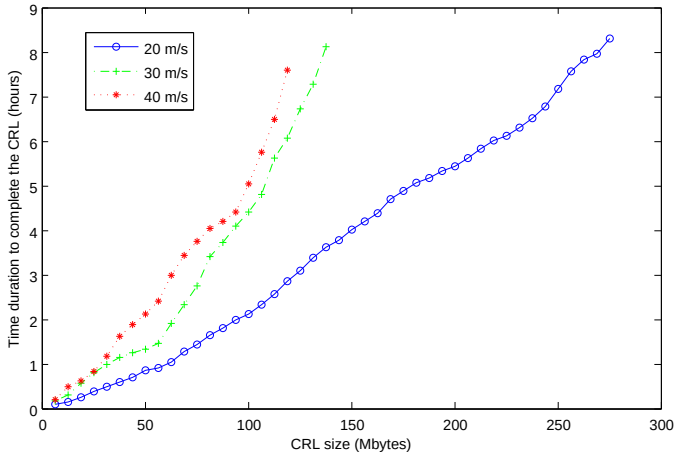


Figure 13: Mean time to download a CRL for a twelve vehicle scenario.

Table 4 shows the mean delays incurred when querying for the status of a given certificate. With *transmission delay* we denote the time to send the CSI query and the corresponding response. If we compare the transmission delay of the different revocation mechanisms, we can observe that ADOPT is the fastest but not so far from COACH. On the other hand, by *computational delay* we denote the time required to compose and validate a CSI response. In this case, ADOPT has the worst computational delay because each CSI response has to be signed by the CA. CRL computational delay is minimal as the CRL is only signed once and to searching the serial number of a certificate in the list has a computational cost of $O(log_2 N)$. COACH only requires one CA signature but a $\mathcal{P}$ath has to be computed each time a CSI response is required, so the computational cost is similar to the CRL. Finally, we define Round-Trip Time (RTT) as the time that takes since a vehicle requests for CSI until the status of a given certificate is validated. Therefore, the RTT is affected by the transmission, computational and propagation delays. ADOPT has the worst RTT due to the multihop transmission of the cached CSI, while CRL and COACH download the CSI directly from the repository in range. In any case, the vehicles' speed affects transmission

Vehicle Speed	Delay								
	COACH			CRL			ADOPT		
	Transmission	*Computational*	*RTT*	*Transmission*	*Computational*	*RTT*	*Transmission*	*Computational*	*RTT*
20 m/s	75 ms	2,401 ms	78 ms	3,521 h	2,400 ms	3,521 h	72 ms	3,612 ms	101 ms
30 m/s	149 ms	2,401 ms	157 ms	8,213 h	2,400 ms	8,214 h	122 ms	3,600 ms	312ms
40 m/s	173 ms	2,401 ms	187 ms	9,811 h	2,400 ms	9,813 h	152 ms	3,600 ms	421ms

Table 4: Delays when querying for CSI.

and RTT delays in all three revocation mechanisms.

Note that with these delays almost any envisioned application for VANET could timely perform the certificate validation. The most delay-constrained applications in VANETs are safety related (see Table 5). Since safety messages are sent out periodically by participating vehicles, in a dense network, a vehicle's onboard unit (OBU) may receive hundreds of such messages in a short time span. The ability to verify the status of these digitally signed messages quickly presents some challenges for OBUs since in order to keep the cost low, OBUs have limited computation power. COACH allows to quickly download fresh CSI and validate the status of a certificate. However, reaching the requirements of some safety applications such as "Pre-Crash Sensing" is not always feasible. Vehicles going at 40m/s or higher will not always be capable of validating the status of pre-crashing messages. In such situations, drivers going at such speeds must be aware of the risk associated with high speed. By contrast, data applications (such as peer-to-peer file sharing) are mostly delay-insensitive, and COACH will perform fine.

Application	Communication Requirements		
	Priority	*Latency (msec)*	*Comm Range*
Intersection Collision Warning	Class 1	∼100	≤ 300 m
Pre-Crash Sensing	Class 1	∼80	≤ 50 m
Transit Vehicle Signal Priority	Class 2	∼100	≤ 1000 m
Electronic Toll Collection	Class 3	∼100	≤ 20 m
Internet Access	Class 4	∼500	≤ 300 m
Roadside Service Finder	Class 4	∼500	≤ 300 m

Table 5: Representative applications.

On the other hand, as it was shown in the previous section, depending on the amount of COACH queries that a vehicle performs, CRL can outperform COACH. However, a vehicle has to generate more than 200,000 COACH queries during the lifetime of the certificate status information (which is typically short) to create a load in the network similar to the one created by a CRL. Figure 14 shows the mean number of COACH queries that a vehicle could generate per second. Note that almost in any case, a vehicle can receive a COACH response per second, and in some cases it can receive more than 180 responses per second. In this sense, a vehicle at 20 m/s can receive an average of 14 responses per second, while a vehicle at 40 m/s reduces the mean number of responses per second to 6. Therefore, any vehicle independently of its speed is able to check the status of at least half dozen of certificate per second. On the other hand, this experimental results show that a vehicle could generate more than 200,000 COACH queries during long road trips. In these cases, COACH will degrade the bandwidth efficiency while still being better in terms of availability than CRL.

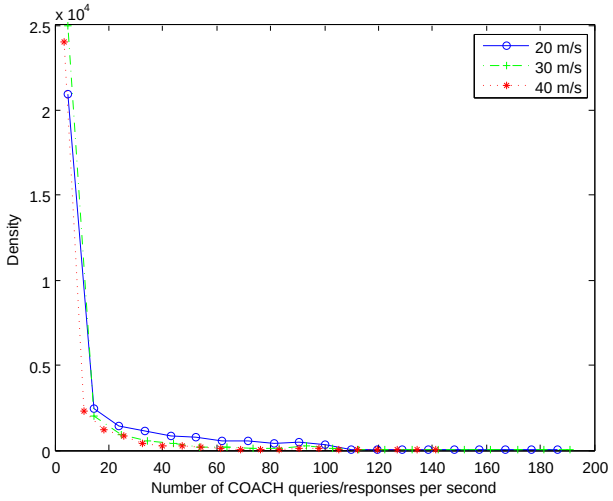


Figure 14: Mean number of COACH queries per second.

Finally, in order to extend the results obtained for the 12-vehicle scenario to a more crowded scenario, we double the number of vehicles that operate in the reference scenario. With more traffic in the network, as expected, we will see that COACH outperforms CRL even more clearly.

Figure 15 shows the throughput obtained for three vehicles at different speeds. Similarly to the 12-vehicle scenario, the throughput varies with time significantly. Again, there are large periods of time during which the vehicles are not able to establish a link with the infrastructure. Hence, faster vehicles are more prone to suffer from network disruptions than slower vehicles. Note that the throughput of any vehicle at 40 m/s is nearly zero during large periods of time, i.e., fast vehicles will not be able to download big files such as a CRL.

In this way, Figure 16 shows the mean time to download a CRL in this scenario. Contrary to the 12-vehicle scenario, no vehicle is able to download a CRL of 146 Mbytes during the 8 hours that the simulation lasted. The maxi-

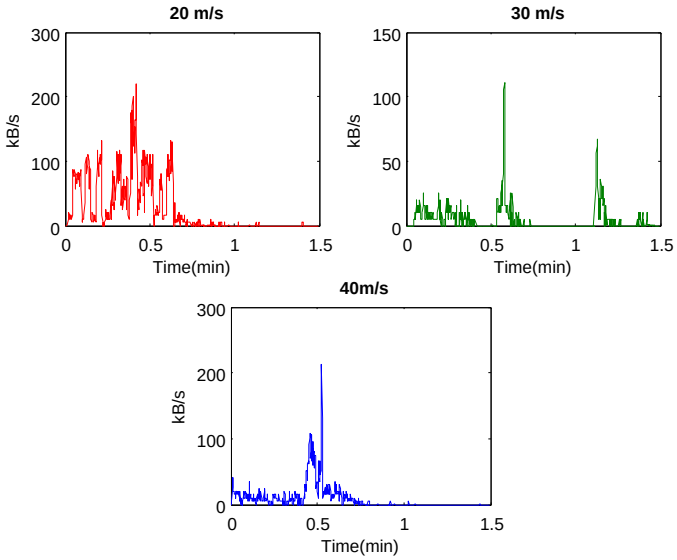


Figure 15: Throughput of 24 vehicles at three different speeds.

mum amount of information that a vehicle is able to download during this amount time is 69 Mbytes, and it is only achieved by slower vehicles. Therefore, with 24 vehicles the amount of time that a vehicle takes to download a CRL is unfeasible for the proper performance of the PKI. It is not viable that a vehicle spend more than 8 hours to download fresh certificate status information.

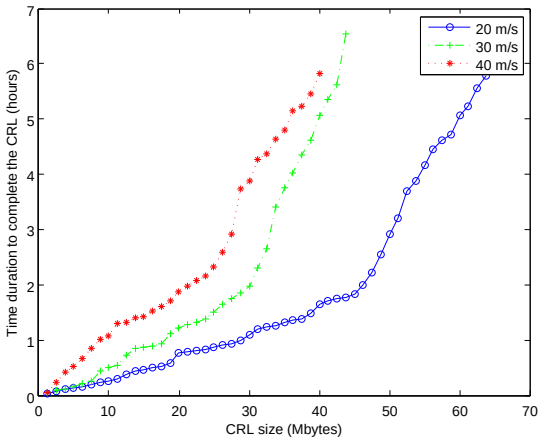


Figure 16: Mean time to download a CRL for a 24-vehicle scenario.

Therefore, it becomes necessary to use another mechanism to distribute the certificate status information. Using COACH a vehicle at 20 m/s can receive an average of 4 responses per second, and a vehicle at 40 m/s can receive 2 responses per second (see Figure 17). That means that in the worst case scenario a car can check at least two certificates per second in average.

In conclusion, COACH works better than CRL in terms of bandwidth efficiency no matter the vehicle speed. In the following we show the way EvCOACH improves COACH performance. In terms of network overhead, COACH and EvCOACH are very similar as they only differ in 15 bytes

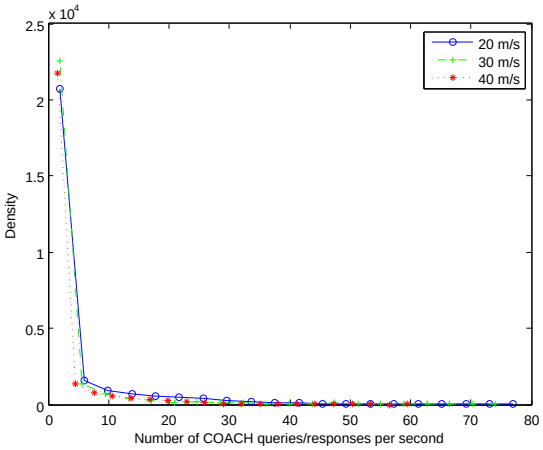


Figure 17: Mean number of COACH queries per second.

in the response size. Therefore, EvCOACH will also work better than CRL in the same way COACH does.

Besides the network load decrease that COACH induces in these scenarios, EvCOACH also enhances the certificate validation workload when there are no new revoked certificates. As explained in Section 4, by including a nonce in the responses, COACH revalidates previous cached certificate status information. Notice that a vehicle does need to download again the CRL and only needs to check the freshness and authenticity of the nonces. Table 6 shows the savings in time and bandwidth that EvCOACH provides when using the revalidation of the COACH tree during consecutive CSI updates. Note that the table shows the best case scenario when a vehicle is able to download a CRL in 3,5 hours. However, the benefits of using this mechanism are expected to be higher in a real scenario where vehicles will spend more than 8 hours to download a 146 Mbytes CRL.

	$d = 1$	$d = 2$	$d = 3$
Network bandwidth saving	$\sim$ 146 Mbytes	$\sim$ 292 Mbytes	$\sim$ 438 Mbytes
Time saving	$\sim$ 3.5 hours	$\sim$ 7 hours	$\sim$ 10.5 hours

Table 6: Network bandwidth and time saving of EvCOACH during consecutive CSI updates.

At this point, we have shown that COACH and EvCOACH performance surpasses standard CRL in the simulated scenarios in terms of CSI availability. Henceforward, we evaluate the overhead that COACH introduces to the network. As explained in the previous section, COACH introduces an extension to the standard CRL in order to allow any vehicle and RSU to become a repository. Figure 18 shows the overhead introduced by the *extended-CRL*.

Note that the overhead introduced is independent of the number of revoked certificates, so that in our scenario with 50,000 vehicles the introduced overhead represents a 0.011065‰ of the total size of the *extended-CRL*. There-

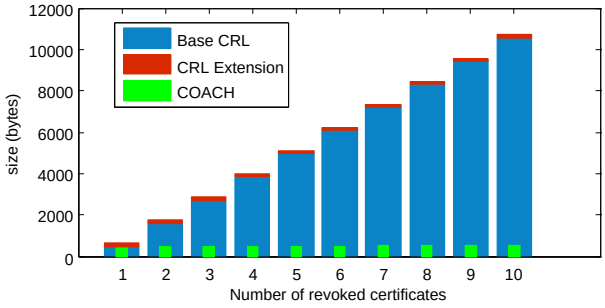


Figure 18: Overhead introduced by the *extended-CRL* (using ECDSA-256).

fore, we can conclude that this overhead is negligible in a VANET scenario. Additionally, Figure 18 shows that the size of the COACH responses grows much slower than the size of the CRL.

7. Conclusions

Local revocation approaches based on threshold cryptography and voting schemes provide mechanisms for revocation management inside the VANET. However, the local validity of the certificate status information and the lack of support for extending its validity to the global network restrain their utilization in the real VANET scenarios. These problems can be mitigated by adapting traditional PKI to the vehicular environment. Standard certificate validation mechanisms, such as CRL or OCSP, do not fit well in a VANET where huge number of nodes are involved and where several pseudonym certificates are assigned in addition to vehicle identity certificates.

In this paper, we have presented COACH, a collaborative certificate status checking mechanism based on an *extended-CRL*. The main advantage of this *extended-CRL* is that the road-side units and repository vehicles can build an efficient structure based on an authenticated hash tree to respond to status checking requests inside the VANET, saving time and bandwidth. In addition, we have described EvCOACH, an extension of the COACH mechanism that improves the performance of the standard protocol by avoiding to reissue useless certificate status information. In this context, EvCOACH allows to revalidate a previously downloaded CRL so that RSUs and repository vehicles reduce the computational cost of building the hash tree.

Analytical results show that allocating a small bandwidth is enough to ensure that vehicles receive certificate status responses within few seconds. The performance improvement is obtained at expenses of adding the signed hash tree extension to the standard-CRL. In this way, COACH becomes an offline certificate status validation mechanism as it does not need trusted responders to operate. Therefore, COACH significantly reduces the complexity of certificate management and achieves great efficiency and scalability, particularly when it is deployed in heterogeneous vehicular networks.

Acknowledgments

This work has been supported partially by the Spanish Research Council with Project (TEC2008-06663-C03-01), by Spanish Ministry of Science and Education with Project CONSOLIDER CSD2007-00004 (ARES), and SERVET (TEC2011-26452) and by Generalitat de Catalunya with Grant 2009 SGR-1362 to consolidated research groups. We would like also to thank the support and reviews of our friends Juan Miguel and Sergi Reñe.

[1] Abrougui, K., & Boukerche, A. (2011). Secure service discovery protocol for intelligent transport systems: proof of correctness. In *Proceedings of the first ACM international symposium on Design and analysis of intelligent vehicular networks and applications* DIVANet '11 (pp. 101–108).

[2] Abrougui, K., Boukerche, A., & Pazzi, R. (2010). Location-aided gateway advertisement and discovery protocol for vanets. *Vehicular Technology, IEEE Transactions on*, *59*, 3843 –3858.

[3] Armknecht, F., Festag, A., Westhoff, D., & Zeng, K. (2007). Cross-layer privacy enhancement and non-repudiation in vehicular communication. In *4th Workshop on Mobile Ad-Hoc Networks (WMAN'07)*.

[4] Bera, R., Bera, J., Sil, S., Dogra, S., Sinha, N., & Mondal, D. (2006). Dedicated short range communications (DSRC) for intelligent transport system. In *IFIP International Conference on Wireless and Optical Communications Networks* (pp. 1–5).

[5] Berkovits, S., Chokhani, S., Furlong, J., J.Geiter, & Guild, J. (1995). *Public key infrastructure study: Final report*. Technical Report MITRE Corporation for NIST.

[6] Chen, W., Guha, R., Chennikara-Varghese, J., Pang, M., Vuyyuru, R., & Fukuyama, J. (2010). Context-driven disruption tolerant networking for vehicular applications. In *Vehicular Networking Conference (VNC), 2010 IEEE* (pp. 33 –40).

[7] Fan, C.-I., Hsu, R.-H., & Tseng, C.-H. (2008). Pairing-based message authentication scheme with privacy protection in vehicular ad hoc networks. In *Proceedings of the International Conference on Mobile Technology, Applications, and Systems* Mobility '08 (pp. 82:1–82:7).

[8] Forné, J., Muñoz, J. L., Esparza, O., & Hinarejos, F. (2009). Certificate status validation in mobile ad hoc networks. *Wireless Commun.*, *16*, 55–62.

[9] Haas, J., Hu, Y.-C., & Laberteaux, K. (2011). Efficient certificate revocation list organization and distribution. *Selected Areas in Communications, IEEE Journal on*, *29*, 595 –604.

[10] Haas, J. J., Hu, Y.-C., & Laberteaux, K. P. (2009). Design and analysis of a lightweight certificate revocation mechanism for vanet. In *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking* VANET '09 (pp. 89–98). New York, NY, USA: ACM.

[11] Hubaux, J., Capkun, S., & Luo, J. (2004). The security and privacy of smart vehicles. *Security Privacy, IEEE*, *2*, 49 –55.

[12] IEEE (2006). IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages. *IEEE Std 1609.2-2006*, (pp. 1–117).

[13] Jain, S., Fall, K., & Patra, R. (2004). Routing in a delay tolerant network. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications* SIGCOMM '04 (pp. 145–158).

[14] Jiang, D., & Delgrossi, L. (2008). IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE* (pp. 2036–2040).

[15] Kim, J., Baek, J., Kim, K., & Zhou, J. (2011). A privacy-preserving secure service discovery protocol for ubiquitous computing environments. In *Proceedings of the 7th European con-*

ference on Public key infrastructures, services and applications EuroPKI'10 (pp. 45–60).

[16] Krajzewicz, D., Hertkorn, G., Rössel, C., & Wagner, P. (2002). Sumo (simulation of urban mobility); an open-source traffic simulation. In *4th Middle East Symposium on Simulation and Modelling (MESM2002)* MESM2002 (pp. 183–187).

[17] Laberteaux, K. P., Haas, J. J., & Hu, Y.-C. (2008). Security certificate revocation list distribution for vanet. In *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking* VANET '08 (pp. 88–89).

[18] Lin, X., Lu, R., Zhang, C., Zhu, H., Ho, P.-H., & Shen, X. (2008). Security in vehicular ad hoc networks. *Communications Magazine, IEEE*, *46*, 88 –95.

[19] Lu, R., Lin, X., Zhu, H., Ho, P.-H., & Shen, X. (2008). Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (pp. 1229 –1237).

[20] Luby, M. (2002). Lt codes. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on* (pp. 271–280).

[21] Ma, C., Hu, N., & Li, Y. (2006). On the release of crls in public key infrastructure. In *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*. Berkeley, CA, USA: USENIX Association.

[22] Mackay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*.

[23] Mackay, D. J. C. (2005). Fountain codes. *Communications, IEE Proceedings-*, *152*, 1062–1068.

[24] Marias, G., Papapanagiotou, K., Tsetsos, V., Sekkas, O., & Georgiadis, P. (to appear). Using trust evidences to deploy a distributed ocsp in manets. *EURASIP Journal on Wireless Communications and Networking Special Issue on Wireless Network Security,*, .

[25] Marias, G. F., Papapanagiotou, K., & Georgiadis, P. (2005). Adopt. a distributed ocsp for trust establishment in manets. *11th European Wireless Conference 2005*, .

[26] McCanne, S., & Floyd, S. (2000). The Network Simulator NS-2. `http://www.isi.edu/nsnam/ns/`.

[27] Merkle, R. (1989). A certified digital signature. In *Advances in Cryptology (CRYPTO89). Lecture Notes in Computer Science* 435 (pp. 234–246). Springer-Verlag.

[28] Merkle, R. (1990). A certified digital signature. In *Advances in Cryptology — CRYPTO' 89 Proceedings* (pp. 218–238). Springer Berlin / Heidelberg volume 435 of *Lecture Notes in Computer Science*.

[29] Moschetta, E., Antunes, R. S., & Barcellos, M. P. (2010). Flexible and secure service discovery in ubiquitous computing. *Journal of Network and Computer Applications*, *33*, 128–140.

[30] Myers, M., Ankney, R., Malpani, A., Galperin, S., & Adams, C. (1999). *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. RFC 2560 Internet Engineering Task Force.

[31] Nowatkowski, M., Wolfgang, J., McManus, C., & Owen, H. (2010). The effects of limited lifetime pseudonyms on certificate revocation list size in vanets. In *IEEE SoutheastCon 2010 (SoutheastCon), Proceedings of the* (pp. 380 –383).

[32] Papadimitratos, P., Buttyan, L., Holczer, T., Schoch, E., Freudiger, J., Raya, M., Ma, Z., Kargl, F., Kung, A., & Hubaux, J.-P. (2008). Secure vehicular communication systems: design and architecture. *Communications Magazine, IEEE*, *46*, 100 –109.

[33] Papadimitratos, P., Buttyan, L., Hubaux, J.-P., Kargl, F., Kung, A., & Raya, M. (2007). Architecture for secure and private vehicular communications. In *Telecommunications, 2007. ITST '07. 7th International Conference on ITS* (pp. 1 –6).

[34] Papadimitratos, P., Mezzour, G., & Hubaux, J.-P. (2008). Certificate revocation list distribution in vehicular communication systems. In *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking* VANET '08 (pp. 86–87).

[35] Raya, M., & Hubaux, J.-P. (2005). The security of vehicular ad hoc networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks* SASN '05 (pp. 11–21).

[36] Reñé, S., Mata, J., Alins, J., Hernadez, C., Muñoz-Tapia, J. L., Óscar Esparza, & Caubet, J. (2010). VANET emulator platform. `http://anka.upc.es`.

[37] Reñé, S., Gañán, C., Caubet, J., Alins, J., Mata, J., & Muñoz, J. L. (2011). Analysis of video streaming performance in vehicular networks. In *The First International Conference on Advanced Communications and Computation*. Barcelona, Spain.

[38] Shokrollahi, A. (2006). Raptor codes. *IEEE/ACM Trans. Netw.*, *14*, 2551–2567.

[39] Taliwal, V., Jiang, D., Mangold, H., Chen, C., & Sengupta, R. (2004). Empirical determination of channel characteristics for dsrc vehicle-to-vehicle communication. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks* VANET '04 (pp. 88–88). New York, NY, USA: ACM.

[40] Walleck, D., Li, Y., & Xu, S. (2008). Empirical analysis of certificate revocation lists. In *Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security* (pp. 159–174).

[41] Wasef, A., Jiang, Y., & Shen, X. (2010). DCS: An Efficient Distributed-Certificate-Service Scheme for Vehicular Networks. *Vehicular Technology, IEEE Transactions on*, *59*, 533 –549.

[42] Wasef, A., & Shen, X. (2009). Maac: Message authentication acceleration protocol for vehicular ad hoc networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE* (pp. 1 –6).

[43] Zhang, C., Lu, R., Lin, X., Ho, P.-H., & Shen, X. (2008). An efficient identity-based batch verification scheme for vehicular sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (pp. 246 –250).

International Journal of Innovative
Computing, Information and Control
Volume 8, Number 11, November 2012

 ISSN 1349-4198
pp. 7925–7942

RISK-BASED DECISION MAKING FOR PUBLIC KEY INFRASTRUCTURES USING FUZZY LOGIC

Carlos Gañán, Jose L. Muñoz, Oscar Esparza
Jorge Mata-Díaz and Juanjo Alins

Department of Telematics Engineering
Universitat Politècnica de Catalunya
C. Jordi Girona 31, Barcelona 08034, Spain
{ carlos.ganan; jose.munoz; oesparza; jmata; juanjo }@entel.upc.edu

Received July 2011; revised February 2012

Abstract. *Public key infrastructures (PKIs) are complex systems that are responsible for giving users enough information to make reasonable trust judgments about one another. The validation of public keys is hence of great importance. In general, validation of public keys is achieved by public-key certificates. However, in some circumstances certificates have to be revoked. Certificate revocation is generally achieved using Certificate Revocation Lists (CRLs) but the use of CRLs implicitly entails the risk of trusting a certificate that is not included in the list. This can be because the actual status of the certificate is unknown to the CA, or because the CRL is not updated. In this context, in this article we propose a fuzzy risk-based decision making system to assist any network user to make easier its decision-making process when using CRLs.*
Keywords: Risk management, Public key infrastructure, Revocation, Fuzzy logic

1. **Introduction.** From its earliest days in academia, the Internet was designed with the assumption that it was something akin to a "private club", with the main goal being the free exchange of ideas and information. With the birth of the World Wide Web in the early 1990s, the Internet was opened up to anyone, and it soon became clear that something would need to be done to allow users, public and technical alike, to confirm that they were communicating with correctly identified parties. Public Key Infrastructure (PKI) was developed to solve this problem. Moreover, PKI is envisioned as the security solution to provide integrity, authentication and non-repudiation to state-of-the-art networks such as vehicular networks, wireless sensor networks or mobile ad-hoc networks.

PKI is an information technology infrastructure that enables network users to securely and privately exchange information through the use of a public and a private key pair that is obtained and shared through a trusted authority. The public key infrastructure provides for a digital certificate that can identify an individual or an organization. Digital certificates are the means of accurately and reliably distributing public keys to users needing to encrypt messages or to verify digital signatures. Certificates are signed by certification authorities (CAs) and they are issued with a planned lifetime, which is defined through a validity start time and an explicit expiration date. Once issued, a certificate becomes valid when its validity start time is reached, and it is considered invalid after its expiration date. However, various circumstances may cause a certificate to become invalid prior to the expiration of the validity period. Such circumstances include the compromise or suspected compromise of the private key associated to the certificate, requiring to change the name of the subject of a certificate, and modifying the association type between subject and CA, for example, when an employee terminates employment with

an organization. Thus, the PKI has to collect and distribute information about revoked certificates. Currently deployed PKIs rely mostly on Certificate Revocation Lists (CRLs) for handling certificate revocation [1]. A CRL is a list identifying revoked certificates; it is signed by a CA and made available at public distribution points. The CRL has a validity period, and updated versions of the CRL are published before the previous CRL's validity period expires.

However, the use of CRLs to manage revocation in PKIs becomes a risk source. Each new CRL contains a large number of recent revoked certificates that differs from the previous CRL. The number of new revoked certificates will vary depending on the time elapsed since the previous CRL publication. These new revoked certificates are unknown to the user during the validity interval of the current valid CRL. It is during this validity interval when a user could be operating with a revoked certificate without knowing it. In this context, any user will be taking certain risk of operating with an unknown revoked certificate. Therefore, total security is unattainable, even under the unrealistic assumption that CRLs can be delivered to everyone instantaneously. A private key may be compromised long before the compromise is noticed and the certificates revoked. This cannot be handled by current revocation schemes, but it should be taken into consideration when analyzing the inherent risk in a PKI. This article is mainly motivated by the lack of current revocation schemes to manage this risk. This risk associated with the use of a PKI cannot be completely removed, but it can be analyzed and controlled. It is clear that different applications have different risk requirements and that different users have different preferences in the risk-cost balance. Therefore, a PKI aiming to support multiple applications should provide a revocation interface that is tunable. Users should be able to set different recency requirements based on their needs and resources. The aim of this article is to develop a new risk analysis method to identify and assess this risk in an acceptable way in which any risk information is processed and reliably applied to the users' decision-making process.

Previous works in the literature [2-6] acknowledged the existence of an operational risk when using a revocation mechanism such as CRLs. However, these works neither quantified this risk nor provided a means to deal with it. Authors in [7] calculated the probability of considering a certificate as valid when the real status known by the PKI is revoked. This work could be considered as the first step towards making network users aware of the hazards of operating under a PKI, though authors just provided a simple metric to measure the percentage of unknown revoked certificates. The main drawbacks of this proposal are that the standard CRL has to be extended to allow users to calculate this probability, and authors did not deal with the potential consequences of trusting an outdated CRL. To the best of our knowledge, our proposal is the first model that provides a risk indicator to help network users in their decision-making process, taking into account all the risk sources that are present in a PKI.

However, decision making is a tough process. It involves dealing with a lot of uncertainty and projecting what the outcome might be. Depending on the projection of the final outcome, a decision has to be made. So, in order to ease the decision-making process we propose the utilization of a fuzzy system. The user can ease its decision-making process by utilizing a fuzzy approach to risk-based decision making. This fuzzy approach will allow users to further strengthen their previous belief of proceeding in an interaction with a probable trusted user (i.e., a user whose certificates have not been revoked). The proposed risk-based decision making model combines the possibility of interacting with an

illegitimate user and its possible consequences and gives an output to the trusting user[1]. Fuzzy logic is used to model uncertainty, and similarly, the decision-making process deals with predicting the possible uncertain outcome and deciding the future course of action based on the predicted uncertain outcome.

Now, being more specific, we can precise that the goals of this article are to explore the inherent risk in conventional PKIs and to show how fuzzy logic risk analysis techniques can successfully help in the analysis of this risk. By focusing on these goals, we have modeled and characterized a risk-based decision-making system based on fuzzy logic. To that end, in the first place, we analyze the risk of operating with CRLs, determine the possibility of trusting a revoked certificate and analyze the possible consequences of an interaction with a potential illegitimate network user. Then, taking into account the information that users can obtain from the CRLs, we design a fuzzy inference system that gives as output the risk of operating with a particular CRL. Our proposed model can provide to the users an idea of how risky is to operate with their current CRL, and it can also help them to make risk-based decisions. Finally, a case study on risk analysis of a CRL issued by an actual CA (GoDaddy) is used to show the validity of the proposed model. The results of the risk assessment in the case study are represented as risk score, located in a defined range, and risk category with linguistic words, which indicate that by using the proposed methodology the risk associated with CRLs can be assessed effectively and efficiently.

Following this research method, we obtain a fuzzy system that models risk and assists users in their decision-making processes related to certificate revocation. Our solution takes into account a set of key risk factors to estimate the risk of operating when using a particular revocation service. In this respect, we have identified potential risk sources involved in the revocation system and we have characterized them using fuzzy logic. Based on the estimated risk, users can decide whether to interact or not with another PKI user. The results show that although this CA is issuing CRLs with a frequency of only one day, there is still an inherent risk that our model is able to measure.

The rest of this article is organized as follows. In Section 2, we briefly review the basics of fuzzy logic. In Section 3, we identify and characterize a fuzzy inference system that allows estimating the risk of operating with CRLs. Next, in Section 4, we present an empirical case study using data collected from one of the most extended CAs. Finally, in Section 5, we conclude and point out possible future directions.

2. **Fuzzy Logic.** Fuzzy logic aims to model human thinking and reasoning. The key advantage of fuzzy methods is how they reflect the human mind in its remarkable ability to store and process information that is imprecise, uncertain, and resistant to classification [8]. This kind of logic intends to equip computers with the ability to process special data and to work by making use of human experiences and insights. When human logic solves problems, it creates verbal rules such as "if <event realized> is this, the <result> is that" [9]. Fuzzy logic tries to adapt these verbal rules and the human capability to make decisions to computers [10]. It uses verbal variables and terms together with verbal rules [11]. Usually, verbal rules and terms used in human decision-making process are fuzzy rather than precise. Adapting human logic system to computers increases problem-solving capabilities of computers.

Verbal terms and variables are expressed mathematically as membership degrees and membership functions. Fuzzy decision-making mechanisms use symbolic verbal phrases instead of numeric values. Systems that use fuzzy logic are alternatives to the difficulty

[1]A trusting user is a network agent that makes an informed decision of whether to interact with another agent or not, by analyzing beforehand the possible level of risk that could be present in that interaction.

of mathematical modeling of complex non-linear problems and fuzzy logic meets mathematical modeling requirement of a system.

In this context, fuzzy logic emerges as an alternative to the classical logic where every proposition must either be "true" or "false". Instead, fuzzy logic asserts that things can be simultaneously "true" and "not true", with a certain membership degree to each class [12]. It is based on membership functions and linguistic parameters to express vagueness in security issues. Fuzzy logic has the power to handle the concept of "partial truth" to quantify uncertainties associated with linguistic variables. It allows defining a *degree of membership* of an element in a set by means of a membership function. For classical or *crisp* sets, the membership function only takes two values: 0 (non-membership) and 1 (membership). In fuzzy sets the membership function can take any value from the interval $[0, 1]$. The value 0 represents complete non-membership, the value 1 represents complete membership, and values in between are used to represent partial membership [13].

Summing up, fuzzy logic provides a way to use imprecise and uncertain information generated by the system and human judgments in a precise way. In the case of a PKI, as the revocation data available do not provide proper statistical treatment, fuzzy arithmetic is clearly suitable, since it works well for addressing poorly characterized parameters and linguistic variables. Using a fuzzy inference system, we will show that we are able to map a given input set of variables (e.g., CRL age or revocation causes) to an output (e.g., risk indicator). The mapping then provides a basis from which decisions can be made, or patterns discerned. The process of fuzzy inference involves membership functions, logical operations, and If-Then rules. There are two main types of fuzzy inference systems: Mamdani-type and Sugeno-type. These two types of inference systems vary somewhat in the way outputs are determined. In this article, we will use the Mamdani-type inference system due to its adequacy to the problem in question.

2.1. **Mamdani's fuzzy inference method.** Mamdani's fuzzy inference method is the most commonly used fuzzy methodology [14, 15]. Mamdani's method was among the first control systems built using fuzzy logic [16]. It was proposed by Mamdani as an attempt to control a steam engine and boiler combination by synthesizing a set of linguistic control rules obtained from experienced human operators [17]. Mamdani's effort was based on Zadeh's paper on fuzzy algorithms for complex systems and decision processes [18].

The Mamdani-style fuzzy inference process is performed in four steps (see Figure 1):

1. Fuzzification of the input variables.
2. Rule evaluation (inference).
3. Aggregation of the rule outputs (composition).
4. Defuzzification.

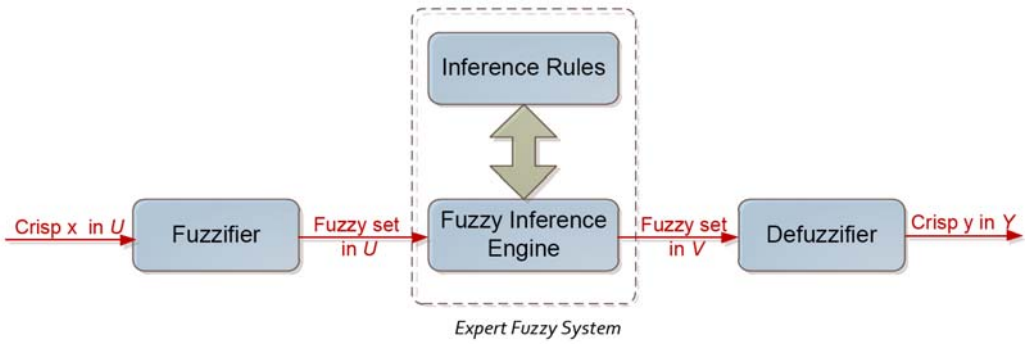


FIGURE 1. A fuzzy logic inference system

A fuzzification operator has the effect of transforming crisp data into fuzzy sets. In most of the cases fuzzy singletons are used as fuzzifiers:

$$fuzzifier(x_0) := \bar{x}_0, \tag{1}$$

where x_0 is a crisp input value from a process.

Suppose now that we have two input variables x and y. A fuzzy control rule

$$R_i : \text{if } x \text{ is } A_i \text{ and } y \text{ is } B_i \text{ then } z \text{ is } C_i,$$

is implemented by a *fuzzy implication* R_i and is defined as:

$$R_i(u, v, w) = [A_i(u) \text{ and } B_i(v)] \to C_i(w), \tag{2}$$

where the logical connective *and* is implemented by the minimum operator, i.e.,

$$\begin{aligned}[A_i(u) \text{ and } B_i(v)] \to C_i(w) &= [A_i(u) \times B_i(v)] \to C_i(w) \\ &= \min [A_i(u), B_i(v)] \to C_i(w).\end{aligned} \tag{3}$$

Fuzzy control rules are combined by using the sentence connective *also*. Since each fuzzy control rule is represented by a fuzzy relation, the overall behavior of a fuzzy system is characterized by these fuzzy relations.

In other words, a fuzzy system can be characterized by a single fuzzy relation which is the combination of the fuzzy relations in the rule set. The combination in question involves the sentence connective *also*. Symbolically, if we have the collection of rules:

R_1 : if x is A_1 and y is B_1 then z is C_1,
also
R_2 : if x is A_2 and y is B_2 then z is C_2,
also
...
R_n : if x is A_n and y is B_n then z is C_n.

The procedure for obtaining the fuzzy output of such a knowledge base consists of the following three steps:

- Finding the firing level of each of the rules;
- Finding the output of each of the rules;
- Aggregating the individual rule outputs to obtain the overall system output.

To infer the output z from the given process states x, y and fuzzy relations R_i, we apply the compositional rule of inference:

R_1 : if x is A_1 and y is B_1 then z is C_1,
also
R_2 : if x is A_2 and y is B_2 then z is C_2,
also
...
R_n : if x is A_n and y is B_n then z is C_n,
input x is $\bar{x}_0$ and y is $\bar{y}_0$

Consequence: z is C

where the consequence is computed by:

$$\text{consequence} = \mathbf{Agg}\ (\text{fact } \circ R_1, \ldots, \text{ fact } \circ R_n). \tag{4}$$

That is,

$$C = \mathbf{Agg}(\bar{x}_0 \times \bar{y}_0 \circ R_1, \ldots, \bar{x}_0 \times \bar{y}_0 \circ R_n), \tag{5}$$

taking into consideration that

$$\bar{x}_0(u) = 0, \quad u \neq x_0, \tag{6}$$

and

$$\bar{y}_0(u) = 0, \quad v \neq y_0. \tag{7}$$

The computation of the membership function of C is very simple:

$$C(w) = \mathbf{Agg}\ \{A_1(x_0) \times B_1(y_0) \to C_1(w), \ldots, A_n(x_0) \times B_n(y_0) \to C_n(w)\} \tag{8}$$

for all $w \in W$.

In the particular case of a Mamdani inference system, the fuzzy implication is modeled by Mamdani's minimum operator and the sentence connective *also* is interpreted as ORing the propositions and defined by the `max` operator.

Thus, the procedure for obtaining the fuzzy output of such a knowledge base can be formulated as:

- The firing level of the i-th rule is determined by:

$$\alpha_i = \min(A_i(x_0), B_i(y_0)).$$

- The output of the i-th rule is calculated by:

$$C_1'(w) = \min(\alpha_i, C_i(w)), \quad \forall w \in W.$$

- The overall system output, C, is obtained from the individual rule outputs C_i' by:

$$C(w) = \max(C_i'(w)), \quad \forall w \in W.$$

The output of the inference process so far is a fuzzy set, specifying a possibility distribution of control action. In the on-line control, a nonfuzzy (crisp) control action is usually required. Consequently, one must defuzzify the fuzzy control action (output) inferred from the fuzzy control algorithm, namely:

$$z_0 = defuzzifier(C), \tag{9}$$

where z_0 is the nonfuzzy control output and $defuzzifier$ is the defuzzification operator.

Defuzzification is a process to select a representative element from the fuzzy output C inferred from the fuzzy control algorithm. The most often used defuzzification operators are Center-of-Area/Gravity, First-of-Maxima, Middle-of-Maxima, Max-Criterion and Height Defuzzification [19].

It is possible, and in many cases much more efficient, to use a single spike as the output membership function rather than a distributed fuzzy set. This is sometimes known as a singleton output membership function, and it can be thought of as a pre-defuzzified fuzzy set. This enhances the efficiency of the defuzzification process because it greatly simplifies the computation required by the more general Mamdani method, which finds the centroid of a two-dimensional function. Rather than integrating across the two-dimensional function to find the centroid, the weighted average of a few data points. Sugeno type systems support this type of model [20]. In general, Sugeno type systems can be used to model any inference system in which the output membership functions are either linear or constant.

3. **Risk Assessment Model for PKI.** A risk analysis should cover all aspects of risks in the revocation process in question and should also specify how the risks involved are to be minimized. Therefore, it should include sufficient particulars to demonstrate that hazards with the potential to cause the network failure can be identified and evaluated, and that the appropriate measures have been taken to reduce risks.

A typical risk assessment framework consists of four stages: risk identification, risk assessment, risk response, and risk monitor and review. The nature of revoking certificates has imposed substantial uncertainties and subjectivities in the risk analysis process. However, no risk assessment method has been proposed to quantify this risk. Fuzzy reasoning techniques have proven useful to handle ill-defined and complex problems arising in other environments to reach a reliable decision [21-24].

In the following, we define a Fuzzy Inference System based on the Mamdani model to capture the risk of operating with a PKI when CRLs are used in the revocation scheme. To do so, in first place, we analyze the risk factors involved in using CRLs and their consequences, and then, we describe each one of the components of the model, i.e., the fuzzifier, the rules of the fuzzy logic system and the defuzzifier. The details are described in the following sections.

3.1. **Analyzing the risk when operating with CRLs.** As discussed earlier, any network user has to make decisions whether to interact or not with a probable legitimate user. Our proposal consists in making users aware of the risk they are taking when assuming as comprehensive the information contained in their cached CRL. Then, users will be able to make an informed decision by analyzing the possible risk that could be present in their interaction. The risk analysis of the trusting user in its potential interaction with an illegitimate user can be done by:

1. Determining the possibility of operating with users that have their certificate revoked;
2. Determining the possible consequences of operating with an illegitimate user.

Hence, the trusting user should consider these two factors for each probable illegitimate user in order to determine the possible risk associated with this interaction. Based on the analysis, the user can make an informed decision of whether to interact or not with another user of the network.

3.1.1. *Determining the possibility of trusting a revoked certificate.* Before operating with another network user, each user has to check the legitimacy of that user. This is achieved by checking the status of her certificate. To check this status, users have to corroborate that the serial number of the user in question is not contained in the CRL. However, though this serial number is not included in the CRL, it could be revoked. As users operate with cached CRLs, certificates that have been revoked after the issuance instant of the CRL are unknown to the users. Thus, as the information contained in the CRL is not totally comprehensive, users have to determine the possibility of trusting a revoked certificate.

This possibility of trusting a revoked certificate captures the extent to which a user thinks that another user of the network is illegitimate. This possibility increases with the time elapsed since the issuance of the CRL. The trusting user can determine this possibility of interacting with a probable illegitimate user by analyzing the information contained in the CRL, i.e., the number of revoked certificates and the issuance and update time of the CRL. Thus, the trusting user should analyze the possibility of operating with users whose certificates have been revoked, and in accordance to the context and criteria of its future interaction with them.

With revocation, users could control this possibility by, for instance, setting freshness requirements for CRL acceptance. Smaller freshness requirements require lower communication costs but lead to a higher risk. Setting the right freshness requirement requires risk analysis and balancing the risk and the cost. It is clear that different applications have different risk requirements and that different users have different preferences in the risk-cost balance. Users should be able to control the possibility of trusting a revoked certificate by setting different CRL recency requirements based on their needs and resources.

3.1.2. *Determining the possible consequences of an interaction.* The trusting user in order to gauge the potential risk in an interaction should also determine the possible consequences of operating with the probable illegitimate user apart from determining the possibility of trusting a revoked certificate. For instance, in a peer-to-peer financial interaction, the possible loss that a trusting user could suffer is usually the financial loss in its resources that are involved in the interaction.

The consequences of operating with users whose certificate has been revoked will be modeled over a scale of 0-10 representing the loss incurred. The possible consequences will vary depending on the revocation cause of the certificates. The PKIX/X.509 certificate and CRL specification [25] defines nine reason codes for revocation of a public-key certificate (see Table 1).

Note that we have defined a weight value w_i for each of the possible revocation causes representing the aforementioned potential threat incurred during the interaction with an illegitimate user. This weighting will allow us to give more importance to those certificates which were revoked due to a key compromise or malicious use. This weighting is purely intuitive as there are some revocation causes that pose bigger threats to the users than other causes. For instance, the compromise of the private key of the CA is more dangerous and has potentially more disastrous consequences than a superseded certificate.

TABLE 1. Revocation codes, weight values w_i and description

Numerical Code	Revocation Code	w_i	Description
(1)	`keyCompromise`	9	Private key has been compromised.
(2)	`cACompromise`	10	Certificate authority has been compromised.
(3)	`affiliationChanged`	1	Subject's name or other information has changed.
(4)	`superseded`	0	Certificate has been superseded.
(5)	`cessationOfOperation`	1	Certificate is no longer needed.
(6)	`certificateHold`	3	Certificate has been put on hold.
(7)	`removeFromCRL`	0	Certificate was previously on hold and should be removed from the CRL.
(8)	`privilegeWithdrawn`	5	Privileges granted to the subject of the certificate have been withdrawn.
(9)	`aACompromise`	10	Attribute authority has been compromised.

Once the user determines the possibility of trusting a revoked certificate and the potential consequences, she should combine those to determine the risk in order to assist decision making. As mentioned earlier, decision making is a tough process as it involves in dealing with a lot of uncertainty. The trusting user, in spite of determining the possibility of trusting a revoked certificate and the possible loss in its resources, might still be uncertain or undecided whether to interact or not with the particular user. To alleviate this problem, we propose the utilization of a fuzzy system which will help the trusting user in its decision-making process. We describe the fuzzy system in the next section.

3.2. **Developing a fuzzy risk based decision making system.** Once the possibility of trusting a revoked certificate and its consequences have been determined, we need a systematic approach to synthesize these constituents of risk into a given risk value for making an informed decision. To this end, we propose the use of a fuzzy approach. The main aim of the fuzzy decision making system is to assist the trusting user with the decision making process. To achieve that, we propose that the trusting user inputs the relative values of the probable trusted user to the fuzzy system, which in turn evaluates them according to the pre-defined rules. Based on the evaluations of the rules, an output is given to the trusting user. The output of the fuzzy system will be a risk value that depending on the user's attitude towards risk, will decide to proceed or not.

3.2.1. *Inputs of the fuzzy inference system.* The first stage to build the fuzzy risk inference system consists in identifying the factors that contribute to the risk. In this way, we identify the key risk factors (KRF) in the revocation process. These key factors are directly related to the aforementioned process of determining the possibility of having a revoked certificate and the consequences of operating with a user whose certificate has been revoked.

The herein proposed mapping technique consists in identifying the key risk factors, capable of reasonably signaling priorities and strategies for risk management purposes. These key risk factors will be those variables, either quantitative or qualitative, which together will serve the purpose of estimating the probability and severity of risk events at the task level.

As proposed in [26], the definition of the KRFs should observe five convenient features:

- *Relevancy*: variables should effectively capture a specific KRF;
- *Generality*: variables can be used across processes or tasks;
- *Non-redundancy*: avoid correlated KRFs;
- *Measurability*: variables should be quantifiable and verifiable;
- *Monitoring facility*: cost and simplicity of monitoring.

According to these features, the proposed KRFs are

1. **Number of revoked certificates** ($NumRev$): as users have cached CRLs which include the list of revoked certificates and their revoked date, users can know the number of revoked certificates per day;
2. **Revocation categories** ($RevCat$): CRLs can also include the revocation cause of each certificate;
3. **Age of the CRL** (CRL_{age}): using also the information contained in the CRL; users can calculate the time elapsed since the issuance of the CRL.

3.2.2. *Fuzzification procedure.* Afterwards, the fuzzification procedure has to be defined to capture KRFs and to be able to translate them into quantitative variables. The foundation of this procedure is the design of the fuzzy sets and the membership functions. The fuzzy set theory will make possible to obtain the imprecise and vague, yet valuable and irreplaceable, judgment of the people associated with the tasks and processes to be

evaluated. It would be clumsy and imprecise to ask for true or false, yes or no, 1 or 0 answers when dealing with variables such as expertise, impact or probability.

In order to translate the judgment of the people into a quantitative variable, the corresponding *membership* functions should be defined for each KRF. Membership functions can have different shapes. The most commonly used shapes are triangular, trapezoidal, Gaussian and bell shaped membership functions. Therefore, for each KRF we choose an appropriate membership function.

The first KRF, the number of revoked certificates, is described by Gaussian shaped membership functions in the following three fuzzy sets as depicted in Figure 2:

- *Low*: the number of revoked certificates per day is low;
- *Moderate*: the number of revoked certificates is neither low nor high;
- *High*: the number of revoked certificates per day is high.

In our fuzzy system, the range of the input $NumRev$ is within $[0, 20]$.

The second KRF, the revocation category, is described by a Generalized bell-shaped membership function. Three different categories are defined according to weight values defined in the Table 1 (see Figure 3).

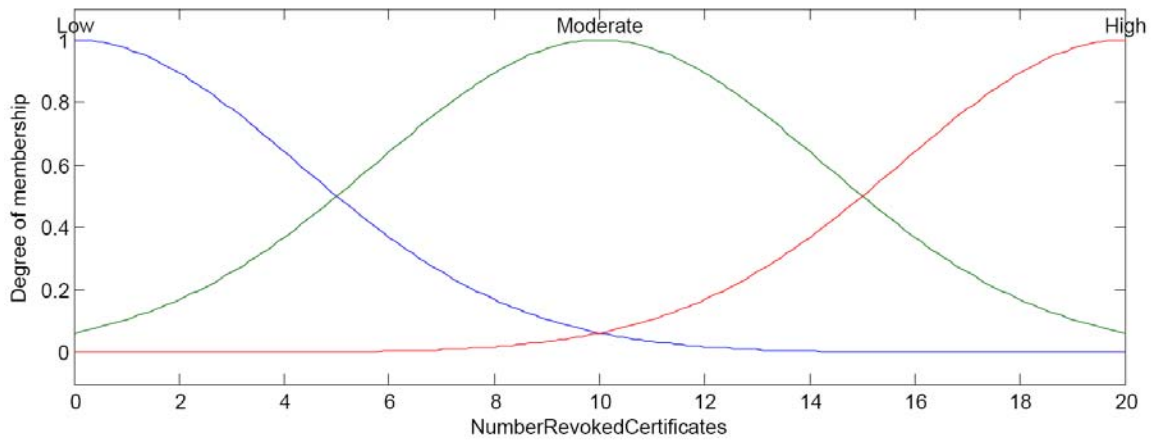


FIGURE 2. Number of revoked certificates as a fuzzy variable

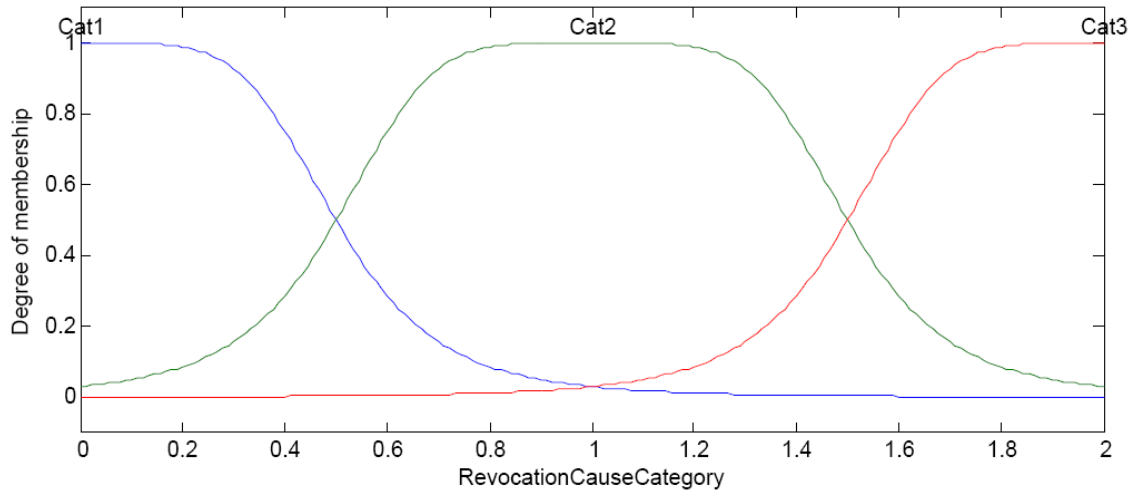


FIGURE 3. Revocation cause categories as a fuzzy variable

- *Category 1*: This category includes revoked certificates that represent a low risk to the network. Thus, this category includes revocation causes such as superseded, change of affiliation or cessation of operation;
- *Category 2*: This category includes revoked certificates that represent a moderate risk to the network. Thus, this category includes revocation causes such as privilege withdrawn or certificate on hold;
- *Category 3*: This category includes revoked certificates that represent a high risk to the network. Thus, this category includes revocation causes such as key compromise, CA compromise or Attribute Authority compromise.

In our fuzzy system the range of the input $RevCat$ is within $[0, 2]$.

Finally, we represent the last KRF, the CRL age, by means of a triangular shaped membership function (see Figure 4).

- *New*: Describes CRLs that have recently being updated. In a scale of 0-34 hours, this category ranges from 0 to 9.6 hours.
- *Old*: Describes CRLs that were updated some days ago. In a scale of 0-34 hours, this category ranges from 2.4 to 21.6 hours.
- *Very Old*: Describes CRLs that have not been updated for several weeks. In a scale of 0-34 hours, this category ranges from 14.4 to 33.6 hours.

The output of risk is defined in the following five classes as shown in Figure 5:

1. *Unacceptability High*: High probability that trusting a potentially revoked certificate will cause important damages in the network and potential losses.
2. *High*: High probability of trusting a revoked certificate and cause moderate damages in the network.
3. *Moderate*: Probability of trusting a revoked certificate is intermediate and the potential consequences are limited.
4. *Low*: Risk of operating with an illegitimate user is low.
5. *Negligible*: There is almost no risk in trusting the information contained in the CRL as comprehensive.

3.2.3. *Rules for the fuzzy logic system.* According to the Mamdani approach, we need some rules to process the inputs to let the fuzzy system to conclude at an output. Linguistic rules in the fuzzy system consists of two parts, an antecedent (between the IF and

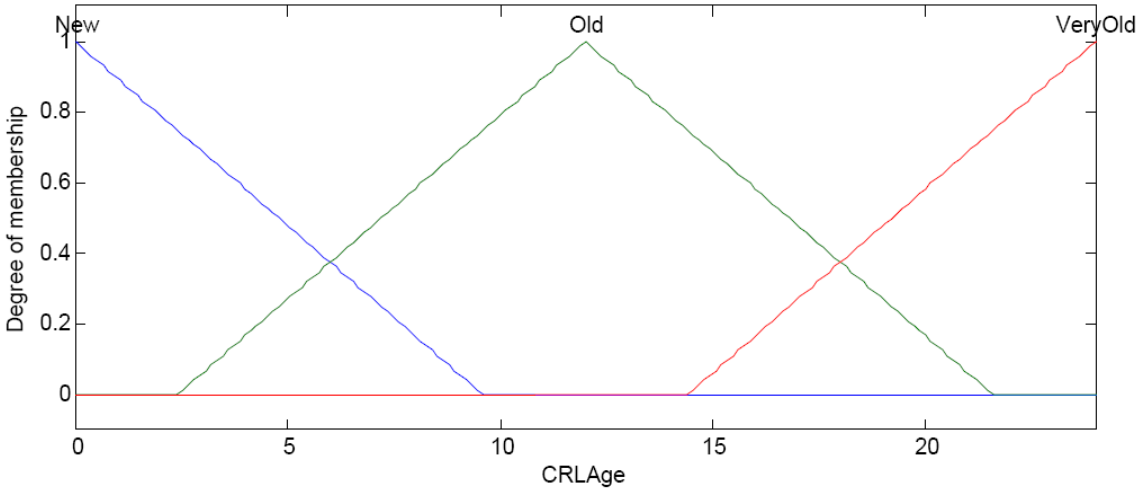


FIGURE 4. CRL age as a fuzzy variable

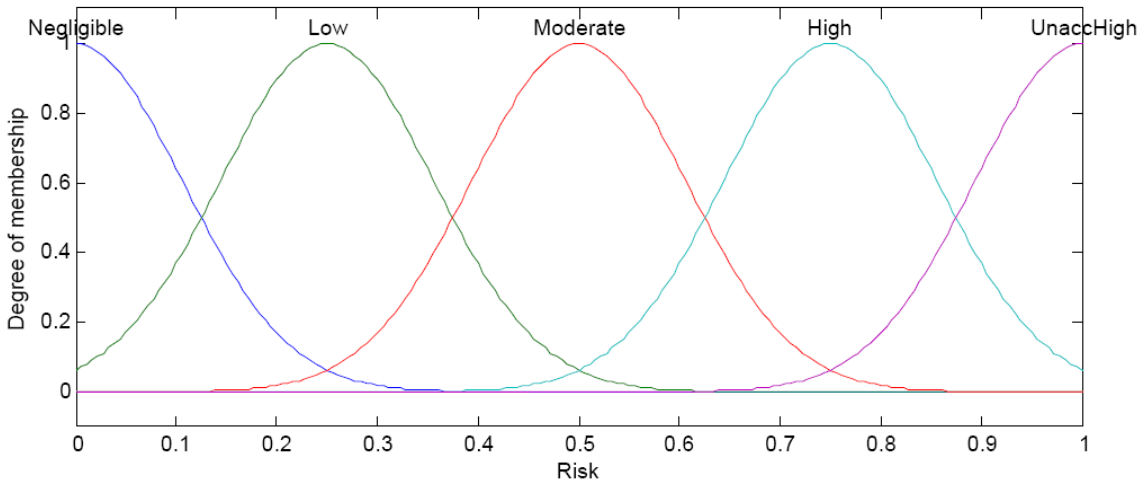


FIGURE 5. Risk as a fuzzy variable

THEN) and consequent (following THEN). There are 3 inputs to our fuzzy system and 3 fuzzy sets. Hence, the total number of rules is: $3^3 = 27$. However, some of these rules are correlated so they can be simplified. There is not a method for establishing the optimal number of inference rules, but achieving an intuitive, smooth and continuous solution space for every combination of KRFs is a fair rule of thumb. Following this guideline, we define eleven inference rules. These rules relate inputs and outputs as:

R_1: If ($NumRev$ is Low) and (CRL_{age} is New) then (Risk is Negligible)
R_2: If ($NumRev$ is High) and (CRL_{age} is New) then (Risk is Low)
R_3: If ($NumRev$ is Low) and (CRL_{age} is Old) and ($RevCat$ is Cat1) then (Risk is Low)
R_4: If ($NumRev$ is Low) and (CRL_{age} is Old) and ($RevCat$ is Cat2) then (Risk is Moderate)
R_4: If ($NumRev$ is Low) and (CRL_{age} is Old) and ($RevCat$ is Cat3) then (Risk is High)
R_6: If ($NumRev$ is Moderate) and (CRL_{age} is Old) then (Risk is High)
R_7: If ($NumRev$ is High) and (CRL_{age} is Old) then (Risk is High)
R_8: If (CRL_{age} is VeryOld) and ($RevCat$ is Cat3) then (Risk is UnaccHigh)
R_9: If (CRL_{age} is VeryOld) and ($RevCat$ is Catl) then (Risk is Moderate)
R_{10}: If (CRL_{age} is VeryOld) and ($RevCat$ is Cat2) then (Risk is High)
R_{11}: If ($NumRev$ is Moderate) and (CRL_{age} is New) then (Risk is Low)

This set of inference rules (or knowledge base) has the objective of deconstructing expert's knowledge and encoding it in a form that the fuzzy logic inference system is capable of mimicking human's reasoning capabilities to solve complex systems. Therefore, an expert (or group of experts) analyzes the KRFs, their different linkages and their relation to the linguistic variables in the output space, resulting in a list or set of educated inference rules that will solve simultaneously any combination of inputs and calculate the expected OR Indicator.

3.2.4. *Defuzzification.* Having specified the input space, the output space, and the rule base, the method for estimating the expected risk is to be defined. Cox in [27] highlights centroid's consistency and well-balanced approach, its sensitiveness to the height and width of the total fuzzy region and the smooth changes in the expected value of the output across observations. Additionally, Cox affirms that it behaves in a manner similar to Bayesian estimates, that is, it selects a value that is supported by the knowledge

accumulated from each executed proposition. Taking into account these advantages and because it is the most used method [27, 28], centroid or center of gravity method is used to defuzzify.

3.3. **Results.** Based on the set of inference rules the fuzzy inference system is capable of inferring all the attainable risk results for any KRFs combination. These results are best presented as a surface plot. Figure 6 is a three-dimensional depiction of the set of rules as a check on consistency. A number of inference methodologies exists for combining inputs and outputs. The method used in the proposed model is that of Mamdani [17] as it is considered the most appropriate.

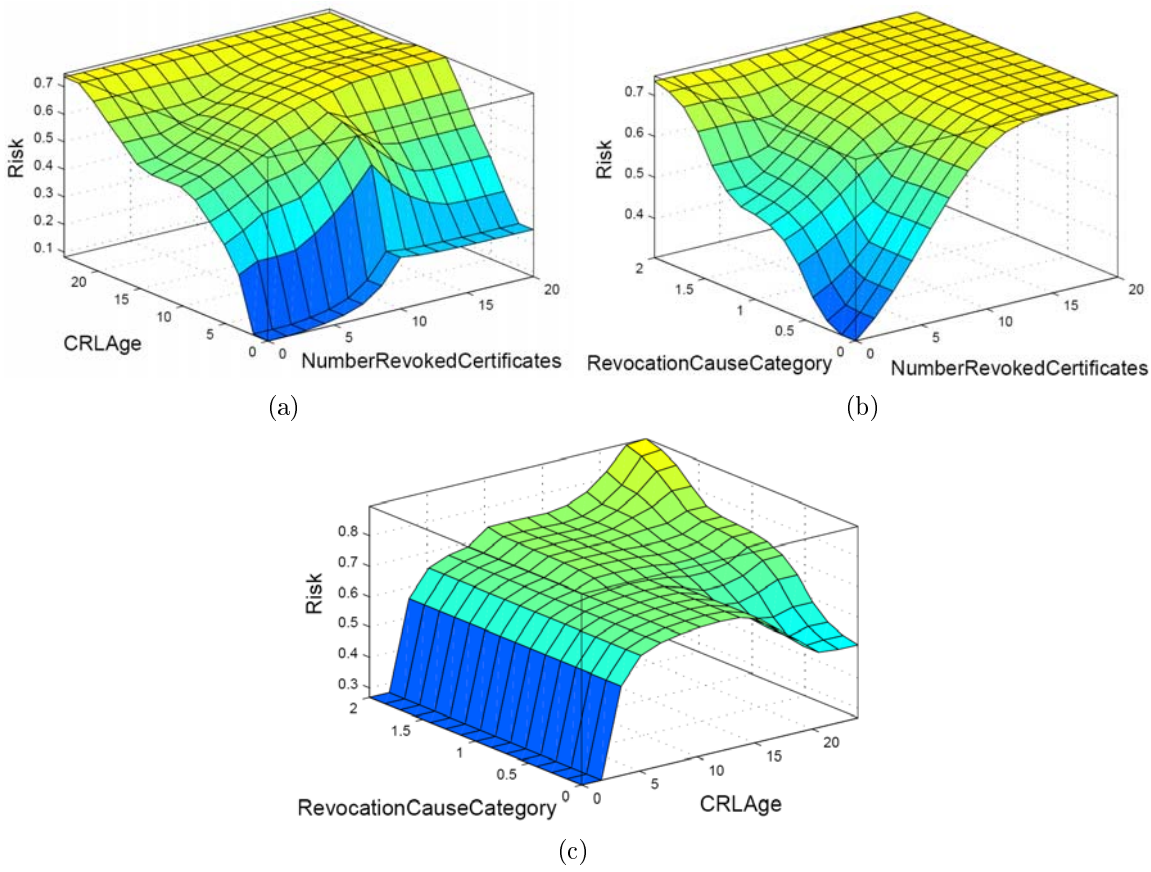


FIGURE 6. Risk indicator as a combination of (a) the CRL age and the number of revoked certificates, (b) the revocation cause categories and the number of revoked certificates, (c) the CRL age and the revocation cause categories

Figure 6, somewhat similar to a probability/severity chart, displays the nonlinear relation between the inputs and the risk indicator, where each combination of these KRFs results in a unique position on the surface. Intuitively, if a revocation event happening has a low (high) impact on the network and a low (high) probability, the risk yields a low (high) outcome, where intermediate results are also considered according to the knowledge base.

4. **A Case Study: GoDaddy.** Finally, to corroborate the benefits of the presented model, we analyze the case of a company that issues digital certificates. The selected company is GoDaddy, which is currently operating in the SSL market and it is the trusted

provider of Internet infrastructure services for the networked world that leads the global SSL marketplace with a 20.52% share [29]. Notice that though the collected data belongs exclusively to a single certification authority, the analysis of this case study is clearly representative because of the market share of the analyzed CA. Thus, the results obtained here can be extended to any other CA operating in the same market.

Using GoDaddy's Signing Certificate Revocation List [30], we analyzed a large sample of revoked certificates. Note that using this type of CRL, we are covering just one type of certificate: Code-Signing certificates. GoDaddy offers code-signing certificates for use by software developers and software vendors. The purpose of such a certificate is to sign code that users download off the Internet. By signing the code, users can be assured that the code has not been tampered with or corrupted since it was digitally signed with the private key of the software developer. In the online world, where people are not only becoming increasingly aware of security issues, but also worry about viruses and worms, signing the code provides a certain assurance to users that they are getting the software that they're expecting to get.

From each GoDaddy's CRL, we can obtain the three KRFs that our fuzzy inference model needs. To do so, we have to obtain the following parameters:

- Last Update instant of the CRL.
- Next Update instant of the CRL.
- Serial Number of each revoked certificate.
- Revocation Date of each revoked certificate.
- Revocation Code of each revoked certificate.

For the first KRF, we use the validity period of each CRL. This allows us to determine the range of the CRL Age. In the case of GoDaddy, CRLs are issued every 24 hours. Therefore, a CRL will be considered $VeryOld$ after this period of time.

For the second KRF, we need to determine the number of revoked certificates per day. For this purpose, we need the revocation date of each certificate and its serial number. With this information, we can analyze the time evolution of the number of revoked certificates per day (see Figure 7) and we can tally the number of revocation that occurred every day from 2009 to 2011.

Finally, the third KRF is also obtained from the CRL by means of the revocation code. Using the revocation code of each revoked certificate, we can calculate the impact of the revocation causes using the weights established in Table 1. In this context, Figure 8 shows the revocation causes of the certificates contained in GoDaddy's CRL. This analysis covered more than 300,000 certificates. It is worth noting, that the main cause

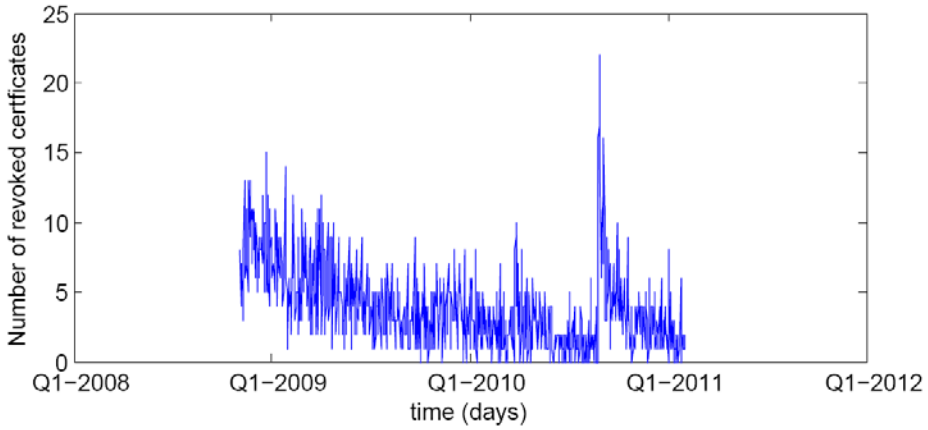


FIGURE 7. Number of revoked certificates evolution

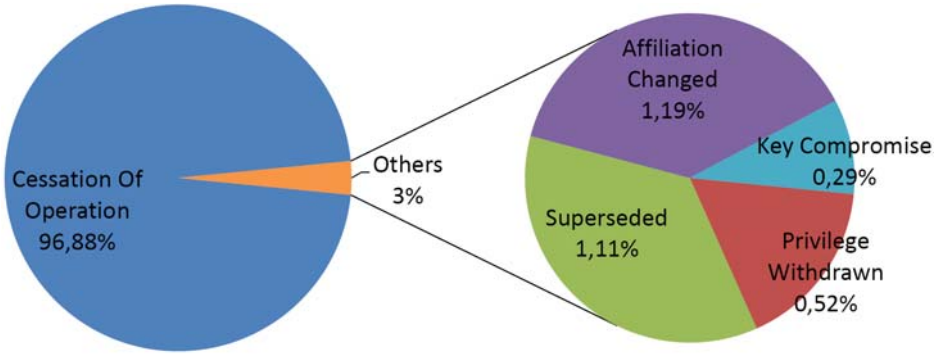


FIGURE 8. Revocation causes of code-signing certificates issued by GoDaddy

TABLE 2. Risk analysis score for ten days

Day	Number Revoked Cert	CRL Age	Revocation Category	Risk
24/01/2009	14	9 hours	Cat 1	0.686
22/04/2009	1	12 hours	Cat 2	0.523
21/05/2009	5	8 hours	Cat 2	0.567
18/05/2009	6	1 hour	Cat 3	0.112
25/08/2010	16	2 hours	Cat 2	0.253
27/08/2010	20	12 hours	Cat 2	0.748
16/09/2010	1	18 hours	Cat 1	0.424
28/09/2010	10	22 hours	Cat 3	0.892
22/10/2010	1	0.5 hours	Cat 1	0.0824
08/11/2010	5	10 hours	Cat 1	0.500

of revocation is the cessation of operation, i.e., the certificate is no longer needed for its original purpose. The rest of revocation causes are highly improbable compared with the main cause. Therefore, when a certificate is revoked by GoDaddy it is highly probable that the revocation is due to a cessation of operation.

Using these data, we choose 10 different days at different hours and we calculate the risk indicator using the proposed fuzzy inference system. Table 2 shows the results obtained. As expected, days with more revoked certificates involve higher risks. In a similar way, as the CRL becomes older the risk also increases. Finally, the category of the revoked certificates also affects significantly in the risk. However, as shown in Figure 8 the most common category is $Cat1$, as the predominant revocation cause has a weight $w_i = 1$.

For computing the previous risk outputs, we have used the Mamdani Method (as shown in Figure 9). For example, for 24/01/2009, we have obtained a scalar value of 0.686. This value is the result of using defuzzification with the centroid method. This method allows us to obtain a crisp output from the fuzzy output. As can be observed, in this particular case, the output value falls in the lower part of risk class defined as $Moderate$.

The previous results show that our risk analysis method is an effective and efficient way to assess the risk associated with the revocation system of GoDaddy and by extension to other SSL providers.

5. **Conclusions.** PKI requires a revocation mechanism to remove illegitimate users. This is commonly achieved using certificate revocation lists (CRLs). However, using CRLs presents a great challenge to users, as they have to make critical decisions based on the information contained in these lists but the information available is not always complete,

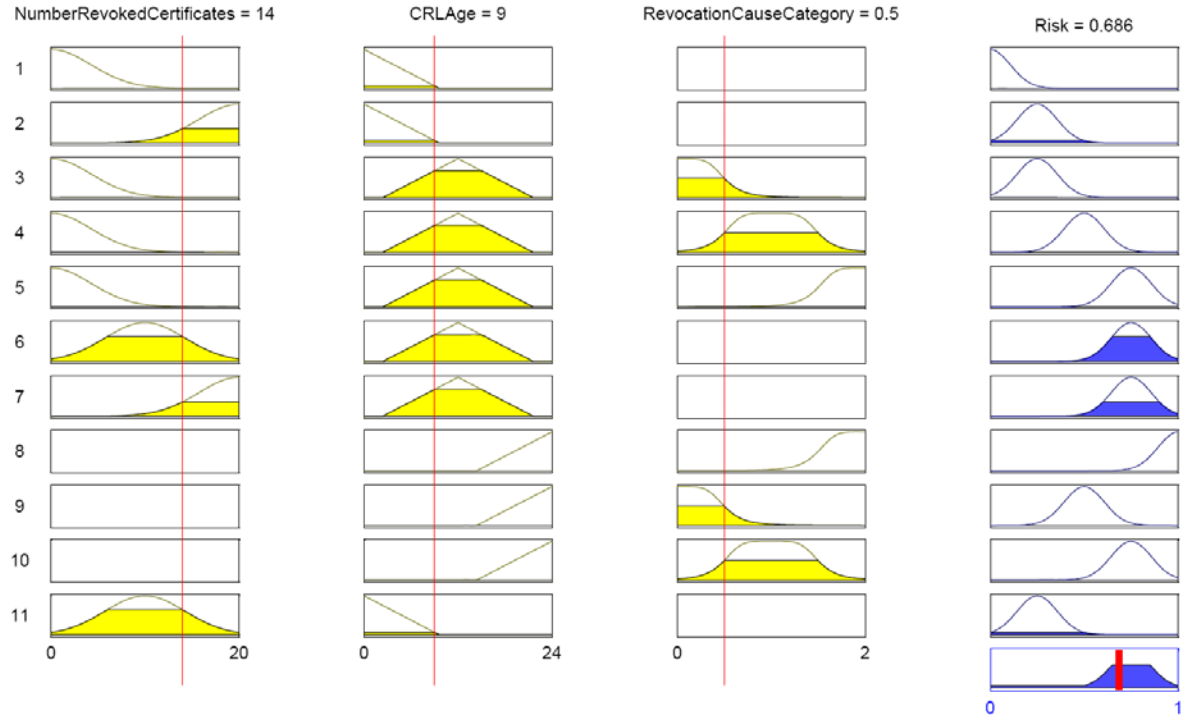


FIGURE 9. Mamdani method for computing the risk output (Jan 24 09:25:44 2009 GMT)

precise or updated. The key finding of this article is a systematic methodology to build a fuzzy system that models risk and assists the user in the decision making process related to certificate revocation. Our system not only considers the possibility of taking as valid a certificate that has been revoked but also other key risk factors (KRF). In this respect, we have identified potential risk sources involved in the revocation system and we have characterized them using fuzzy logic. The inputs given to the fuzzy system can be inferred from a standard CRL and as CRLs are accessible to any PKI user, in practice, everybody can take advantage of our fuzzy system. The output of our system is a measure of the risk of operating with a particular CRL at a given instant. Based on this output, the user can either decide whether to interact or not with another PKI user. Finally, a real certification authority (GoDaddy) has been analyzed using our fuzzy system. The results show that although this CA is issuing CRLs with a frequency of only one day, there is still an inherent risk that our model is able to measure.

Both academia and industry could benefit from our proposal. On the one hand, using the proposed fuzzy inference system, CAs could set policies to manage the issuance of CRLs to control the inherent risk of revocation within the PKI. CAs need to maintain an equilibrium between the costs of issuing a CRL, which increase significantly when a CRL is released frequently, and the risk, which tends to skyrocket if a CRL is not released in a timely manner. On the other hand, researchers could use our model to measure the risk of new revocation mechanisms that operate in novel environments such as vehicular networks or sensor networks.

As a final remark, we have to mention that our study has some limitations but probably each limitation provides an opportunity for further research. In first place, we assume that the CRL contains only one type of certificate. In practice, it could be the case that a CRL includes different types of certificates. Depending on the type of certificate, the involved risk changes. Thus, it could be interesting to analyze how the type of

certificate affects the risk-based decision making. Secondly, we have only analyzed the specific case of a PKI using CRLs as revocation mechanism. However, there are other revocation mechanisms such as the Online Certificate Status Checking Protocol (OCSP) that also involve risk when deployed. Analyzing the differences regarding the operational risk among the different revocation mechanisms could be another interesting area for future research.

Acknowledgment. This work is funded by the Spanish Ministry of Science and Education under the projects CONSOLIDER-ARES (CSD2007-00004) and TEC2011-26452 "SERVET", and by the Government of Catalonia under grant 2009 SGR 1362.

REFERENCES

[1] R. Housley, W. Polk, W. Ford and D. Solo, Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile, *RFC 3280, Internet Engineering Task Force*, 2002.

[2] N. Li and J. Feigenbaum, Nonmonotonicity, user interfaces, and risk assessment in certificate revocation (position paper), *Proc. of the 5th International Conference on Financial Cryptography, Lecture Notes in Computer Science*, vol.2339, pp.166-177, 2002.

[3] B. Fox and B. LaMacchia, Certificate revocation: Mechanics and meaning, *International Conference on Financial Cryptography, Lecture Notes in Computer Science*, vol.1465, pp.158-164, 1998.

[4] R. L. Rivest, Can we eliminate certification revocation lists? *International Conference on Financial Cryptography, Lecture Notes in Computer Science*, vol.1465, pp.178-183, 1998.

[5] P. McDaniel and A. Rubin, A response to can we eliminate certificate revocation lists, *Proc. of the 4th International Conference on Financial Cryptography, Lecture Notes in Computer Science*, vol.1962, pp.245-258, 2000.

[6] C.-I Hsu, C. Chiu and M. S.-H. Ho, The prediction of PKI security performance using PSO and bayesian classifier, *ICIC Express Letters*, vol.3, no.4(A), pp.1031-1036, 2009.

[7] J. L. Muñoz, O. Esparza, C. Gañán and J. Parra-Arnau, PKIX certificate status in hybrid MANETs, *WISTP, Lecture Notes in Computer Science*, vol.5746, pp.153-166, 2009.

[8] L. A. Zadeh, Fuzzy sets, *Information and Control*, pp.338-353, 1965.

[9] L. A. Zadeh, The calculus of fuzzy if/then rules, *Fuzzy Days'92*, pp.84-94, 1992.

[10] L. A. Zadeh, Commonsense knowledge representation based on fuzzy logic, *IEEE Computer*, pp.61-65, 1983.

[11] L. A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, *Information Science*, pp.301-357, 1975.

[12] L. A. Zadeh, The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets Syst.*, vol.11, pp.197-198, 1983.

[13] T. J. Ross, *Fuzzy Logic with Engineering Applications*, John Wiley & Sons, 2010.

[14] F. Eshragh and E. H. Mamdani, A general approach to linguistic approximation, *International Journal of Man-Machine Studies*, vol.11, pp.501-519, 1979.

[15] E. H. Mamdani and S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *Int. J. Hum.-Comput. Stud.*, vol.51, pp.135-147, 1999.

[16] E. H. Mamdani, Application of fuzzy logic to approximate reasoning using linguistic synthesis, *Proc. of the 6th International Symposium on Multiple-Valued Logic*, pp.196-202, 1976.

[17] E. H. Mamdani, Application of fuzzy algorithms for control of simple dynamic plant, *Proc. of the Institution of Electrical Engineers*, vol.121, no.12, pp.1585-1588, 1974.

[18] L. A. Zadeh, Fuzzy logic and its application to approximate reasoning, *IFIP Congress'74*, pp.591-594, 1974.

[19] R. Fullér, Introduction to neuro-fuzzy systems, *Advances in Soft Computing*, 2000.

[20] M. Sugeno, *Industrial Applications of Fuzzy Control*, Elsevier Science Inc., 1985.

[21] Y. Liao, C. Ma and C. Zhang, A new fuzzy risk assessment method for the network security based on fuzzy similarity measure, *The 6th World Congress on Intelligent Control and Automation*, vol.2, pp.8486-8490, 2006.

[22] D. Zhao, J Wang and J. Ma, Fuzzy risk assessment of the network security, *International Conference on Machine Learning and Cybernetics*, pp.4400-4405, 2006.

[23] A. S. Sendi, M. Jabbarifar, M. Shajari and M. Dagenais, Femra: Fuzzy expert model for risk assessment, *The 5th International Conference on Internet Monitoring and Protection*, pp.48-53, 2010.
[24] C. Hu and C. Lv, Method of risk assessment based on classified security protection and fuzzy neural network, *Asia-Pacific Conference on Wearable Computing Systems*, pp.379-382, 2010.
[25] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley and W. Polk, Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile, *RFC 5280, Internet Engineering Task Force*, 2008.
[26] S. Scandizzo, Risk mapping and key risk indicators in operational risk management, *Economic Notes*, vol.34, no.2, pp.231-256, 2005.
[27] E. Cox, *The Fuzzy Systems Handbook: A Practitioner's Guide to Building, Using, and Maintaining Fuzzy Systems*, Academic Press Professional, Inc., 1994.
[28] S. N. Sivanandam, S. Sumathi and S. N. Deepa, *Introduction to Fuzzy Logic Using MATLAB*, Springer-Verlag, New York, 2006.
[29] WhichSSL, *SSL Market Share*, http://www.whichssl.com/ssl-market-share.html, 2010.
[30] *Legal Repository from GoDaddy*, http://certs.godaddy.com/anonymous/repository.seam.

Toward Revocation Data Handling Efficiency in VANETs

Carlos Gañán, Jose L. Muñoz, Oscar Esparza
Jorge Mata-Díaz and Juanjo Alins

Universitat Politècnica de Catalunya (Departament Enginyeria Telemàtica)**
{carlos.ganan, jose.munoz, oesparza, jmata,juanjo}@entel.upc.es

Abstract. Vehicular Ad Hoc Networks (VANETs) require some mechanism to authenticate messages, identify valid vehicles, and remove misbehaving ones. A Public Key Infrastructure (PKI) can provide this functionality using digital certificates, but needs an efficient mechanism to revoked misbehaving/compromised vehicles. The IEEE 1609.2 standard states that VANETs will rely on the use of certificate revocation lists (CRLs) to achieve revocation. However, despite their simplicity, CRLs present two major disadvantages that are highlighted in a vehicular network: CRL size and CRL request implosion. In this paper, we point out the problems when using CRLs in this type of networks. To palliate these issues, we propose the use of Authenticated Data Structures (ADS) that allow distributing efficiently revocation data. By using ADS, network entities can check the status of a certificate decreasing the peak bandwidth load in the distribution points.

Keywords: Certification, PKI, Authenticated Data Structures.

1 Introduction

In the last decade, wireless communication between vehicles have drawn extensive attention for their promise to contribute to a safer, more efficient, and more comfortable driving experience in the foreseeable future. This type of communications have stimulated the emergence of Vehicular ad hoc networks (VANETs) which consist of mobile nodes capable of communicating with each other (i.e. Vehicle to Vehicle Communication -V2V communication) and with the static infrastructure (i.e. Vehicle to Infrastructure Communication -V2I communication). To make these communications feasible, vehicles are equipped with on-board units (OBUs) and fixed communication units (road-side units, RSUs) are placed

** This work is funded by the Spanish Ministry of Science and Education under the projects CONSOLIDER-ARES (CSD2007-00004) and TEC2011-26452 "SERVET", and by the Government of Catalonia under grant 2009 SGR 1362.

along the road. Applying short range wireless technology based on IEEE 802.11, multi-hop communication facilitates information exchange among network nodes that are not in direct communication range [1].

However, the open-medium nature of these networks and the high-speed mobility of a large number of vehicles make necessary the integration of primary security requirements such as authentication, message integrity, non-repudiation, and privacy [2]. Without security, all users would be potentially vulnerable to the misbehavior of the services provided by the VANET. Hence, it is necessary to evict compromised, defective, and illegitimate nodes. The basic solution envisioned to achieve these requirements is to use digital certificates linked to a user by a trusted third party. These certificates can then be used to sign information. Most of the existing solutions manage these certificates by means of a central Certification Authority (CA) [3]. According to IEEE 1609.2 standard [4], vehicular networks will rely on the public key infrastructure (PKI). In PKI, a CA issues an authentic digital certificate for each node in the network. Therefore, an efficient certificate management is crucial for the robust and reliable operation of any PKI. A critical part of any certificate-management scheme is the revocation of certificates.

Regarding the revocation of these certificates, some proposals allow revocation without the intervention of the infrastructure at the expense of trusting other vehicles criteria; and other proposals are based on the existence of a central entity, such as the CA, which is in charge of taking the revocation decision for a certain vehicle. Again, according to the IEEE 1609.2 standard [4], vehicular networks will rely on the existence of a CA. In this sense, it is stated that these networks will depend on certificate revocation lists (CRLs) and short-lived certificates to achieve revocation. CRLs can be seen as black lists that enumerate revoked certificates along with the date of revocation and, optionally, the reasons for revocation.

As the network scale of VANETs is expected to be very large and to protect the privacy of users each vehicle has many temporary certificates (or called pseudonyms), the CRLs are expected to be quite large. Moreover, CRLs have also associated a problem of request implosion, i.e., vehicles may become synchronized around CRL publication instant, as they may request CRL at or near the moment of publication. This burst of requests may cause network congestion that may introduce longer latency in the process of validating a certificate. To reduce the potential network and computational overhead imposed by any CRL distribution mechanism, some optimizations for organizing, storing, and exchanging CRL information have been proposed. In [2, 5], it is proposed a way of

compress CRLs using Bloom filters. Their method reduces the size of a CRL by using about half the number of bytes to specify the certificate serial number for revocation. However, the use of this probabilistic structure has associated a false positive rate that diminishes the efficiency of the revocation service.

In this paper, we explore the benefits of using authenticated data structures (ADS), such as binary trees or skip lists, to manage revocation data in VANETS. These structures are a model of computation where untrusted responders answer certificate status queries on behalf of the CA and provide a proof of the validity of the answer to the user. Although VANETs can greatly benefit from the use of ADSs, to the best of our knowledge there has been no proposal of deploying the revocation service by means of an ADS. By using these structures, both CRL issues are palliated: the CA is no longer a bottleneck as there are several responders that act on its behalf; and the revocation data can be checked without downloading the whole CRL.

2 CRLs' problematic in VANETs

As stated in the trial-use standard [4], for a certificate authority (CA) to invalidate a vehicle's certificates, the CA includes the certificate serial number in the CRL. The CA then distributes the CRL so that vehicles can identify and distrust the newly revoked vehicle. The distribution should spread quickly to every vehicle in the system.

However, the distribution itself poses a great challenge due to the size of the CRL. As a CRL is a list containing the serial numbers of all certificates issued by a given certification authority (CA) that have been revoked and have not yet expired, its distribution causes network overhead. Moreover, the CRL size increases dramatically if only a small portion of the OBUs in the VANET is revoked. To have an idea of how big the CRL size can be, consider the case where 1% of the total number of the OBUs in the United States is revoked. Recall that in a VANET, each vehicle owes not only an identity certificate, but also several pseudonyms. The number of pseudonyms may vary depending on the degree of privacy and anonymity that it must be guaranteed. According to Raya, Papadimitratos, and Hubaux in [5] the OBU must store enough pseudonyms to change pseudonyms about every minute while driving. This equates to about 43,800 pseudonyms per year for an average of two hours of driving per day. In the United States alone, 255,917,664 "highway" registered vehicles were counted in 2008, of which 137,079,843 passenger cars [6]. In

this case, the CRL would contain around 100 billion revoked certificates. Assuming that certificates can be identified by a 16 byte fingerprint (the size of one AES block), the CRL size would be of 1,7 TB approximately. Only the amount of memory necessary to storage this CRL makes it impossible its deployment. Therefore, the CRL size has to be reduced.

The CRL size can be reduced by using regional CAs. However, there appears a trade-off between the size of the CA region and size of the CRL, as well as the management complexity of the entire PKI system for VANETs. The least complicated region to manage would be a single large area, such as the entire United States, with a single CA responsible for every certificate and pseudonym. However, this gives place to CRLs of several terabytes. Therefore, it is necessary to divide the CRL information according to regional areas. In this sense, if we divide the entire United States by cities (i.e. 10,016 cities according to the U.S. census bureau), the CRL size is reduced to around 170 Mbytes. Using the 802.11a protocol to communicate with RSUs in range, vehicles could have between 10-30 Mbps depending on the vehicle's speed and the road congestion. Therefore, in the best case a vehicle will need more than 45 seconds to download the whole CRL. Under non-congested conditions, any vehicle should be able to contact the infrastructure for more than 45 seconds, and therefore download the CRL. In scenarios where vehicles are not able to keep a permanent link with the infrastructure for this amount of time, techniques such as Bloom filter or Digital Fountain Codes could be used to download the CRL. Therefore, though the problem of having a huge CRL is mitigated by the use of such techniques, the restraints imposed by the distribution affect the freshness of the revocation data.

A direct consequence of this significant time to download a CRL is that a new CRL cannot be issued very often, so its validity period has to be shortened. This validity period directly determines how often a vehicle has to update the revocation information. Therefore, the validity period of the CRL is critical to the bandwidth consumption. In this context, it appears another trade-off between the freshness of revocation information and the bandwidth consumed by downloading CRLs. Large validity periods will decrease the network overhead at expenses of having outdated revocation information. Small validity periods will increase the network overhead but users will have fresh information about revoked certificates. As CRLs cannot be issued every time there is a new revoked certificate, vehicles will be operating with revocation information that is not comprehensive. Therefore, they will be taking certain risk of trusting a certificate that could be potentially revoked.

3 Using Authenticated Data Structures for certificate revocation in VANETs

By replicating revocation data at untrusted responders near users, VANETs can enhance its performance but that replication causes a major security challenge. Namely, how can a vehicle verify that the revocation data replicated at the RSUs are the same as the original from the CA? A simple mechanism to achieve the authentication of replicated revocation data consists of having the digitally sign each revocation entry and replicating the CA signature too. However, in VANETs where the revocation data evolves rapidly over time, this solution is inefficient. To achieve higher communication and computation efficiency, we propose the use of authenticated data structures (ADS) to handle the revocation service in VANETs. ADSs are a model of computation where untrusted responders answer certificate status queries on behalf of the CA and provide a proof of the validity of the answer to the user. In this section, first we introduce the architecture necessary to adopt ADSs. Then, we describe different ADSs and their main benefits.

3.1 System Architecture

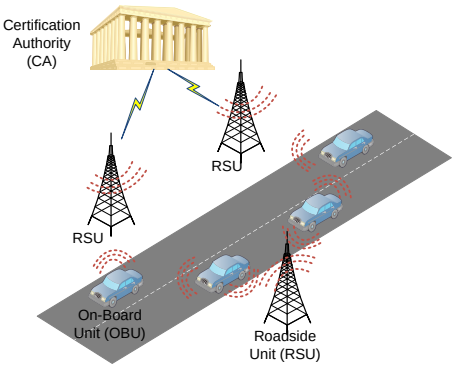


Fig. 1. System Architecture.

The system architecture to support ADSs consists in an adaptation of a PKI system to the vehicular environment. The ADS model involves a structured collection $\mathcal{R}$ of revoked certificates and three parties: the certification authority (CA), the road side units (RSUs), and the vehicles. A repertory of query operations and optional update operations are

assumed to be defined over $\mathcal{R}$. These three parties present a hierarchical architecture (see Fig. 1) which consists of three levels: the CA is located at level 1, as it is the top of the system. RSUs are located at level 2. Finally, the on-board units (OBUs) are located at the bottom of the hierarchy. Note that without loss of generality we consider a single the central trusted authority at the root, but it could be further divided into different state level trusted authorities and additionally a group of city level trusted authorities can be placed under every state authority.

The main tasks of each entity are:

1. The CA is responsible for generating the set of certificates that are stored in each OBU. It is also responsible for holding the original version of $\mathcal{R}$ and making it accessible to the rest of the entities. By definition of TTP, the CA should be considered fully trusted by all the network entities, so it should be assumed that it cannot be compromised by any attacker. In fact, in our proposal the CA is the only trusted entity within the network. Whenever an update is performed on $\mathcal{R}$, the CA produces structure authentication information, which consists of a signed time-stamped statement about the current version of $\mathcal{R}$.
2. RSUs are fixed entities that are fully controlled by the CA. They can access the CA anytime because they are located in the infrastructure-side, which does not suffer from disconnections. RSUs maintain a copy of $\mathcal{R}$. They interact with the CA by receiving from the CA the updates performed on $\mathcal{R}$ together with the associated structure authentication information. RSUs also interact with vehicles by answering queries on $\mathcal{R}$ posed by the vehicles. In addition to the answer to a query, RSUs also return answer authentication information, which consists of (i) the freshest structure authentication information issued by the CA; and (ii) a proof of the authenticity of the answer. If the CA considers that an RSU has been compromised, the CA can revoke it.
3. OBUs are in charge of storing all the certificates that a vehicle possesses. An OBU has abundant resources in computation and storage and allows any vehicle to communicate with the infrastructure and with any other vehicle in its neighborhood. OBUs pose queries on $\mathcal{R}$, but instead of contacting the CA directly, it contacts the RSU in range. However, OBUs only trust the CA and not the RSU about $\mathcal{R}$. Hence, it verifies the answer from the RSU using the associated answer authentication information.

3.2 System Requirements

- *Low computational cost*: The computations performed internally by each entity (CA, RSU, and OBU) should be simple and fast.
- *Low communication overhead*: CA-to-RSU communication (update authentication information) and RSU-to-OBU communication (answer authentication information) should be as small as possible.
- *High security*: the authenticity of the answers given by a RSU should be verifiable.

3.3 Authenticated Data Structures

Several ADSs have been proposed in the literature (mainly in the context of data base management) that fulfill the aforementioned requirements. In this section, we describe a repertoire of ADSs and to what extent they are capable of improving the revocation service.

Merkle Hash trees A Merkle hash tree (MHT) [7] is essentially a tree structure that is built with a collision-resistant hash function to produce a short cryptographic description of $\mathcal{R}$. The leaf nodes hold the hash values of the data of interest, i.e., the serial number of the revoked certificates (SN_1,SN_2,...,SN_n); and the internal nodes hold the hash values that result from applying the hash function to the concatenation of the hash values of its children nodes. In this way, a large number of separate data can be tied to a single hash value: the hash at the root node of the tree. MHTs can be used to provide an efficient and highly-scalable way to distribute revocation information.

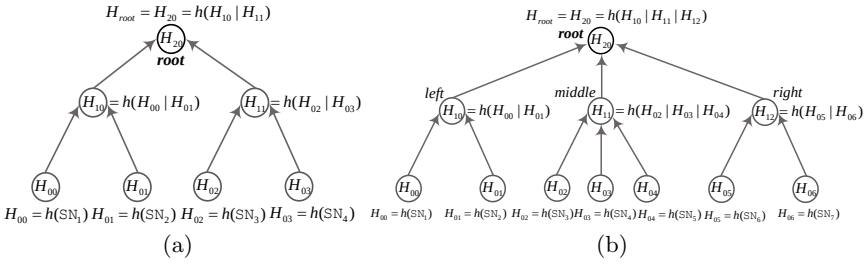


Fig. 2. Sample trees (a) MHT (b) 2-3.

A sample MHT is presented in Figure 2(a). The authentication of an element is performed using a verification path, which consists of the

sibling nodes of the nodes on the path from the leaf associated with the element to the root of the tree. The root value is signed and the collision-resistant property of the hash function is used to propagate authentication from the root to the leaves. This construction is simple and efficient and achieves signature amortization, where only one digital signature is used for signing a large collection of data. The hash tree uses linear space and has $O(log\ n)$ (where n denotes the number of revoked certificates) proof size, query time and verification time. An ADS based on hash trees can also achieve $O(log\ n)$ update time.

2-3 trees A standard 2-3 tree [8] is a tree where all leaves are at the same height and each node (except leaves) has two or three children. It has the nice property that leaf removal and insertion incur only logarithmic complexity because these operations only involve the nodes related to the path from the relevant leaf to the root.

Each leaf of such a 2-3 tree stores an element of set $\mathcal{R}$, and each internal node stores a one-way hash of its children's values. Thus, the CA-to-RSU communication is reduced to $O(1)$ entries, since the CA sends insert and remove instructions to the RSUs, together with a signed message consisting of a timestamp and the hash value of the root of the tree. RSUs respond to a membership query for an element SN_i as follows: if SN_i is in $\mathcal{R}$, then RSUs provide the path from the leaf storing SN_i to the root, together with all the siblings of the nodes on this path; else (SN_i is not in$\mathcal{R}$), RSUs provide the leaf-to-root paths from two consecutive leaves storing SN_j and SN_k such that $j < i < k$, together with all siblings of the nodes on these paths. By tracing these paths, OBUs can recompute the hash values of their nodes, ultimately recomputing the hash value for the root, which is then compared against the signed hash value of the root for authentication. As with MHTs, these trees achieve $O(log\ n)$ proof size, query time, update time and verification time.

One-way accumulator One-way accumulator (OWA) functions [9] allow a CA to digitally sign a collection of objects as opposed to a single one. The main advantage of this approach is that the validation of a response takes constant time and requires computations simple enough to be performed in resource-constrained devices. This type of ADS achieves a tradeoff between the cost of updates at the CA and queries at the RSUs, with updates taking $O(k + log(\frac{n}{k}))$ time and queries taking $O(\frac{n}{k})$ time, for any fixed integer parameter $1 \leq k \leq n$. For instance, one can achieve $O(\sqrt{n})$ time for both updates and queries.

Skip Lists Skip lists [10] are probabilistic ADSs that provide an alternative to balanced tree. Skip lists are sorted linked lists with extra links, designed to allow fast search in $\mathcal{R}$ by taking "shortcuts". The main idea is to enhance linked lists, which connect each element in the data sequence to its successor, by also connecting some elements to successors further down the sequence. Roughly half of the elements have links to their two-hop successor, roughly a quarter of the elements have links to their four-hop successor, and so on. As a result, during traversal from SN_i to element SN_j, the traversal path follows repeatedly the longest available link from the current element that does not overshoot the destination SN_j, and thereby reaches SN_j in fewer steps than would be possible by just traversing every intervening element between SN_i and SN_j. Compared with balanced trees, a skip list presents the following benefits:

- It is easy to implement and practically efficient in search, especially update time.
- It is space compact, where space is allocated when needed, while empty space is preserved in balanced tree.
- It is main memory index, while balanced tree are disk-based index.

Finally, Table 1 shows a comparison of the asymptotic performance of the main ADS versus traditional revocation mechanisms such as CRL or OCSP. Note that with ADSs, the revocation service can be greatly improved both in computation and communication overhead.

method	space	update time	update size	query time	query size	verifying time
CRL	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$
OCSP	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(1)$
MHT	$O(n)$	$O(log\ n)$	$O(1)$	$O(log\ n)$	$O(log\ n)$	$O(log\ n)$
2-3 tree	$O(n)$	$O(log\ n)$	$O(1)$	$O(log\ n)$	$O(log\ n)$	$O(log\ n)$
Skip Lists	$O(n)$	$O(log\ n)$	$O(1)$	$O(log\ n)$	$O(log\ n)$	$O(log\ n)$
OWA	$O(n)$	$O(k + log(\frac{n}{k}))$	$O(k)$	$O(\frac{n}{k})$	$O(1)$	$O(1)$

Table 1. Comparison of the main ADS vs traditional revocation mechanisms.

3.4 Certificate Status Validation Protocol

The certificate status validation protocol consists in three stages.

1. *Revocation Service Setup*: The CA creates a CRL by appending the serial number of any revoked certificate. Then, it computes the corresponding ADS from the set $\mathcal{R}$ of revoked certificates contained in the

CRL. Once the ADS is computed, the CA signs the resulting time-stamped digest of the data structure, i.e., a collision resistant succinct representation of the data structure. The digest is transmitted to all the RSUs via a secure wireline together with the corresponding CRL. RSUs can either implement a push or pull protocol to transmit the digest to the vehicles in range.

2. *Certificate Status Updating*: Depending on the CA's policy, when an update is necessary, the CA recomputes the ADS and generates a new signed digest that is transmitted to the RSUs. Note that depending on the ADS, the data structure should be computed again or only update and delete operations should be performed. The new ADS is transmitted to the RSUs again, so that they could answer to validation queries.
3. *Certificate Status Querying*: OBUs query any RSU in range about the status of a particular certificate (SN_i). If $SN_i \in \mathcal{R}$, then the RSU computes the path necessary to allow OBUs to compute the digest and check that it matches the signed digest. If $SN_i \notin \mathcal{R}$, then the RSU computes the path of two consecutive certificates in $\mathcal{R}$ and transmit them to the requesting OBU. This OBU can then recompute the digest for both revoked certificates and be sure that $SN_i \notin \mathcal{R}$.

4 Evaluation

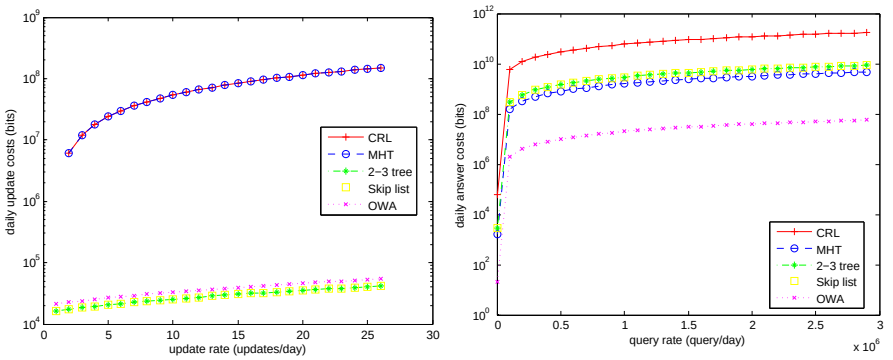
In the following, we compare the communication costs of using ADSs with the tradition CRL mechanism. To that end we define a set of parameters (see Table 2).

Parameter	Meaning of the parameter
N	Total number of certificates ($n = 3,000,000$)
k	Average number of certificates handled by a CA ($k = 30,000$)
p	Percentage of revoked certificates($p = 0.1$)
q	Number of certificate status queries issued per day ($q = 3,000,000$)
T	Number of updates per day ($T = 1$)
s_{SN}	Size of a serial number ($s_{SN} = 20$)
s_{sig}	Size of a signature ($s_{sig} = 1,000$)
s_{hash}	Size of the hash function ($s_{hash} = 128$).

Table 2. Notation

Using this notation, the CRL daily update cost is $T \cdot n \cdot p \cdot s_{SN}$ as each CA sends the whole CRL to the corresponding RSUs in each update.

The CRL daily query cost is $q \cdot p \cdot k \cdot s_{SN}$ as for every query the RSU sends the whole CRL to the querying OBU. When using ADS, these costs are drastically reduced. Note that no matter the type of ADS, OBUs do not have to download the whole CRL, and they only download status information about the certificate they want to operate with. Regarding MHTs, the RSUs have to recompute the tree in each update, so that daily update cost is $T \cdot n \cdot p \cdot s_{SN}$. However, to answer an OBU's query the RSU only needs to send up to $1 + log_2(pk)$ numbers, resulting in $q \cdot s_{hash}(1+log_2(pk))$ bits. In the case of 2-3 trees, to update the directory, the CA sends difference lists of total daily length of $\frac{n \cdot p \cdot s_{SN}}{365} + T \cdot s_{sig}$; and answer to OBUs' queries results in $2 \cdot q \cdot s_{hash} \cdot log_2(pk)$ bits. Similarly, skip lists need $2log_2\lceil pk \rceil$ number to answer an OBU's query and the same update cost than the 2-3 tree. With OWAs, the size of answer are drastically reduced to roughly s_{sig}, and the update cost depends on the accumulator configuration. We use Matlab R2011b to evaluate these costs.



(a) (b)

Fig. 3. (a) Daily CA-to-RSU update costs vs. update rate, (b) RSU-to-OBU query cost vs query rate.

Note that the costs will vary mainly depending on the total number of revoked certificates, the update rate and the number of queries. Figure 3(a) shows how the CA-to-RSU update communication costs of the different revocation mechanisms depend on the update rate (all other parameters are held constant). Note that any ADS is much more robust and efficient than CRL, even allowing once per hour updates. Regarding the query costs, as ADSs have smaller proof to validate the status of a

certificate they provide a more bandwidth efficient solution than CRL (see Fig. 3(b)).

5 Conclusions

In this paper, we consider the problem of certificate authentication and revocation in VANETs. We have proposed the use of authenticated data structures to handle the revocation service over VANETs. After discussing the issues of deploying CRLs in these environments, we show that ADSs are more robust to changes in parameters, and allow higher update/query rates than traditional revocation mechanisms. In addition, the adoption ADS reduces both the communication and the computational overhead in the OBUs. For our future work, we will investigate the use of mobile repositories under the context of the proposed schemes.

References

1. D. Jiang and L. Delgrossi. IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2036–2040, May 2008.
2. Maxim Raya and Jean-Pierre Hubaux. The security of vehicular ad hoc networks. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, SASN '05, pages 11–21, 2005.
3. P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya. Architecture for secure and private vehicular communications. In *Telecommunications, 2007. 7th International Conference on ITS*, pages 1 –6, June 2007.
4. IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages. *IEEE Std 1609.2-2006*, pages 1–105, 2006.
5. Jason J. Haas, Yih-Chun Hu, and Kenneth P. Laberteaux. Design and analysis of a lightweight certificate revocation mechanism for vanet. In *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking*, VANET '09, pages 89–98, New York, NY, USA, 2009. ACM.
6. Bureau of Transportation Statistics U.S. Department of Transportation. Number of u.s. aircraft, vehicles, vessels, and other conveyances. `http://www.bts.gov/publications/national_transportation_statistics/html/table_01_11.html`, 2009. [Online; accessed 31-July-2011].
7. R.C. Merkle. A certified digital signature. In *Advances in Cryptology (CRYPTO89). Lecture Notes in Computer Science*, number 435, pages 234–246. Springer-Verlag, 1989.
8. M. Naor and K. Nissim. Certificate Revocation and Certificate Update. *IEEE Journal on Selected Areas in Communications*, 18(4):561–560, 2000.
9. Josh Benaloh and Michael de Mare. One-way accumulators: a decentralized alternative to digital signatures. In *Workshop on the theory of cryptographic techniques on Advances in cryptology*, EUROCRYPT '93, pages 274–285, 1994.
10. William Pugh. Skip lists: a probabilistic alternative to balanced trees. *Commun. ACM*, 33:668–676, June 1990.

Impact of the revocation service in PKI prices

Carlos Gañán, Jose L. Muñoz, Oscar Esparza
Jorge Mata-Díaz and Juanjo Alins

Universitat Politècnica de Catalunya (Departament Enginyeria Telemàtica)**
{carlos.ganan, jose.munoz, oesparza, jmata,juanjo}@entel.upc.es

Abstract. The ability to communicate securely is needed for many network applications. Public key infrastructure (PKI) is the most extended solution to verify and confirm the identity of each party involved in any secure transaction and transfer trust over the network. One of the hardest tasks of a certification infrastructure is to manage revocation. Research on this topic has focused on the trade-offs that different revocation mechanisms offer. However, less effort has been paid to understand the benefits of improving the revocation policies. In this paper, we analyze the behavior of the oligopoly of certificate providers that issue digital certificates to clients facing identical independent risks. We found the prices in the equilibrium, and we proof that certificate providers that offer better revocation information are able to impose higher prices to their certificates without sacrificing market share in favor of the other oligarchs. In addition, we show that our model is able to explain the actual tendency of the SSL market where providers with worst QoS are suffering loses.

Keywords: PKI pricing, SSL certificates, CRLs.

1 Introduction

Nowadays, there is a wide range of technology, products and solutions for securing electronic infrastructures. As with physical access security, the levels of security implemented should be commensurate with the level of complexity, the applications in use, the data in play, and the measurement of the overall risk at stake. A consensus has emerged among technical experts and information managers in government and industry that Public Key Infrastructure (PKI) offers the best feasible solution to these issues. PKI [1] has been a popular, yet often reviled technology since its adoption in the early nineties.

Currently deployed PKIs rely mostly on Certificate Revocation Lists (CRLs) for handling certificate revocation [2]. Although CRLs are the most widely used way of distributing certificate status information, much research

** This work is funded by the Spanish Ministry of Science and Education under the projects CONSOLIDER-ARES (CSD2007-00004), FPU grant AP2010-0244, and TEC2011-26452 "SERVET", and by the Government of Catalonia under grant 2009 SGR 1362.

effort has been put on studying other revocation distribution mechanisms in a variety of scenarios [3, 4]. These studies aim to compare the performance of different revocation mechanisms in different scenarios. However, none of these studies have explicitly modeled the interaction among CAs. In this paper, we model this interaction by using a game-theoretic approach.

With the appearance of novel network environments (e.g VANET or MANET), the quantity of CAs in the SSL certificate market is becoming larger and the market concentration diminishes, but it is not simple to eliminate the oligopoly in the short-term. During the 90s, the certification market, the competition among CAs appears mainly as price competition. In this situation, malignant price competition would be detrimental to the interests of the users and lead to the CA's pay crisis. Facing the situation, the main CAs have begun to change the competitive strategies from basic price competition to price and quality of services (QoS) competition. To provide better QoS, CAs have to improve their revocation service, and specifically the freshness of the CRLs. Users will pay more for a service that issues certificate status information faster. Time-to-revocation metric is visible to costumers by checking the CA's repositories where they publicize the revocation information.

The model of this article deals with an oligopoly of CAs which compete in certificate prices and QoS, and do not know the certificate revocation probability in the next interval for sure. The assumption that the revocation probability is *ex-ante* uncertain is quite logical and intuitive. The number of revoked certificates vary with time and in a manner that cannot predictable with certainty. We show that an uncertain revocation probability introduces a systematic risk that does not decrease by selling more certificates. If CAs are risk averse, this effect relaxes price competition. The equilibrium characteristic of the certification market is found by establishing a price competition model with different QoS. We consider that there are diversities in the certification service quality, and we describe factors that affect the service quality such as the CRL lifetime. By combining the characteristics of the certification market and considering the conveniences of modeling, two key parameters are selected to measure the QoS and a duopoly price competition model with service quality differentiation is established.

2 Related Work

Although PKI has been a widely adopted solution for many years now, very few works have dealt with the impact of the revocation mechanism in the prices CAs offer. Most of the literature [4, 5], intend to optimize the revocation mechanism to minimize the overhead or to improve the reliability. However, the most extended revocation mechanism is still CRL. Authors in [6] analyze the revocation mechanisms based on based on empirical data from a local

network. They conclude that the freshness of the revocation data depends on how often the end entities retrieve the revocation information but the bandwidth cost is high if end entities retrieve the revocation lists often.

Ma *et al.* in [7] propose a series of policies that certification authorities should follow when releasing revocation information. According to this study, a CA should take different strategies when providing certificate services for a new type of certificates versus a re-serving type of certificates. Authors give the steps by which a CA can derive optimal CRL releasing strategies and they prove that a CA should release CRLs less frequently in the case that the fixed cost is higher, the variable cost is higher, the liability cost is lower, or the issued age of certificates is shorter. Similarly authors in [8] authors address the CRL release problem through a systematic and rigorous approach which relies on a mix of empirical estimation and analytical modeling. They propose four different models which seek to exploit the variation in certificate specific properties to provide guidance to the CA in determining the optimal CRL release intervals and the associated costs. However, none of these works neither analyze the impact of CRLs releasing policies in the prices that the CA charges nor model the interaction among CAs. In this paper, we address these issues using a game theoretic approach.

3 Modeling the Certificate Provider Competition

To formalize our arguments we describe a model of the certificate market with profit-maximizing certification authorities and a continuum of network users. When a user requests the status of given certificate, the CA does not always provides the most updated information but a pre-signed CRL [4, 5]. In this context, the CA will bear the liability cost due to any damage that may occur between the revocation of a certificate and the release of the CRL.

3.1 Demand for certificates

We consider an oligopoly of A CAs, indexed by $i = 1, \cdots, A-1$, and N users in the economy, where N is large relative to A.Each user has the same strictly concave expected utility function and faces the risk to lose l when using a revoked certificates. The probability π of operating with a revoked certificate is equal for each user in the network, and conditional on π operating with revoked certificates of different users are statistically independent. This probability is out of the user's control so that no moral hazard problem arises. Except for their probabilities of operating with revoked certificates, individuals are assumed to be identical. However, π is not known *ex-ante* with certainty but is a random variable distributed on $[\underline{\pi}; \overline{\pi}]$ with cumulative

density function $F(\pi)$. Each user has an initial wealth $w > 0$. When operating with a revoked certificate, users may suffer a loss. We assume that the user's wealth exceeds the potential loss, that is, $l \leq w$.

Users can purchase different certificate types from the CA with different revocation updating service. We characterize this product by the price of the certificate $P_i > 0$ and an indemnity $C_i > 0$ the CA pays to the user if it suffers from an attack and operates with another user whose certificate was revoked. Note that as CRLs are not issued each time a certificate is revoked but periodically, users will be operating with outdated information. Let (P_i, C_i, t_i, s_i) be a certificate contract offered by CA_i which specifies the price P_i to be paid by an user and the level of coverage C_i paid to the user if an attack takes place and she operates with a revoked certificate. Let t_i represent the CRL updating interval, and s_i represent the security level.

Let us assume that the total utility U which users can get after they purchase a certificate consists of two parts. The first part is wealth utility which represented by U_w the other part is QoS utility which the applicant can get after they obtained the CA's services, represented by U_{QoS}. The total utility U is defined as:

$$U(P_i, C_i, t_i, s_i) = \alpha_1 U_w + \alpha_2 U_{QoS}, \forall \alpha_k \in [0,1] \text{ and } \sum \alpha_k = 1;\ k = 1, 2. \tag{1}$$

where α_i represents the significance level of U respectively.

On the one hand, we calculate the wealth utility. If no attack due to misuse of a revoked certificate happens after the user has purchase the service the CA, a user gains $w - P_i$, on the contrary a user gains $w - P_i + C_i$. We assume that all users have same loss with two-point distribution:

$$\mu = (w - P_i)(1 - \pi) + (w - P_i + C_i)\pi \ = w - P_i + \pi C_i, \tag{2}$$

$$\sigma^2 = \pi(1 - \pi)C_i^2. \tag{3}$$

Hence we can characterize the wealth utility by the mean and variance of Eq. (2) and Eq. (3) respectively. Thus, we can define U_w as a mean-variance utility function:

$$U_w(P_i, C_i) = \mu - R\sigma^2, \tag{4}$$

where R represents the Arrow-Pratt index of absolute risk aversion. This means that the larger R is, the more risk averse the user is and the smaller U_w is.

On the other hand, let U_{QoS} be a linear function of the QoS that the CA offers. Thus, we define U_{QoS} as:

$$U_{QoS}(t_i, s_i) = \pi\theta \left(\beta_1 s_i + \beta_2 \frac{1}{t_i}\right), \forall \beta_k \in [0,1], \sum \beta_k = 1 \text{ and } \theta > 0;\ k = 1, 2. \tag{5}$$

where θ represents the quality preference parameter of the user, and β_1 represents the user's preference to security level and β_2 represents the user's

preference to CRL issuing interval. Note that the higher the level of security the CA provides, the larger U_{QoS} is; the longer the CRL updating interval is, the smaller U_{QoS} is. It is also worth noting that θ is unknown to the CAs a priori.

In order to calculate the total utility of the user, we must unify the dimension of the security level and the CRL updating interval. Thus, using (1),(4) and (5) the total utility is calculated as:

$$U(P_i, C_i, t_i, s_i) = \alpha_1[w - P_i - \pi C_i - R\pi(1-\pi)C_i^2] + \alpha_2\left[\pi\theta\left(\beta_1 s_i + \beta_2 \frac{1}{t_i}\right)\right]. \quad (6)$$

Note that according to this expression, users are willing to pay higher prices for those certificates whose issuer provides a better QoS. Note that issuing certificate status information faster, highly increases the QoS of the revocation service. Thus, certificates linked to a better revocation service provide more utility to the user.

3.2 Supply of certificates

We consider an oligopoly of CAs operating in the certification market. CAs compete for users by offering certificates and CRLs. The service qualities of their CA products are also different. The level of service quality is mainly shown by the CRL updating interval and the security level[1].

When choosing a CA, a user takes into account several factors. Our goal is to gauge the impact of the revocation service on the certificate prices. However, it should be noted that, for convenience, many website owners choose the registrar's authority regardless of the price. Before issuing a certificate, the CA verifies that the person making the request is authorized to use the domain. The CA sends an email message to the domain administrator (the administrative or registrant contact, as listed in the Whois database) to validate domain control. If there is no contact information in the Whois database or the information is no longer valid, the customer may instead request a Domain Authorization Letter from his/her registrar and submit the letter to the CA as proof of his/her domain control. If the administrative/registrant contact fails to approve the certificate request, the request is denied. This authentication process ensures that only an individual who has control of the domain in the request can obtain a certificate for that domain. Therefore as CAs compete by quoting a certificate price which has associated a particular quality of service, we have Bertrand competition. The CA that quotes the lowest certificate price with the highest QoS sells to all users.

[1] Note that additional QoS parameters could be introduced in the model. In fact, CAs distinguish themselves by offering additional value-added services (e.g. GoDaddy bundling domain registration with certificate issuance), turn-around time, etc.

4 Equilibrium Certificate Providers

In this section we consider the certification industry with an oligopoly of A certification authorities and analyze the competitive forces that determine equilibrium of certificate selling. Our main goal is to find the prices at which CAs obtain their maximum profit, i.e., when they reach the game equilibrium. Recall that these certificates differ in the QoS so that $\forall i, j; i \neq j, t_i \neq t_j$ and $s_i \neq s_j$. We assume that the certification market is covered in full. Users will intend to maximize their utility, i.e.:

$$\theta^* = \arg\max_{\theta} \quad U(P_i, C_i). \tag{7}$$

On the other hand, CAs will intend to minimize their costs. The CA's costs consists of fixed and variable costs. Each time a new CRL is issued, a CA incurs both fixed and variable costs. The fixed cost depends on two factors. The fix component is due to the release of a new CRL, and does not depend on the number or certificate type. The variable factor depends on the number of certificates contained in the CRL (i.e. depends on the size of the CRL) and on the type of certificate (i.e. certificate with higher security level induce higher costs). Note that in this variable cost it is included the cost of processing each certificate revocation request. We define the service quality cost of CA_i (i.e. $Q(s_i, t_i)$) as a variable that includes both fixed and variable costs associated to the QoS. The first and second derivative of $Q(s_i, t_i)$ with respect to s_i, t_i are positive. Hence, we can calculate the gain function G_i of any CA_i:

$$G_i = \theta^* P_i - Q(s_i, t_i), \tag{8}$$

where the gain function captures the overall profits of CA_i for a given certificate product characterized by (P_i, C_i).

We assume that the game between the two CAs is static with incomplete information, they choose the respective certificate price at the same time to maximize their profits. Now we differentiate (8) with respect to P_i and C_i. In order to obtain the certificate price and the coverage in the equilibrium, let each derivative formula equal to zero. Solving the resulting linear system, we will obtain the price of each CA P_i^* and the corresponding coverage C_i^*.

$$P_i^* : \frac{\partial G_i}{\partial P_i} = 0, \quad C_i^* : \frac{\partial G_i}{\partial C_i} = 0. \tag{9}$$

4.1 Duopoly of CAs

To better illustrate the results obtained in the previous section, we particularize the case of the oligopoly to a duopoly where only two CAs are offering

certificates. This simplification, we allows us to draw some conclusion that can be easily extrapolated to the real scenario where there are more than a dozen CAs. To show that the level of service quality depends on the CA, we assume that the CA indexed by $i = 1$ offers better quality than the second CA in both QoS parameters, i.e., $t_1 < t_2$ and $s_1 > s_2$.

Following the methodology aforementioned, we have to find the prices in the equilibrium. In this situation, first we find the value of θ^* at which a user has no obvious trend between the certificates offered by different CAs.

$$\alpha_1[w - P_1 - \pi C_1 - R\pi(1-\pi)C_1^2] + \alpha_2\left[\pi\theta\left(\beta_1 s_1 + \beta_2\frac{1}{t_1}\right)\right] = \alpha_1[w - P_2 - \pi C_2 - R\pi(1-\pi)C_2^2] + \alpha_2\left[\pi\theta\left(\beta_1 s_2 + \beta_2\frac{1}{t_2}\right)\right], \tag{10}$$

which results in:

$$\theta^* = \frac{\alpha_1\left(P_1 - P_2 + \pi C_1(1 + RC_1 - R\pi C_1) - \pi C_2(1 - RC_2 + R\pi C_2)\right)}{\pi\alpha_2 K} \tag{11}$$

where $K = \beta_1(s_1 - s_2) + \beta_2\left(\frac{1}{t_1} - \frac{1}{t_2}\right)$. So the market demand of CA_2 is θ^*, and the demand of CA_1 is $1 - \theta^*$.

Using (8) we calculate the gain function G_i of CA_1 and CA_2 :

$$G_1 = (1 - \theta^* P_1) - Q(s_1, t_1), \tag{12}$$

$$G_2 = \theta^* P_2 - Q(s_2, t_2). \tag{13}$$

We obtain the certificate price and the coverage in the equilibrium :

$$P_1^* = \frac{2\pi\,\alpha_2 K}{3\alpha_1} \qquad P_2^* = \frac{\pi\,\alpha_2 K}{3\alpha_1}, \qquad C_1^* = C_2^* = \frac{1}{2R\,(-1+\pi)}. \tag{14}$$

From these results we can conclude that:

- In the equilibrium, when both CAs achieve their maximum gain, CA_1 obtains a higher price than CA_2. This is mainly due to the fact that when both CAs have associated the same probability of an attack, as the QoS of the first CA is better so that CA_1 can set a higher price per certificate.
- In the equilibrium, the coverage that each CA should establish is the same and is inversely proportional to the risk-aversion and the probability of operating with a revoked certificate.

5 Analysis and Results

5.1 Impact of the preference ratio $\frac{\alpha_2}{\alpha_1}$

As the ratio between the preference of QoS utility and wealth utility of the user increases (i.e., users are more interested in a high security service and

a good revocation mechanism) the prices of both CAs in the equilibrium also increase. This effect is reasonable, as the improvement of the revocation mechanism gives a higher security level which also increases the costs. This cost increment is compensated with a higher price in the equilibrium. Analyzing two CAs operating in the oligopoly such that $t_i < t_j$ and $s_i > s_j$, it is worth noting that the increment speed of CA_i's QoS is faster than that of CA_j, so the increment speed of its certificate price is also faster than the other CA.

5.2 Impact of the security level difference

When the level of security that a CA offers is much higher than in the others, the certificate value is also much higher. Thus, CAs that offer certificates with higher level of encryption and larger keys are able to make their certification product differentiable. For instance, SSL security levels vary depending upon the way on SSL certificate is installed on a server and the configuration used. SSL is simple to use but its security can be compromised if basic installation an configurations are not completed to a competent level, hackers are then able to decrypt the security on a badly installed SSL certificate. Once the certificates of a CA are differentiable from the other CAs, CAs do not have to use malignant prices anymore to compete. As the difference of this QoS between CAs becomes bigger, the prices that they can charge also increase. Note that if the preference extent which the user shows to the security level (i.e. β_1) increases, the differences in the certificates as products will be more apparent, thus the increase in the CA's certificate prices will also increase. The same results are expected with the increment of the interest of the users to a better service from the CAs (α_2), that is, not higher security but also a more efficient revocation mechanism.

5.3 Impact of the QoS of the revocation mechanism

CAs that are able to offer revocation mechanisms with fresher information and high availability are able to make their certification product differentiable. Recall that this QoS increase of the revocation mechanism induces higher costs, as revocation information has to be issued more frequently. These costs are compensated with an increase of the price that CAs can charge for the certificates in the equilibrium. The reasons are the same that in the previous case, but now users pay more attention to the revocation mechanism rather than to the level of security. Analytically, that means that β_2 increases, so that the user is more interested in the efficiency of the revocation mechanism. This increase induces a proportional increase in the equilibrium prices of the CAs. Note that in this case, the increase of CA_i

which has higher QoS of the revocation mechanism is faster than that of CA_j. Again, the CA that has better service (no matter if it is higher security level or a more efficient revocation mechanism) has the advantage in competition.

5.4 Impact of the revocation probability

Logically, with an increase of the probability of operating with a revoked certificate, CAs charge more for their certificates. The reason is obvious as the CAs set they price mainly based on a forecast of this probability. An increase of π will induce an increase of the compensation expenses that a CA will have to pay to any victim of an attack due to the misuse of a revoked certificate. Consequently, this increase will lead to a proportional increase of compensation cost and service cost so that the CAs have to increase their prices to compensate the cost increases. Note that this increase is twice faster in the case of the CA_i.

6 Case Study: SSL Providers

Finally, to corroborate the benefits of the presented model, we analyze the case of current SSL providers that issue digital certificates. An SSL certificate can be obtained from amounts as low as $43 to as high as $3000 per year. Whilst the type of encryption can be the same, the cost is determined by the rigour of the certification process as well as the assurance and warranty that the vendor can provide. Table 1 shows the prices and QoS that the leading CAs operating in the SSL Certificate market are offering. The SSL Certificate market was traditionally dominated by a small number of players, namely VeriSign and Thawte. Whilst in a monopolistic position they had the capability of charging inflated prices for a commodity product. However new providers with no necessity to hold prices high were able to offer SSL certificates at far more reasonable prices.

The SSL certificate vendors provide insurance against the misuse of certificates and this differs from one vendor to another. Verisign provides warranties of up to $250,000 while Entrust and GoDaddy offer a $10,000 warranty. The higher the insurance, the more inscription/authentication is provided by the SSL vendors. Analyzing Table 1, it is worth noting that not always a lower price means lower quality. Therefore, it is evident that current CAs operating in this market are competing both in price and quality of service.

To test whether these factors are determinant factors for the certficate prices, we perform a multivariate regression analysis explaining the yearly price of SSL certificates. General regression investigates and models the relationship between a response (Certificate price) and predictors (Warranty, issuing interval and CRL lifetime). Note that the response of this model is

SSL Provider	Product Name	Price/Year($)	Warranty($)	Assurance	Mean Issuing time	Mean CRL lifetime
COMODO	EnterpriseSSL Platinum	311.80	1,000,000	High	Under 1 hour	4 days
COMODO	InstantSSL Pro	169.80	100,000	High	Under 1 hour	4 days
Verisign	Secure Site Pro Cert	826.67	2,500,000	High	2-3 days	15 days
Verisign	Managed PKI for SSL Std	234.00	100,000	High	2-3 days	15 days
GeoTrust	QuickSSL Premium	118.00	100,000	Low	Immediate	10 days
GeoTrust	True BusinessID	159.20	100,000	High	2 days	10 days
Go Daddy	Standard SSL	42.99	10,000	Low	Immediate	1 day
Go Daddy	Standard Wildcard	179.99	10,000	Low	Immediate	1 day
Entrust	Advantage SSL Certificates	167.00	10,000	High	2 days	1 week
Entrust	Standard SSL Certificates	132.00	10,000	High	2 days	1 week
Thawte	SSL 123	129.80	-	Low	Immediate	1 month
Thawte	SGC Super cert	599.80	-	High	2 days	1 month

Table 1. SSL Certificate Types and Services offered by main CAs [9].

continuous, but you we have both continuous and categorical predictors. You can model both linear and polynomial relationships using general regression. With this model we determine how the certficate price changes as a particular predictor variable changes. We use data from a survey of CAs performed in 2010 [9]. The obtained regression model is expressed in the following equations for high and low assurance certificates, respectively:

$$Price/Year(\$) = 98,4353 + 0,000220857\, W - 0,549141\, \overline{I_{time}} + 8,6116\, \frac{1}{\overline{CRL_{Lf}}},$$

$$Price/Year(\$) = 20,0405 + 0,000220857\, W - 0,5491411\, \overline{I_{time}} + 8,6116\, \frac{1}{\overline{CRL_{Lf}}},$$

where W denotes the warranty, $\overline{I_{time}}$ is the mean issuing time, and $\overline{CRL_{Lf}}$ is the mean lifetime of the CRLs issued by the CA.

Note that both regression equations show that the coefficient of the predictor associated to the CRL's mean lifetime is significant. In fact, the p-value associated to this predictor is $0,008$ which indicates that is statistically significantly. Overall, the variables within the model are explaining a large portion of the variation in the certificate price. With a coefficient of determination R^2 above the 81%, we are capturing important drivers of certificate prices. The residuals from the analysis are normally distributed, i.e., no evidence of nonnormality, skewness, or unidentified variables exists.

Using the proposed model, we are able to explain these different prices and the corresponding market share and they potential evolution. First we analyze the number of revoked certificates as it will determine the probability of operating with a revoked certificate. Figure 1 shows the evolution of the daily number of revoked certificates per CA. These data were collected from different SSL CRLs that the CAs make public at their repositories. It is worth

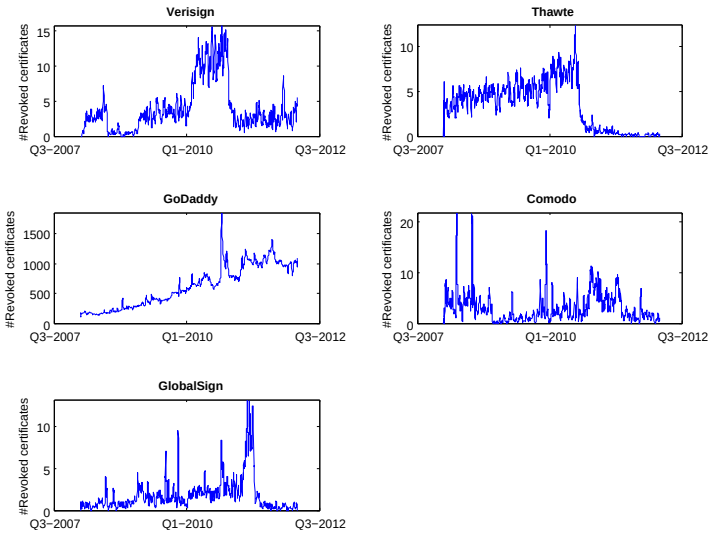


Fig. 1. Evolution of the daily number of revoked certificates per CA.

noting, that the number of revoked certificates highly varies depending on the CA. Thus, GoDaddy revokes more than 500 certificates per day on average while VerSign revokes less than 4 certificates per day on average. Therefore, the probability π of operating with revoked certificates is higher when trusting certificates issued by GoDaddy. As our model shows, using expression (14), the probability π directly affects the price of the certificate. Thus, as GoDaddy has a higher π, we would expect to charge less for its certificates. However, the price is quite similar to its competitors. Thus, GoDaddy is not able to sell as much certificates as the other oligarchs, and its market share is smaller.

Our model would expect GoDaddy to compete not only in prices but also in QoS to gain market share. As our model shows, the reaction of GoDaddy to compete in the oligopoly is to offer better quality of service. From table 1, we can see that GoDaddy is the CA that issues CRLs more often. Using this CRL releasing policy, users increase their utility and, at the same time, the probability of operating with a revoked certificate is also reduced. However, the variable costs increase due to this way of issuing CRLs. Similarly, Comodo intends to gain market share by decreasing the time it takes to issue a certificate and also reducing the CRL lifetime. Note that VeriSign, the leading CA, is the one who is offering the worst QoS, both in terms of CRL lifetime and time to issue a new certificate.

7 Conclusions

The market of certificate providers can be described as an oligopoly where oligarchs compete not only in price but also in quality of service. In this paper we have modeled this oligopoly using a game theoretic approach to find the prices in the equilibrium. We have been able to capture the QoS of the products offered by a CA, by means of the timeliness of the revocation mechanism and the security level. In our model of the certification industry with profit-maximizing CAs and a continuum of individuals we showed that although the undercutting process in certification prices seems similar to the price setting behavior of firms in Bertrand competition there exists a crucial difference depending onf the QoS of the revocation service. The solution of the game for two CAs in the oligopoly that offer certificates with different QoS shows that the revenues of the CA which provides a better revocation mechanism and a higher security level are larger. Therefore, a CA when setting the prices of its certificate and the compensation expenses, it has to take into account not only the probability of operating with a revoked certificate, but also the quality of the revocation mechanism and the security level. Thus, any CA should comprehensively consider the difference in quality of its services compared with other CAs.

References

1. C. Adams and S. Farrell. Internet X.509 Public Key Infrastructure Certificate Management Protocols. RFC 2510, Internet Engineering Task Force, March 1999.
2. R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, Internet Engineering Task Force, April 2002.
3. T. Perlines Hormann, K. Wrona, and S. Holtmanns. Evaluation of certificate validation mechanisms. *Comput. Commun.*, 29:291–305, February 2006.
4. A. Arnes. Public key certificate revocation schemes. 2000. Queen's University. Ontario, Canada. Master Thesis.
5. D.A. Cooper. A more efficient use of Delta-CRLs. In *2000 IEEE Symposium on Security and Privacy. Computer Security Division of NIST*, pages 190–202, 2000.
6. Mona H. Ofigsbø, Stig Frode Mjølsnes, Poul Heegaard, and Leif Nilsen. Reducing the cost of certificate revocation: a case study. In *Proceedings of the 6th European conference on Public key infrastructures, services and applications*, EuroPKI'09, pages 51–66, Berlin, Heidelberg, 2010. Springer-Verlag.
7. Chengyu Ma, Nan Hu, and Yingjiu Li. On the release of CRLs in public key infrastructure. In *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*, Berkeley, CA, USA, 2006.
8. Nan Hu, Giri K. Tayi, Chengyu Ma, and Yingjiu Li. Certificate revocation release policies. *J. Comput. Secur.*, 17:127–157, April 2009.
9. WhichSSL. SSL Market Share, 2010. [Online] http://www.whichssl.com/ssl-market-share.html.

On the self-similarity nature of the revocation data

Carlos Gañán, Jorge Mata-Díaz, Jose L. Muñoz
Oscar Esparza and Juanjo Alins

Universitat Politècnica de Catalunya, Telematics Department, Barcelona (Spain)
{carlos.ganan,jmata,jose.munoz,oscar.esparza,juanjo}@entel.upc.edu

Abstract. One of the hardest tasks of a Public Key Infrastructure (PKI) is to manage revocation. Different revocation mechanisms have been proposed to invalidate the credentials of compromised or misbehaving users. All these mechanisms aim to optimize the transmission of revocation data to avoid unnecessary network overhead. To that end, they establish release policies based on the assumption that the revocation data follows uniform or Poisson distribution. Temporal distribution of the revocation data has a significant influence on the performance and scalability of the revocation service. In this paper, we demonstrate that the temporal distribution of the daily number of revoked certificates is statistically self-similar, and that the currently assumed Poisson distribution does not capture the statistical properties of the distribution. None of the commonly used revocation models takes into account this fractal behavior, though such behavior has serious implications for the design, control, and analysis of revocation protocols such as CRL or delta-CRL.

Keywords: Self-similarity, Certification, Public Key Infrastructure, Revocation.

1 Introduction

Today we are in the midst of an electronic business revolution. It is of utmost importance that mechanisms are set up to ensure information and data security. Organizations have recognized the need to balance the concern for protecting information and data with the desire to leverage the electronic medium. Public Key Infrastructure (PKI) is a step toward providing a secure environment by using a system of digital certificates and certificate authorities (CAs). However, one of the most important aspects in the design of a PKI is certificate revocation.

Certificate revocation is the process of removing the validity of a certificate prematurely. There could be multiple reasons for revoking a certificate; such as the certificate holder leaves the organization or there is

a suspicion of private key compromise. When a certificate is revoked, the information about the revoked certificate needs to be published. Some of the methods that a CA can use to revoke certificates are:

- Periodic Publication Mechanisms: Information about revoked certificates can be posted on a certificate server so that the users are warned from using those certificates. This mechanism includes the use of Certificate Revocation Lists (CRL) and Certificate Revocation Trees (CRT). A CRL is a signed list of certificates that have been revoked or suspended. CRT is a revocation technology, which is based on Merkle hash trees, where the tree represents all known certificate revocation information relevant to some known set of PKI communities.
- Online Query Mechanisms: Online Query Mechanisms comprise Online Certificate Status Protocol (OCSP) and Online Transaction Validation Protocols. OCSP is used to obtain online revocation information about certificates, and Online Transaction Validation Protocols are used for online validation, such as business transactions through credit cards.

A revocation method is selected by an organization based on the cost, infrastructure, and volumes of transactions that are expected. To gauge these costs, different revocation mechanisms are tested under the assumption that the revocation events follow a specific probability distribution. Most theoretical frameworks and simulation studies for performance evaluation assume that the temporal distribution of queries follows a Poisson distribution. Thus, organizations estimate the infrastructure needed to deploy the PKI and the associated costs. However, in this article, we demonstrate that revocation data is statistically *self-similar*, that none of the commonly used revocation models is able to capture this fractal behavior, and that such behavior has serious implications for the design, control, and analysis of revocation protocols such as CRLs.

We start by analyzing the validity of Poisson-like process assumption. We use publicly available CRLs from different certification authorities (containing more than 300,000 revoked certificates over a period of three years). Our analysis demonstrates that the Poisson distribution fails to capture the statistical properties of the actual revocation process. We also see that the Poisson distribution grossly under-estimates the bandwidth utilization of the revocation mechanism. At first glance, this might look like an obvious result, since after all as a memoryless process, Poisson distribution cannot be expected to model periodic trends like daily, weekly and monthly cycles in revocation rates. We show however that the modeling inability transcends simple cycles. In particular, we will show that

self-similarity has a severe detrimental impact on the revocation service performance.

Results of our analysis, including burstiness at all scales, strongly suggest self-similar nature of revocation events. We confirm this by estimating the Hurst parameter for the observed distribution and showing that the estimates validate self-similar nature of the revocation lists. Beyond invalidating Poisson-like distributions, this proof of self-similarity has the important implications on CA utilization, throughput, and certificate stratus checking time. Intuitively, as the revocation process is bursty (non-uniformly distributed) the CA will be partially idle during low burst periods and vice versa. Thus, the revocation lists will grow non-uniformly, and current updating policies will result bandwidth inefficient.

The rest of this article is organized as follows. Section 2 gives the necessary statistical background required to understand self-similar processes and long range dependency. In Section 3, we discuss the methodology we used to collect and analyze real-world revocation data. We demonstrate self-similar nature of the revocation data, followed by a Hurst parameter estimation. In Section 4 we discuss how the observed self-similarity has crucial implications on performance of the revocation service. Next section discusses the related work in the area. Finally, we conclude in Section 6.

2 Background

2.1 Self-Similar Processes

A phenomenon which is self-similar looks the same or behaves the same when viewed at different degree of magnification. Self-similarity [1] is the property of a series of data points to retain a pattern or appearance regardless of the level of granularity used and can be the result of long-range dependence (LRD) in the data series. One of the main properties of the self-similar data is burstiness [1]. Bursty data do not possess a stable mean value. Significant differences in the mean value are one of the reasons why bursty data are more difficult to control than shaped one. If a self-similar process is bursty at a wide range of timescales, it may often exhibit long-range dependence. Long-range-dependence means that all the values at any time are correlated in a positive and non-negligible way with values at all future instants.

A stochastic process $Y(t)$ is *self-similar* with Hurst parameter H if for any positive stretching factor d, the distribution of the rescaled and

reindexed process $d^{-H}Y(dt)$ is equivalent to that of the original process $Y(t)$. This means for any sequence of time points $t_1, \ldots, t_n$ and any positive constant d, the collections $\{d^{-H}Y(dt_1), \ldots, d^{-H}Y(dt_n)\}$ and $\{Y(t_1), \ldots, Y(t_n)\}$ are governed by the same probability law. When the values of H are in the interval $(0.5, 1)$, the process presents LRD. A value of H equal to 0.5 indicates the absence of LRD. This means that the smoothing with aggregation is much slower for self-similar processes, the greater the degree of self-similarity, the slower will be smoothing with aggregation.

Three implications of self-similarity are:

- No natural length of bursts.
- Presence of bursts in all time scales.
- Process does not smooth out on aggregation.

2.2 Statistical Tests For Self-Similarity

The practical way to estimate degree of self-similarity is to measure the values of Hurst exponent. In this paper we use five methods to test for self-similarity (details about these methods are described in [2, 3]).

The first method, the variance-time plot, relies on the slowly decaying variance of a self-similar series. The variance of $Y^{(m)}$ is plotted against m on a log-log plot; a straight line with slope (β) greater than -1 is indicative of self-similarity, and the parameter H is given by $H = 1 - \beta/2$. The second method, the R/S plot, uses the fact that for a self-similar dataset, the rescaled range or R/S statistic grows according to a power law with exponent H as a function of the number of points included (n). Thus the plot of R/S against n on a log-log plot has slope which is an estimate of H. The third approach, the periodogram method, uses the slope of the power spectrum of the series as frequency approaches zero. On a log-log plot, the periodogram slope is a straight line with slope close to the origin.

While the preceding three graphical methods are useful for exposing faulty assumptions (such as non-stationarity in the dataset) they do not provide confidence intervals. The fourth method, called the Whittle estimator does provide a confidence interval, but has the drawback that the form of the underlying stochastic process must be supplied. The two forms that are most commonly used are fractional Gaussian noise (FGN) with parameter $1/2 < H < 1$, and Fractional ARIMA(p,d,q) with $0 < d < 1/2$ (for details see [2])). These two models differ in their assumptions about the short-range dependences in the datasets; FGN assumes no short-range

dependence while Fractional ARIMA can assume a fixed degree of short-range dependence. There are several other methods in frequency and time domain to measure the Hurst parameter.

Finally, we use the Detrended Fluctuation Analysis (DFA) [4], which aims to highlight the long-range dependence of a time series with trend. DFA method is a version for time series with trend of the method of aggregated variance used for a long-memory stationary process. It consists in aggregating the process by windows with fixed length, detrending the process from a linear regression in each window, computing the standard deviation of the residual errors (the DFA function) for all data, and finally, estimating the coefficient of the power law from a log-log regression of the DFA function on the length of the chosen window.

3 Examining the self-similarity of the revocation process

3.1 Data Collection

In order to capture the temporal correlation of the revocation process, first we have to gather a large sample of revocation data. The approach we follow consists in collecting revocation data from different certification authorities using their available CRLs. In particular, we built some scripts to download and preprocess the CRLs from the following CAs[1]: VeriSign, GoDaddy, Thawte, and Comodo.

Issuer Name	Number of Revoked Certificates	Last Update	Next Update
GoDaddy	932,900	2012/02/01	2012/02/03
VeriSign	5,346	2012/02/02	2012/02/16
Comodo	2,727	2012/02/03	2012/02/06
GlobalSign	7,591	2012/02/02	2012/03/03
Thawte	8,061	2012/02/01	2012/02/16

Table 1: Description of the collected CRLs.

Though we concentrate our analysis on CRL because it is the most common and simplest method for certificate revocation [6], we expect the

[1] According to NetCraft's survey [5], using these CAs we cover most of the world market for SSL.

captured pattern to be extensible to any other revocation mechanism (e.g. OCSP).

Once downloaded the revocation data, we preprocess these data to remove duplicated information (e.g. certificates that are revoked due to several reasons). Note that when a revoked certificate expires, it typically remains in the CRLs for one additional publication interval, so we preprocess the CRLs to remove expired certificates too. In this sense, Thawte's and GlobalSign's CRLs may contain duplicate entries for the same certificate because of their policy statements. These policy statements impose that a certificate that is revoked by several reasons must be included in the CRL as many times as the number of revocation reasons. Thus, we remove any duplicate entry from the composite dataset, and tally the number of revocations per day. Finally, we build a dataset that covers non-expired revoked certificates from 2008 to 2012 (see Figure 1).

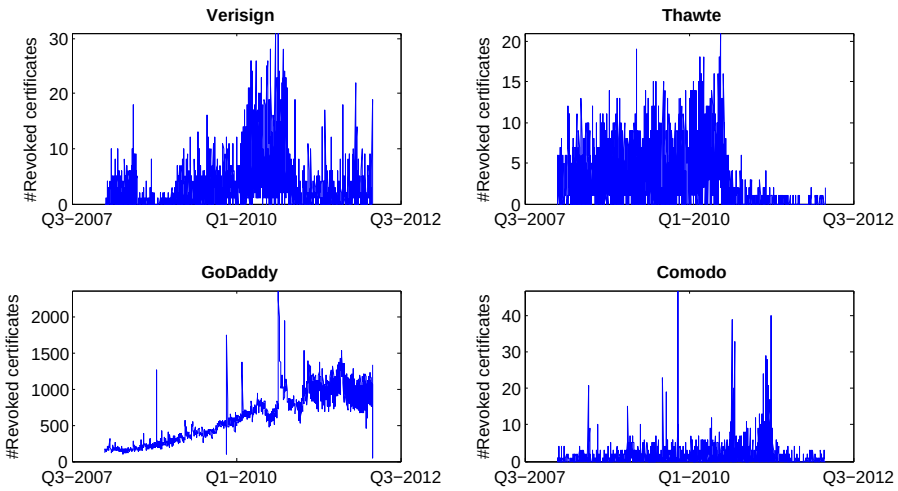


Fig. 1: Number of daily revoked certificates evolution for each CA.

3.2 Evidence of Burstiness

Before providing formal estimation of self-similarity, we provide a graphical evidence of bursty nature of the revocation data at different time scales. We also show that this observed burstiness is not accounted by the Poisson distribution. In Figure 2, we show the revocation logs in four different time scales-ranging from 1 hour to 1 day. Each plot is obtained by

changing the time resolution. In Figure 2 we can observe different evident trends; (i) Burstiness in all time scales: the burstiness of the revocation process does not disappear when changing the time scales. (ii) Lack of natural length of bursts: The figure shows burstiness ranging from days to months. Note that the full duration of the figure with the largest time slot is 1,000 days, and some of the bursts have many hours of duration.

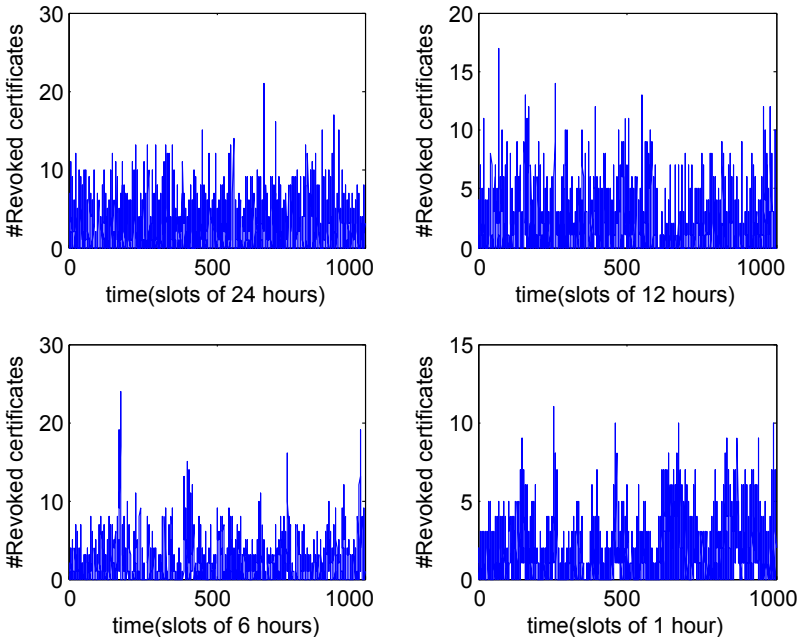


Fig. 2: Revocation Bursts over Four Orders of Magnitude.

In addition, it is worth noting the difference between this bursty pattern and a Poisson process. A Poisson process smooths out with large time scales and resembles a uniformly distributed white noise at higher time scales. In contrast to the revocation process, in a Poisson process the burstiness vanishes in coarse time scales, longer length bursts are absent, and bursts smooths out much faster. Thus, the trends of self-similarity present in the revocation data discussed above are totally absent for Poisson processes.

Therefore, modeling the revocation process as Poisson is clearly inadequate, and is thus likely to give unrealistic results. We will elaborate this

analysis in the next section, and discuss the consequences of self-similarity in the following sections

3.3 Statistical Analysis of Self-Similarity

In this section, we use five different methods to estimate the Hurst parameter to demonstrate the long range dependency of the revocation events formally. Since there are different manifestations of self-similarity, different methods in time and frequency domains are used in practice for the estimation (see Sec. 2.2). Note that when using these estimators with real-life revocation data containing noise, cycles and trends, they might estimate different values of the Hurst parameter. For that reason, we use multiple methods, report the correlation coefficients and confidence intervals by different methods, and visually inspect the data for trends and cycles. The chances of estimates agreeing on real data is small [7], but if most of the estimates are above 0.5 the LRD is likely to exist.

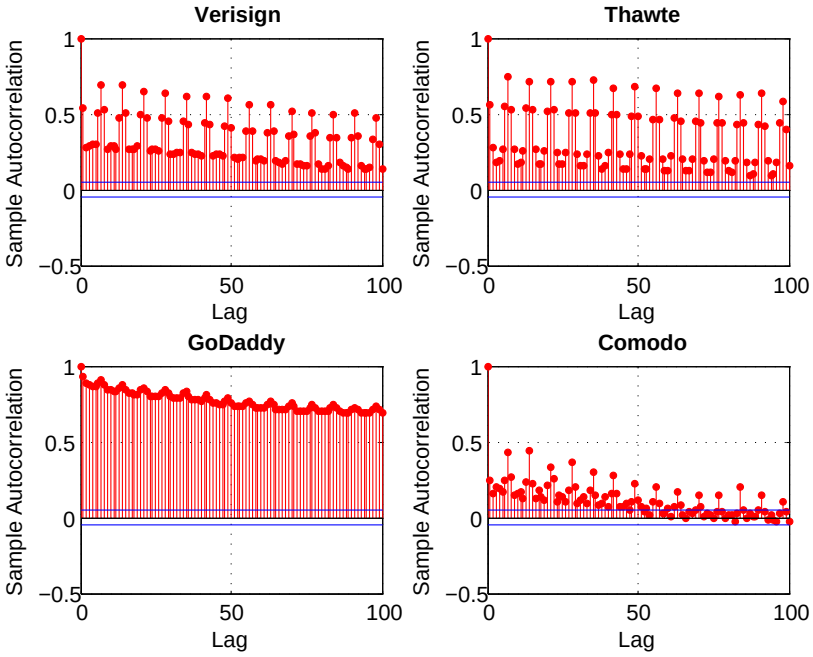


Fig. 3: Autocorrelation function of the revocation process per CA.

First of all, we start analyzing the autocorrelation of the revocation data. Recall that in a self-similar process autocorrelations decay hyperbolically rather than exponentially fast, implying a nonsummable autocorrelation function $\sum_k r(k) = \infty$ (long-range dependence). For the frame data, the empirical autocorrelation functions $r(k)$ are shown in Fig. 3, with lag k ranging from 0 to 100. Notice that $r(k)$ decreases slower than exponentially no matter the CA. The curve does decay toward zero, but it does so extremely slowly. The very slowly decaying autocorrelations are indicative of LRD.

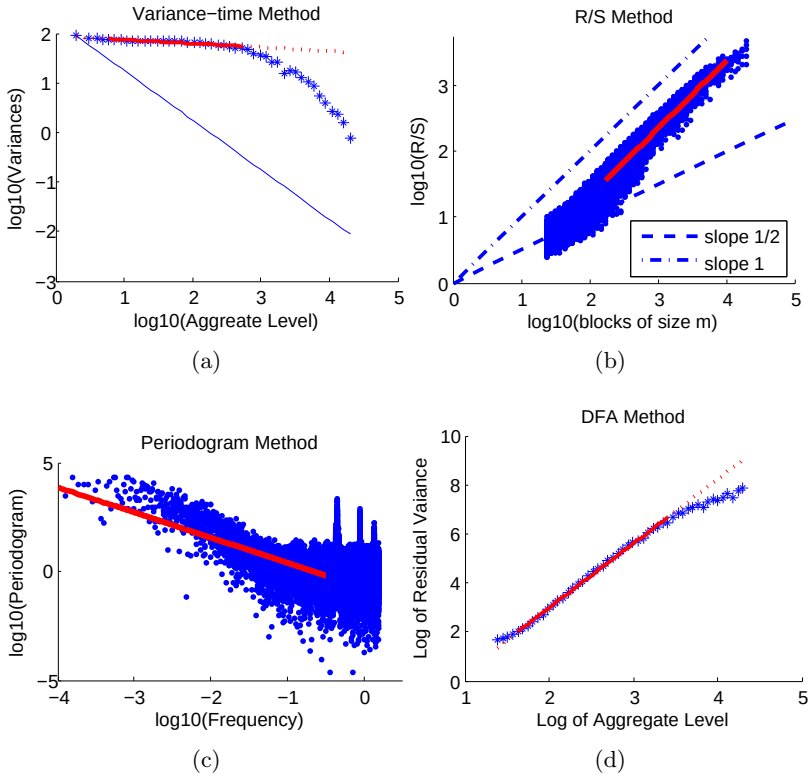


Fig. 4: Graphical methods for checking for self-similarity of the revocation process from GoDaddy (a) variance-time plot, (b) pox plot of R/S, (c) periodogram plot, and (d) DFA plot.

In the following, we use five different methods for assessing self-similarity described in Section 2.2: the variance-time plot, the rescaled range (or

R/S) plot, the periodogram plot, the DFA plot and the Whittle estimator. We concentrated on individual months from our revocation time series, so as to provide as nearly a stationary dataset as possible. To provide an example of these approaches, analysis of a single month from GoDaddy revocation data is shown in Figure 4. The figure shows plots for the four graphical methods: variance-time (upper left), rescaled range (upper right), periodogram (lower left) and DFA (lower right). The variance-time plot is linear and shows a slope that is distinctly different from -1 (which is shown for comparison); the slope is estimated using regression as -0.077, yielding an estimate for H of 0.96. The R/S plot shows an asymptotic slope that is different from 0.5 and from 1.0 (shown for comparison); it is estimated using regression as 0.95, which is also the corresponding estimate of H. The periodogram plot shows a slope of -0.14 (the regression line is shown), yielding an estimate of H as 0.83. Finally, the Whittle estimator for this revocation data (not a graphical method) yields an estimated Hurst value of 0,923 with a 95% confidence interval of $(0.87, 0.95)$.

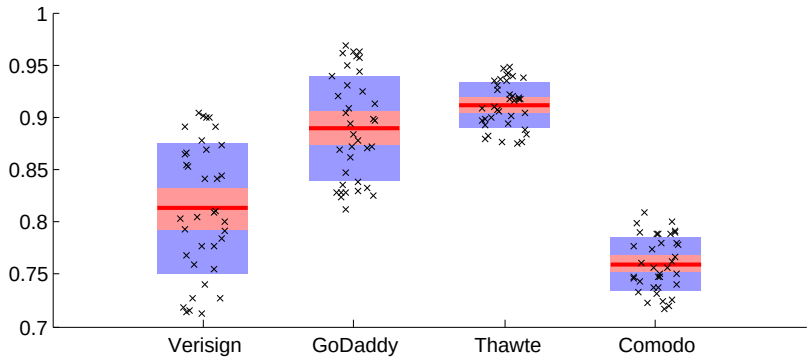


Fig. 5: Summary plot of estimates of the Hurst parameter H for all the CAs.

Once we have seen that GoDaddy presents a significant self-similar pattern, we analyze the rest of the CAs. To that end, we use the whittle estimator to obtain the Hurst value per CA and month. We chose this estimator because it gives more refined measurement than other estimation techniques and it provides confidence levels for the Hurst parameter [8]. Note that we are not interested in estimating the exact value of the Hurst parameter but to prove the existence of self-similarity in the revocation

data. Figure 5 shows the H parameter of each CA and the 95% confidence interval. It is worth noting that depending on the month there are some CAs whose H parameter varies significantly. However, no matter neither the CA nor the month, the Hurst value is always above 0.7. This means that the revocation process of any CA presents LRD.

4 Significance of self-similarity for revocation data management

Our collected data from real CAs show dramatically different statistical properties than those assumed by the stochastic models currently considered in the literature. Almost all these models are characterized by an exponentially decaying autocorrelation function. As a result, they give rise to a Hurst parameter estimate of $\hat{H} = .50$, producing variance-time curves, R/S plots, and frequency domain behavior strongly disagreeing with the self-similar behavior of actual revocation (see Section 3.3). In this section, we emphasize direct implications of the self-similar nature of the revocation data in the performance of the revocation service.

4.1 Impact on the revocation mechanism

As we mentioned before, traditional mechanisms made assumptions about the revocation process to obtain efficient revocation data issuing policies. However, these assumptions neglect the self-similar nature of the revocation data. This has a direct impact due to the "burstiness" of the data and affects the congestion management of the CA/repositories.

To give an idea of the impact of self-similarity, we analyze the work of Cooper in [9] and in [10]. In these works, Cooper analyzed the best way to issue CRLs, segmented CRLs and delta-CRLs in order to decrease the request peak bandwidth. The author assumed that an average of 1,000 certificates are revoked each day and that the CRLs have a fixed validity time. By doing these assumptions, the self-similar behavior of the revocation process is neglected and the results need to be adapted to the reality.

Using the traditional approach, CRLs are published periodically. Under this assumption, CAs expect that consecutive CRLs should have similar size. However, this assumption is proven completely wrong when bursts are present. Thus, consecutive CRLs can differ significantly in the number of revoked certificates they include, and, consequently in their size. Using

the data collected from Verisign[2], we studied how the size of the CRLs varies when CRLs are issued daily. As in [10], we estimate that the size of a CRL is 51 bytes plus 9 bytes for each certificate included on the CRL. If an average of r certificates are revoked each day, certificates are valid for L_c days, and a certificate, at the time of revocation, has an average of $\frac{L_c}{2}$ days until it expires, then the average size of a CRL will be [10]:

$$Size_{CRL} = 51 + 4.5 \cdot r \cdot L_c.$$

We assume that certificates have a lifetime of 365 days [11], therefore we can calculate the daily size of the Verisign CRL for 5 randomly chosen months. We execute the trial several times and check that the same depency is obtained. Figure 6 shows the results in a box-plot. Note that the CRL size has a mean size of around 150 KBytes, but it highly varies due to the revocation bursts. For instance, during March 2008, there were four CRLs that exceed the 300 KBytes. These variations are highly inefficient in terms of bandwidth, as during some days the required bandwidth double the bandwidth needed in previous days. Although this has not become a bottleneck in wired networks, novel scenarios (e.g. Vehicular Networks) cannot afford these variations.

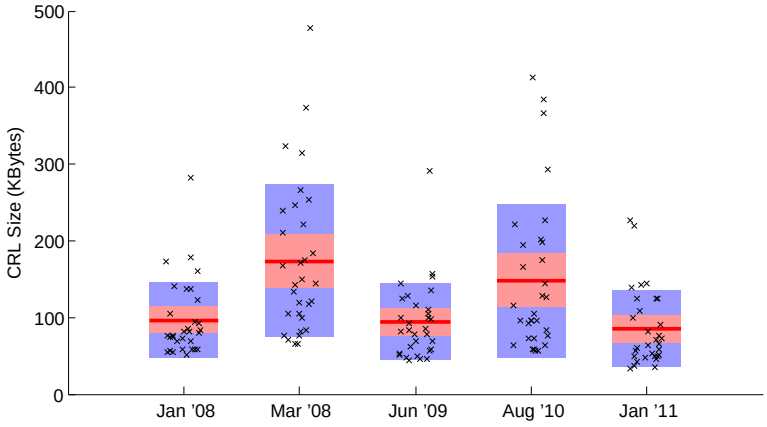


Fig. 6: Estimated daily size of Verisign's CRL.

[2] Note that we use the data from VeriSign to provide a case study of variance in size of the CRLs. The same variance pattern applies to the other CAs, though it is not shown in this article.

However, the self-similarity not only affects traditional CRL issuance, but also its variants that aim to be bandwidth efficient such as delta-CRL. From [10], the bandwidth for a delta-CRL system can be computed as:

$$B = \frac{Nve^{-vt}((51 + 4.5rL_c)e^{-(w+\frac{l}{O}-l)v} + (51 + 9rw))}{(O-1)1 - e^{vl/O} + 1}, \quad (1)$$

where N is the number of valid certificates, v is the validation rate, l is the amount of time that a delta-CRL is valid, L_c is the certificate lifetime, r is the number of certificates revoked per day, w is the window size of the delta-CRL and O is the number of delta-CRLs that are valid at any given time.

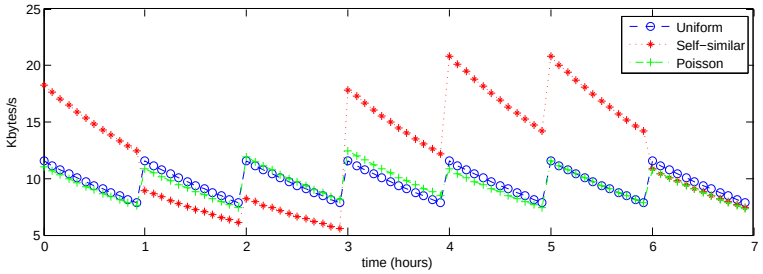


Fig. 7: Delta-CRL BW consumption.

Using the bandwidth as a comparison metric, we can evaluate the impact of the self-similarity. Figure 7 shows the bandwidth necessary to download the revocation data using a sliding window delta-CRL scheme. We have assumed that there are 300,000 relying parties (N) each validating an average of 10 certificates per day (v); delta-CRLs are issued once an hour, are valid for 4 hours (O), and have a window size of 9 hours (w). We have also assumed that an average of 10 certificates are revoked each day (r) and that certificates are valid for 365 days (L_c). Note that depending on the distribution of the revocation process, the required bandwidth presents significant variations. We change the number of certificates revoked per day (r) according to three different distributions (i.e. uniform, Poisson and self-similar) and evaluate the required bandwidth of a delta-CRL system using Eq. (1). Uniform and Poisson distributions present a similar behavior. On the opposite, a self-similar process makes the delta-CRL's size to vary. Thus, the optimal window to issue delta-CRLs should be calculated taking into account the bursty

pattern of the self-similar process. If this pattern is neglected, the peak bandwidth will vary with each delta-CRL issuance making the revocation service bandwidth-inefficient. When with a Poisson or uniform process the maximum peak bandwidth is of ∼12Kb/s, a burst of revocation events causes that some delta-CRL issuance require more than ∼20Kb/s. Therefore, ignoring the self-similar pattern of the revocation process leads to inaccurate network planning.

CRL releasing strategies might be optimized considering the effect of self-similarity. Periodic updates might create bottlenecks at the repositories when all users request new information at the same time. On the other hand, online checking mechanisms such as OCSP, could be computationally overloaded during bursty periods. Such mechanisms that base their efficiency on using pre-signed responses have not been conceived to work under bursty patterns. Therefore, further analysis should be conducted to establish pre-signing policies under bursty revocation periods.

5 Related Work

Most of previous studies fail to capture the characteristics of real-world revocation data; instead, they focus on theoretical aspects of certificate revocation including the model of revocation [9], the revocation cause [12], and the cost of issuing revocation information [13]. Thus, these theoretical models are not able to capture the actual pattern of the revocation data. Most recently, the statistical properties of real revocation data have been studied [14–16]. Nevertheless, the bursty pattern of the revocation process is neglected.

Regarding the traditional way of issuing CRLs, X.509 [17] defines one method to release CRLs. This method involves each CA periodically issuing CRLs. Using this method, the number of revoked certificates contained in each CRL varies significantly. Thus, each CRL has a different size, and the issuance of the CRLs results bandwidth inefficient. Authors in [15] already acknowledged the inefficiencies of the traditional method, and proposed releasing CRLs based on a set of economic costs. However, they assumed a Poisson process when characterizing the number of new certificate revocations, i.e., they neglected the burst pattern. Thus, the resulting CRL releasing policies could be improved by taking into account the self-similarity of the revocation process. Similarly, authors in [18] collected empirical data about the reasons and frequency of user terminations that require certificate revocations, and then model the consequences for certificate revocation. They investigate how to reduce the

cost of certificate revocation by reducing the number of revoked certificates and bandwidth consumption in order to achieve better scalability.

In the same manner, authors in [14] carried out a thorough empirical analysis of the revocation data not only taking into account the number of revoked certificates, but also other factors such as geographical regions and revocation causes. They also conclude that their collected CRLs exhibit exponential distribution patterns. Though they acknowledge the existence of revocation bursts, they do not capture this behavior. On the other hand, authors in [16] suggest a functional form for the probability density function of certificate revocation requests. They choose an exponential distribution function because it adequately approximates the data they collected from a single CA. Based on this assumption, they provide an economic model based on which a CA can choose what they state to be the optimal CRL release interval. However, they do not take into account the self-similar behavior of the revocation data.

6 Conclusions

Current simulation studies for performance evaluation and revocation data release strategies most commonly assume that the temporal distribution of revocation events follows a Poisson distribution. In this paper, we questioned the assumption of Poisson distribution. Our analysis of the revocation data contained in different CRLs provides significant evidence that the real revocation events follow a self-similar distribution. In particular, our analysis showed burstiness at all time-scales, confirming scale-invariance of distribution. We also estimated and showed that Hurst parameter for the daily number of revoked certificates is above 0.5, proving the self-similarity and Long Range Dependence formally.

We then turned our attention to understanding its consequences on the performance of the revocation services. We showed that traditional revocation mechanisms, such as CRLs or delta-CRLs, do not take into account the bursty pattern of the revocation events when establishing the issuing strategies. These bursts increase the maximum peak bandwidth required to provide the revocation data timely. Thus, self-similarity has a profound effect on the engineering of traditional mechanisms and should be taking into account when designing new revocation protocols.

References

1. Walter Willinger, Vern Paxson, and Murad S. Taqqu. *Self-similarity and heavy tails: structural modeling of network traffic*, pages 27–53. 1998.

2. J. Beran. *Statistics for Long-Memory Processes*. Monographs on Statistics and Applied Probability. Chapman & Hall, 1994.
3. Murad S. Taqqu, Vadim Teverovsky, and Walter Willinger. Estimators for long-range dependence: An empirical study. *Fractals*, 3:785–798, 1995.
4. C K Peng, S Havlin, H E Stanley, and A L Goldberger. Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. *Chaos Woodbury Ny*, 5(1):82–87, 1995.
5. Netcraft. Market share of certification authorities, 2009. https://ssl.netcraft.com/ssl-sample-report/CMatch/certs Accessed on 05/2011.
6. Gaurav Jain. Certificate revocation: A survey. http://csrc.nist.gov/pki/welcome.html Accessed on 05/2011.
7. Thomas Karagiannis, Michalis Faloutsos, and Rudolff H. Riedi. Long-range dependence: now you see it, now you don't. In *in Proc. GLOBECOM '02*, pages 2165–2169, 2002.
8. Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, feb 1994.
9. D.A. Cooper. A model of certificate revocation. In *Fifteenth Annual Computer Security Applications Conference*, pages 256–264, 1999.
10. D.A. Cooper. A more efficient use of Delta-CRLs. In *2000 IEEE Symposium on Security and Privacy. Computer Security Division of NIST*, pages 190–202, 2000.
11. Technological infrastructure for pki and digital certification. *Computer Communications*, 24(14):1460 – 1471, 2001.
12. B. Fox and B. LaMacchia. Certificate Revocation: Mechanics and Meaning. In *International Conference on Financial Cryptography (FC98)*, volume 1465, pages 158–164, February 1998.
13. M. Naor and K. Nissim. Certificate Revocation and Certificate Update. *IEEE Journal on Selected Areas in Communications*, 18(4):561–560, 2000.
14. Daryl Walleck, Yingjiu Li, and Shouhuai Xu. Empirical analysis of certificate revocation lists. In *Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security*, pages 159–174, 2008.
15. Chengyu Ma, Nan Hu, and Yingjiu Li. On the release of CRLs in public key infrastructure. In *Proceedings of the 15th conference on USENIX Security Symposium*, volume 15, pages 17–28, 2006.
16. Nan Hu, Giri K. Tayi, Chengyu Ma, and Yingjiu Li. Certificate revocation release policies. *Journal of Computer Security*, 17:127–157, April 2009.
17. ITU/ISO Recommendation. X.509 Information Technology Open Systems Interconnection - The Directory: Autentication Frameworks, 2000. Technical Corrigendum.
18. Mona Ofigsbø, Stig Mjølsnes, Poul Heegaard, and Leif Nilsen. Reducing the cost of certificate revocation: A case study. In *Public Key Infrastructures, Services and Applications*, volume 6391 of *Lecture Notes in Computer Science*, pages 51–66. 2010.

PKIX Certificate Status in Hybrid MANETs

Jose L. Muñoz, Oscar Esparza, Carlos Gañán, Javier Parra-Arnau

Universitat Politècnica de Catalunya (Departament Enginyeria Telemàtica)**
1-3 Jordi Girona, C3 08034 Barcelona (Spain)
{jose.munoz,oscar.esparza,carlos.ganan,javier.parra}@entel.upc.es

Abstract. Certificate status validation is a hard problem in general but it is particularly complex in Mobile Ad-hoc Networks (MANETs) because we require solutions to manage both the lack of fixed infrastructure inside the MANET and the possible absence of connectivity to trusted authorities when the certification validation has to be performed. In this sense, certificate acquisition is usually assumed as an initialization phase. However, certificate validation is a critical operation since the node needs to check the validity of certificates in real-time, that is, when a particular certificate is going to be used. In such MANET environments, it may happen that the node is placed in a part of the network that is disconnected from the source of status data at the moment the status checking is required. Proposals in the literature suggest the use of caching mechanisms so that the node itself or a neighbour node has some status checking material (typically on-line status responses or lists of revoked certificates). However, to the best of our knowledge the only criterion to evaluate the cached (obsolete) material is the time. In this paper, we analyse how to deploy a certificate status checking PKI service for hybrid MANET and we propose a new criterion based on risk to evaluate cached status data that is much more appropriate and absolute than time because it takes into account the revocation process.

Keywords: Certification, Public Key Infrastructure, Revocation, Hybrid MANET, Risk.

1 Introduction

MANETs (Mobile Ad-hoc Networks) are cooperative networks that allow wireless nodes to establish spontaneous communications. As stated in [1], such networks are envisioned to have dynamic, sometimes rapidly-changing, random, multi-hop topologies which are likely composed of relatively bandwidth constrained wireless links. MANETs may operate in

** This work is funded by the Spanish Ministry of Science and Education under the projects CONSOLIDER-ARES (CSD2007-00004), SECCONET (TSI2005-07293-C02-01), ITACA (TSI2007-65393-C02-02), P2PSEC (TEC2008-06663-C03-01) and, by the Government of Catalonia under grant 2005 SGR 01015 to consolidated research groups.

isolation (stand-alone), or they may have gateways to fixed networks. In this last case, the MANET is called "hybrid". Hybrid MANETs are expected to be deployed as an extension to the traditional infrastructure networks. Also notice that the hybrid behaviour can be temporary due to the situation in which an ad-hoc network may be sometimes stand-alone and sometimes connected to the Internet e.g. a subway network in which a MANET user is connected to the Internet while being at the station and disconnected while traveling. The Hybrid MANET scenario is the one considered in this paper.

On the other hand, trust and security are basic requirements to support business applications in this scenario. The public key scheme is the preferred underlying mechanism to provide security services. In a public key scheme, each participant has two keys: a public key (i.e. known by everybody) and a private key (i.e. secret). The announcement of the public key is performed using a signed document called Public Key Certificate (PKC) or simply "certificate" that binds the participant with her public key. The entity that signs the certificate is called "certificate issuer" or "Certification Authority" (CA). In the literature, there are several ways of managing security and trust in MANETs based on public key cryptography. These approaches basically differ in the degree of decentralization of the mechanisms deployed for issuing, publishing and revoking the certificates (these approaches are reviewed in further detail in the next section).

In decentralized architectures such as [2] and [3] the nodes inside the ad-hoc network participate in the certification process. On the other hand, in the centralized architecture the certification process is fully controlled by an external CA that is a Trusted Third Party (TTP). In this case the CA digitally signs certificates, ensuring that a particular public key belongs to a certain user and the overall certification process is performed according to a standard and publicly available policy. Each scheme has its application scenario: decentralized approaches are suitable for autonomous MANETs or hybrid MANETs that do not require a centralized enforced certification mechanism while the centralized approach is suitable for hybrid MANETs in which inter-operability with currently deployed centralized public key infrastructures (PKIs) is required.

The problem of using a centralized approach is that current PKIs are designed for wired and well-connected networks, so adopting PKIs for hybrid MANETs is not an easy task. Mobile users are expected to move across different networks. When the user is in a network with connection to the PKI, she can use all the PKI services such as get a certificate,

launch a status query, etc. However, users may be disconnected from the PKI when they require a real-time PKI service. In this sense, the certificate status checking is a critical service because applications must decide, at the time of usage, whether a certificate is acceptable or not to perform an action. Proposals in the literature suggest the use of caching mechanisms to let the node itself or a neighbour node to store status checking material (typically on-line status responses or lists of revoked certificates). However, to the best of our knowledge the only criterion to evaluate the cached (obsolete) material is the time. In this paper we propose and formulate a new criterion based on risk to evaluate cached status checking data that is much more appropriate and absolute than time because it takes into account the revocation process. The rest of the paper is organized as follows: Section 2 presents an analysis of the main certification approaches for MANET. Section 3 discusses the main issues that have to be solved in order to adapt current PKI status checking mechanisms to MANET. In Section 4, we present our proposal to evaluate cached status data and, finally, we conclude in Section 5.

2 Certificate Management schemes for MANET

In general, certificate management schemes can be classified as:

- Decentralized. The nodes of the MANET participate either fully or partially in the certification process (see Figure 1.b).
- Centralized. Authorities outside the MANET control the certification process according to a global policy (see Figure 1.a).

In the fully decentralized PKI schemes for MANET, like Capkun et al. [3, 4], the nodes of the MANET themselves issue, publish and revoke the certificates. The certificate management is autonomous and self-organized because there is no need for any trusted authority or fixed server and all the nodes have the same role. In this system, like in PGP (Pretty Good Privacy) [5], each user is her own issuer. Certificates are stored and distributed by the nodes in a fully self-organized manner. Each certificate is issued with a limited validity period and it contains its issuing and expiration times. Before a certificate expires, the owner can issue an updated version of the certificate, which contains an extended expiration time. Authors call this updated version the certificate update. Each node periodically issues certificate updates, as long as the owner considers that the user-key bindings contained in the certificate are correct. Trust is achieved via chains of certificates. The nodes build trust paths certifying from one

node to another, as in a friendship circle, forming an authentication ring to achieve the trust relationships with other nodes of the MANET. A decentralized trust management model for pervasive computing environments is presented in [6], where authors overcome the challenges posed by dynamic open environments, making use of the autonomy and cooperative behaviour of the entities.

Another group of public key schemes for MANET is based on threshold cryptography [2]. The idea behind these schemes is to distribute certification duties amongst network nodes. A (k, n) threshold scheme allows the signing private key to be split into n shares such that any k nodes could combine and recover the signing key for a certain threshold $k < n$, whereas $k-1$ or fewer nodes are unable to do so. In this manner, the signing key can be partitioned into n shares and distributed to n nodes using the previous cryptographic technique. For instance, any k of n nodes could then collaborate to sign and issue valid digital certificates or issue status data; whereas a coalition of $k-1$ or fewer nodes would not be able to do so. Notice that this scheme is partially decentralized because it requires an initialization phase in which a centralized authority assigns the role to the n nodes that will act as servers for certificate management. Partially decentralized schemes were first proposed by Zhou and Haas in [7]. This work inspired a practical system called COCA [8] in which a threshold cryptography scheme is implemented for infrastructure-based networks. On the other hand, another system called MOCA [9] extends this idea to ad-hoc networks. In this scheme security is improved by selecting powerful nodes as Certificate Authority servers.

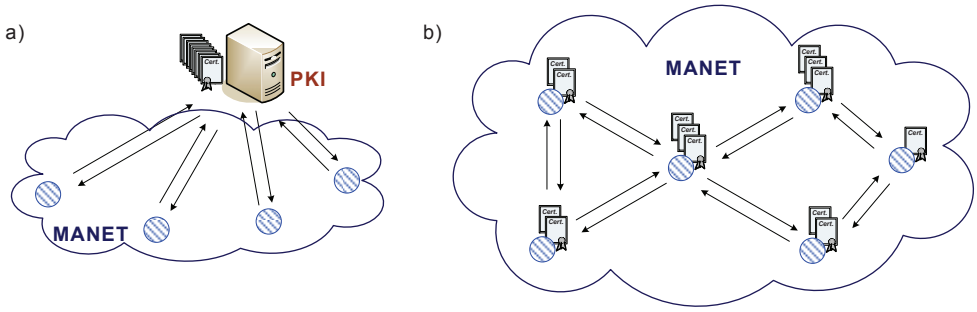


Fig. 1. Centralized and decentralized schemes

Finally, an external public key infrastructure can also be used for the hybrid scenario. In this case, centralized trusted authorities issue, publish and distribute the status (valid/revoked) of certificates according to a well-defined standard methodology. In the Internet, the PKIX [10] is the currently working public key infrastructure. However, PKIX is mostly designed for wired and well-connected networks and adapting the PKIX to the hybrid scenario is a challenging task because MANET nodes are expected to move across different networks, sometimes with on-line connection to the PKIX services and sometimes not. When the user is in a network with connection to the PKI, she can use all the PKI services such as getting a certificate, launching a status query, etc. However, users may be disconnected from the PKIX when they require real-time PKIX services. We discuss the problem of adapting PKI to MANET in more detail in the next section.

3 Adapting PKIX to MANET

The local validity of the certificates in the decentralized approaches may restrict their usability in the hybrid scenario. In this sense, the PKIX approach is suitable for hybrid MANETs that require support for mobility maintaining a centralized enforced certification mechanism and also interoperability with currently deployed PKIs. However, the original design of the PKIX assumes that the user can access at any time to the entities of the infrastructure which is true for wired well-connected networks but not for our scenario.

The first problem that we have to face is the certificate acquisition. A permanent connection of the client to the infrastructure cannot be assumed so the solution is to choose relatively long validity periods for the certificates. The idea is that the user has to pass an initial certification process before she can start operating in the MANET. Once the user has its credential, she can operate in the hybrid scenario without further interaction with the PKI (at least interaction is not required for a quite long time). This way of issuing the certificates can be assumed as an initialization phase equivalent to the initialization phase of the partially decentralized scheme in which the shares are delivered.

On the other hand, a certificate might be revoked (invalidated) prior to its expiration. Among other causes, a certificate may be revoked because of the loss or compromise of the associated private key, in response to a change in the owner's access rights, a change in the relationship with the issuer or as a precaution against cryptanalysis. The revocation policies

determine how the status of the certificates is distributed to the end users. So the PKI is responsible for the certificates not only at the issuing time but also during all the certificate's life-time.

The problem is that PKIX explicit revocation systems were designed for wired and well-connected networks in which repositories and responders have a well-known network address and are always available to users. However, MANETs are dynamic environments in which network topology changes randomly and in which mobile users continuously join and leave the network. Therefore, new mechanisms are necessary to distribute explicit status data in MANETs. Proposals in the literature suggest the use of caching mechanisms to address these problems.

Caching schemes allow to manage arbitrary disconnections between the users and the sources of the status data service. Disconnections are alleviated by storing copies of status data (lists of revoked certificates or on-line responses) in the nodes of the ad-hoc network. These copies are obtained when connection to the infrastructure is available. In general, an ad-hoc caching scheme for any service has four different kinds of nodes [11]: server-nodes, client-nodes, caching-nodes and intermediate-nodes (see Figure 2). For the status checking service:

- *Server-nodes.* These nodes have "always updated data" to offer the status checking service. The server-node has a permanent connection to the certification infrastructure in order to have always fresh status information. Typically, a server-node is an Access Point connected to both to a MANET and to the fixed network.
- *Client-nodes.* These nodes require the status checking service. A *service discovery* mechanism has to be provided to the client so that she can find a node in the network that provides the service.
- *Caching-nodes.* These nodes have cached data and therefore they may also provide the status checking service. A client-node in the absence of connectivity to a server-node or because of performance issues can connect with a close caching-node to obtain the service with cached status data (perhaps quite obsolete data).
- *Intermediate-nodes.* These nodes forward the packets among client and server nodes. They may also store the path to a service provider (whether a server-node or a caching-node) together with service parameters such as data size, the service expected Time-To-Live (TTL), number of hops to reach the provider etc.

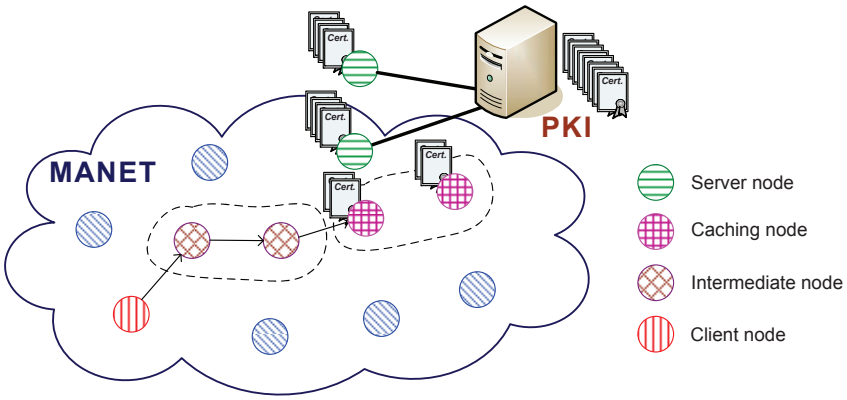


Fig. 2. Four different kinds of nodes in caching schemes.

In the literature we can find some proposals that apply the previous ideas to adapt the PKI status checking standards CRL [12, 13] and OCSP [14] to the MANET. A CRL is a black list with the identifiers of revoked certificates. The integrity and authenticity of the CRL is provided by an appended digital signature. On the other hand, OCSP is a protocol to make the status of certificates available through a request/response mechanism. The OCSP server is called responder and provides signed responses to clients. Next, we give our point of view about this adaptation and we briefly review some remarkable works about this in the literature.

In the case of CRL, server-nodes are nodes that can maintain a stable connection to PKI repositories in order to get the most updated CRL. A caching-node is a node that is willing to collaborate in the certificate status checking service and that has enough cache capacity to store a CRL copy. The caching-node responds to the status requests of client-nodes in the MANET. Notice that a client-node that acquires a valid CRL copy can become a new caching-node. Furthermore, a caching-node that moves to another MANET can collaborate in the new network to provide the service. In this sense, user's mobility helps the status checking service. In [15], the authors investigate the feasibility of using flooding to distribute CRL information in MANETs by simulation. They conclude that the two major factors for flooding to work smoothly are the number of nodes and the communication range. In [16] a MANET cooperative mechanism for certificate validation is presented in order to overcome both the lack of infrastructure and the limited capabilities of the nodes. This solution is

based on an extended-CRL where the repositories can build an efficient structure through an authenticated hash tree.

Regarding OCSP, server-nodes are responders. We can consider that there are only responders placed in the PKI (fixed-responders) or we can consider the possibility of having responders implemented in a mobile node that can be part of a MANET (mobile-responders). Despite this possibility, we discourage the use of mobile-responders because they are server-nodes and as such they are supposed to have updated status data. A server-node for certificate status checking must have connectivity with PKI repositories or fixed-responders to get updated status data but this connectivity is not always guaranteed in a MANET. On the other hand, a responder is a trusted authority so it has a private key that has to protect against intruders. In our view, it makes no sense having a server-node that is exposed to attacks and that may not have useful data. Furthermore, in general, increasing the number of trusted authorities in a system is not desirable, the less number of trusted authorities, the less is the probability of having a private key compromised. Besides, if mobile-responders are used, it is necessary to define a mechanism to trust them which is not trivial. With respect caching-nodes, they store OCSP responses issued by server-nodes and distribute them to client-nodes when they detect a request that fulfils freshness requirements. In [17, 18], there is a complete proposal called ADOPT (Ad-hoc Distributed OCSP for Trust) that describes a caching scheme for OCSP in MANET.

4 Evaluation of cached status data based on Risk

As explained in the previous section, caching and discovery mechanisms are necessary to manage the situation in which a user is not able to reach a PKI status data server. When a disconnection happens, the client-node uses service discovery to find a caching node. Then, the node obtains a cached version of available status data and finally, the node decides what to do with the data. In this sense, the CA issues status data bounded by two time-stamps:

- *thisUpdate.* Instant at which status data have been issued.
- *nextUpdate.* Instant at which updated status data are expected to be issued.

Let us define T_s as the issuing interval of status data (1).

$$T_s = nextUpdate - thisUpdate \tag{1}$$

As data in status responses are time-stamped, users can get an idea about how fresh is the status of a certificate by looking at the *thisUpdate* parameter of the response and, finally a user can take a decision about whether operate or not with a certain certificate. According to [19] the time is the only criterion to help the user to take this decision and to the best of our knowledge this is the only criterion proposed in the literature. However, this is a poor criterion that can be enhanced. In this section, we propose other parameter rather than time to take this decision.

First of all, let us illustrate why time is a poor parameter for our purposes. For instance, consider a status response issued a couple of hours ago. We may wonder: *is it fresh or not?* The answer is obviously that *"it depends"*. Two hours may not be considered a long time if there are a couple of revoked certificates every month but this period can be considered quite long if there are two new revoked certificates per hour. Moreover, a scenario with millions of issued non-expired certificates is not the same as a scenario that has hundreds of certificates. In the former, a couple of new revoked certificates is not so relevant while in the latter a couple of new revocations is quite important. As a conclusion, we need a parameter that considers all these aspects. For this purpose, we define a risk function that aids the user to decide whether to trust or not a certificate. We formally define the function *risk* ($r(t)$) as the *probability of considering a certificate as a valid one when the real status known by the PKI is revoked at time t*.

To find an analytical expression for the risk function we first need to analyse the certificate issuing process. Certificates are issued with a validity period T_c. Obviously $T_c >> T_s$, for instance T_c can be a year while the period of status data issuing can be an hour. The number of *non-expired certificates* ($N(t)$) -including revoked and non revoked certificates- is a stochastic process whose mean value at instant t depends on the certificate issue and certificate expiration processes. It is assumed that the elapsed time since issuing until expiration (T_c) is a constant value for all certificates. Therefore, the expiration process is the same as the issuance process elapsed T_c time units. This process is defined by the certificate issue rate λ_c, which matches with the certificate expiration rate. Hence the mean value of *non-expired certificates* in steady state is the mean quantity of issued certificates before the expiration process begins.

$$E[N(t)] = N = \lambda_C T_C, \quad t > T_C \tag{2}$$

On the other hand, there is a group of *revoked non-expired certificates*, that is to say, certificates that have a valid validity period but that have

been revoked prior to the expiration date and, therefore they are included in the black list. The subset of *revoked non-expired certificates* is included in the set of *non-expired certificates* and the cardinality of that set, $R(t)$, is a stochastic process that it is typically modelled [20] as a fraction or percentage ($p(t)$) of the non-expired certificates (3).

$$R(t) = p(t)N(t) \quad with\ p(t) \leq 1 \tag{3}$$

Assuming that both processes are independent and using expected values:

$$E[R(t)] = E[p(t)]E[N(t)] \tag{4}$$

$$R = pN \tag{5}$$

We further model the expected percentage of revoked certificates as directly proportional to the certification time T_c (6).

$$p = p'T_c \tag{6}$$

This means that larger certification periods will imply more percentage of revoked certificates. On the other hand, smaller certification periods mean less probability of a certificate being revoked during its life-time and therefore low percentage of revoked certificates. Then, the mean value of the *revoked non-expired certificates* can be expressed as:

$$R = p'\lambda_c T_c^2 \tag{7}$$

We have modelled the issuing and revoking processes of the overall system. However, our goal is to model the risk from the point of view of the user, that is to say, we want to find the probability of considering a certificate as a valid one when the real status known by the PKI is revoked.

Let us assume, without loss of generality, that at instant $t_0 = thisUpdate$ a user gets the current black list of revoked certificates from the PKI. Using this list, the user can split the set of *non-expired certificates* into *revoked certificates* and *not revoked certificates*.

Next, we need to define the subset of *operative certificates* as the group of *non-expired certificates* for which the last status known by a user is *not revoked*. Notice that the PKI may know that a certificate considered operative by a user is in fact revoked. However, due to the MANET conditions it is impossible to communicate this situation to the user.

Now, let us assume that the user is not able to connect to the infrastructure any more. As time goes by the set of *operative certificates* will include revoked certificates and the user will need to take decisions about using an operative certificate assuming a certain risk. The *risk function* $r(t)$ can be evaluated as the ratio between the number of *unknown revoked operative certificates* ($R'(t)$) and the number of *operative certificates* ($N'(t)$) as shown in equation (8).

$$r(t) = \frac{E[R'(t)]}{E[N'(t)]} \tag{8}$$

$N'(t)$ (*number of operative certificates*) can be defined as the number of certificates that were not included in the last black list obtained by the user (were not revoked before t_0) and that they have not expired at t. Included in the set of *operative certificates* there is the subset of *unknown revoked operative certificates*. The cardinality of this subset $R'(t)$ is the number of *operative certificates* that are revoked at instant t, that is, they are revoked but this fact is unknown to the user.

At $t_0 = thisUpdate$ the set of *operative certificates* is the same that the set of *not revoked certificates* and, since the user has the same information that the PKI so there is no risk ($r(t_0) = 0$). Besides

$$E[N'(t_0)] = (1-p)N \tag{9}$$

$$E[R'(t_0)] = 0 \tag{10}$$

At the instant $t_0 + T_C$ all the certificates included in the black list will be expired. This means that all *non expired certificates* will be *operative*, and any revoked certificate will be unknown to the user. The *risk* at this moment can be expressed as (11).

$$r(t_0 + T_C) = \frac{E[R'(t_0 + T_C)]}{E[N'(t_0 + T_C)]} = \frac{E[R(t_0)]}{E[N(t_0)]} = p \tag{11}$$

To evaluate the function risk between t_0 and $t_0 + T_C$ we have to observe the processes $N'(t)$ and $R'(t)$ in this interval. After t_0 the variation of the number of *operative certificates* ($N'(t)$) depends on these factors:

- Increases because of the new issues.
- Decreases because of the expiration of operative certificates issued before instant t_0 (the certificates issued later do not expire in the considered interval).

The issuance rate is λ_c that is the same as the expiration rate. But notice that not all expirations concern to *operative certificates*. A fraction p of the expirations corresponds to *revoked non expired certificates*, and the other fraction $1 - p$ corresponds to *operative certificates*. Then the expiration rate of *operative certificates* is $(1 - p)\lambda_c$ (see Figure 3).

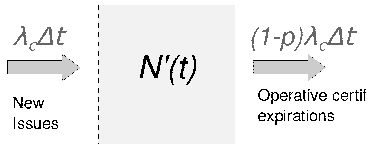


Fig. 3. Evolution of operative certificates

Considering the evolution of the set of *operative certificates* we can evaluate its expected cardinal (12).

$$E[N'(t)] = E[N'(t_0)] + \lambda_C(t - t_0) - (1 - p)\lambda_C(t - t_0) \tag{12}$$

Using (9) we obtain.

$$E[N'(t)] = (1 - p)N + p\lambda_C(t - t_0) \tag{13}$$

Finally, we need an expression for the set of *revoked operative certificates*. This set is the intersection of the set of *operative certificates* and the set of revoked certificates as shown in the Figure 4.

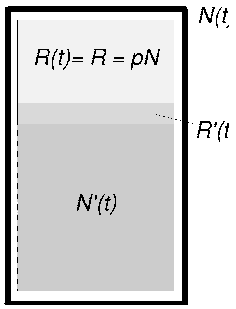


Fig. 4. Sets of certificates

Hence we can express the cardinality of these sets using the following expression.

$$N(t) = R(t) + N'(t) - R'(t) \tag{14}$$

Therefore,

$$R'(t) = R(t) + N'(t) - N(t) \tag{15}$$

We obtain the expected value of the number of revoked operative certificates using (15), (2), (5) and (13).

$$E[R'(t)] = p\lambda_C(t - t_0) \tag{16}$$

To obtain the *risk* function we use the expressions (13), (16) and the expression of its definition (8).

$$r(t) = \frac{p(t - t_0)}{(1 - p)T_c + p(t - t_0)} \tag{17}$$

The previous expression is valid for instants of time $t \ \epsilon \ t_0 \leq t \leq t_0 + T_c$ and fulfils with the expected results of expressions (10) and (11). Notice that the risk function allows a user to compute the probability of considering a non-expired certificate as non-revoked when the real status known by the PKI is revoked.

On the other hand, it is remarkable that unlike time which is a relative parameter, the risk function gives the user an absolute parameter to aid her taking the decision of trusting or not a particular certificate. This decision must be taken when the user is disconnected from the infrastructure and therefore it is taking into consideration cached (obsolete) status data.

Finally, the risk function should be used as follows:

- In first place, the CA signs the status data with the two standard time-stamps (*thisUpdate* and *nextUpdate*) but it also adds the current parameter p. The CA can calculate this parameter because it knows the current number of issued non-expired certificates and the current number of non-expired revoked certificates.
- When the user has to evaluate status data, she knows T_c as this is the certification period included in her certificate.
- Then, the user obtains p from the status data.

– Next, the user can compute the risk at current time t by replacing t_0 with *thisUpdate* in the risk function.
– Finally, the user can take a decision about a target certificate with the risk value computed.

5 Conclusions

Decentralized certification architectures for MANET such as self-organized PKIs and PKIs based on threshold cryptography generally provide certificate validation mechanisms inside the MANET. However, local validity of the certificates and inter-operability with currently deployed PKIs may restrict their usability in an hybrid MANET scenario. If a centralized certification infrastructure such as PKIX is used, then certificate validation becomes one of the main problems. This is because users need to ensure at the time of usage that the certificate they are relying upon has not been revoked but at the same time trusted servers of PKIX may be unavailable. Besides, standard status checking mechanisms of the fixed network are not directly usable because they are designed for always connected users.

In this sense, caching schemes allow to manage arbitrary disconnections between the users and the sources of the status data service. Disconnections are alleviated by storing copies of status data (lists of revoked certificates or on-line responses) in the nodes of the ad-hoc network. These copies are obtained when connection to the infrastructure is available. On the other hand, a service discovery mechanism is necessary to find the nodes that have cached material. In this paper, we have reviewed and analysed all these issues for adapting the standard PKIX status checking mechanisms to hybrid MANET.

Despite the caching scheme allows the users to obtain status data during disconnections, the cached status data is likely to be outdated. When using cached status data a node could operate with a revoked certificate considering it is a valid one. In this paper, we have presented a novel scheme which provides users within the MANET with an absolute criterion to determine whether to use or not a target certificate when updated status data is not available. By taking into account information about the revocation process, users can calculate a *risk* function in order to estimate whether a certificate has been revoked while there is no connection to a status checking server. Finally, it is also worth to mention that this new criterion can be applied to other networks than hybrid MANETs if these networks are based on an off-line explicit revocation scheme.

Abbreviations

ADOPT Ad-hoc Distributed OCSP for Trust.
CA Certification Authority.
COCA Cornell On-line Certification Authority.
CRL Certificate Revocation List.
MANET Mobile Ad-hoc Network.
MOCA Mobile Certificate Authority.
OCSP On-line Certificate Status Protocol.
PGP Pretty Good Privacy.
PKI Public Key Infrastructure.
PKIX Public Key Infrastructure (X.509).
TTL Time-To-Live.
TTP Trusted Third Party.

References

1. S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501 (Informational), January 1999.
2. Y. Desmedt and Y. Frankel. Threshold cryptosystems. in advances in cryptology—crypto'89. In *the Ninth Annual International Cryptology Conference*, volume 435 of *LNCS*, pages 307–315. Springer-Verlag, 1989.
3. S. Capkun, L. Buttyan, and J.P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2003.
4. J-P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC'01)*, 2001.
5. J. Zsako. PGP Authentication for RIPE Database Updates. RFC 2726 (Proposed Standard), December 1999.
6. F. Almenárez, A. Marín, C. Campo, and C. García. Managing ad-hoc trust relationships in pervasive environments. In *Proceedings of the Workshop on Security and Privacy in Pervasive Computing SPPC*, 2004.
7. L. Zhou and Z.J. Haas. Securing ad hoc networks. *IEEE Networks*, 13(6):24–30, 1999.
8. L. Zhou, F.B. Schneider, and R.V. Renesse. Coca: A secure distributed on-line certification authority. *ACM Transactions on Computer Systems*, 20(4):329–368, 2002.
9. S. Yi and R. Kravets. Moca: Mobile certificate authority for wireless ad hoc networks. In *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP'O2)*, 2002.
10. Pkix chapter of the ietf. www.ietf.org/html.charters/pkix-charter.html.
11. L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(1):77–89, 2006.
12. R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC 2459 (Proposed Standard), January 1999. Obsoleted by RFC 3280.

13. S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile. RFC 3820 (Proposed Standard), June 2004.
14. M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560 (Proposed Standard), June 1999.
15. H. W. Go, P. Y. Chan, Y. Dong, A. F. Sui, S. M. Yiu, Lucas C. K. Hui, and Victor O. K. Li. Performance evaluation on crl distribution using flooding in mobile ad hoc networks (manets). In *ACM Southeast Regional Conference archive. Proceedings of the 43rd annual southeast regional conference*, volume 2, pages 75–80, Kennesaw, Georgia, 2005.
16. J. Forné, J. L. Muñoz, O. Esparza, and F. Hinarejos. Certificate status validation in mobile ad hoc networks. *IEEE Wireless Communications*, 16(11):55–62, 2009.
17. G. F. Marias, K. Papapanagiotou, V. Tsetsos, O. Sekkas, and P. Georgiadis. Integrating a trust framework with a distributed certificate validation scheme for manets. *Wireless Communications and Networking*, 1155(10):1–18, 2006.
18. G. F. Marias, K. Papapanagiotou, V. Tsetsos, O. Sekkas, and P. Georgiadis. Integrating a trust framework with a distributed certificate validation scheme for manets. *EURASIP Journal on Wireless Communications and Networking*, 2006(2):1–18, 2006.
19. A. Deacon and R. Hurst. The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments. RFC 5019 (Proposed Standard), September 2007.
20. A. Arnes. Public key certificate revocation schemes, February 2000. Queen's University. Ontario, Canada. Master Thesis.

References

[1] WhichSSL, "SSL Market Share," 2010, [Online] Available: http://www.whichssl.com/ssl-market-share.html.

[2] "Car2car Communication Consortium," [Online] Available: http://www.car-to-car.org.

[3] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.*, vol. 15, pp. 39–68, January 2007.

[4] A. S. for Testing and M. (ASTM), "Standard specification for telecommunications and information exchange between roadside and vehicle systems-5ghz band dedicated short range communications (DSRC) medium access control (MAC) and physical layer (PHY) specifications," ASTM International, Technical Report ASTM E2213 - 03(2010), 2010.

[5] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, ser. SASN '05, 2005, pp. 11–21.

[6] J. P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *Security Privacy, IEEE*, vol. 2, no. 3, pp. 49 –55, May 2004.

[7] P. Papadimitratos, L. Buttyan, J. P. Hubaux, F. Kargl, A. Kung, and M. Raya, "Architecture for secure and private vehicular communications," in *Telecommunications, 2007. ITST '07. 7th International Conference on ITS*, Jun. 2007, pp. 1 –6.

[8] "IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages," *IEEE Std 1609.2-2006*, pp. 1–105, 2006.

[9] J. Iliadis, D. Spinellis, D. Gritzalis, B. Preneel, and S. Katsikas, "Evaluating certificate status information mechanisms," in *Proceedings of the 7th ACM conference on Computer and communications security*, ser. CCS '00. New York, USA: ACM, 2000, pp. 1–8.

[10] G. F. Marias, K. Papapanagiotou, and P. Georgiadis, "ADOPT. a distributed OCSP for trust establishment in MANETs," in *11th European Wireless Conference 2005*, 2005.

[11] T. P. Hormann, K. Wrona, and S. Holtmanns, "Evaluation of certificate validation mechanisms," *Comput. Commun.*, vol. 29, pp. 291–305, February 2006.

[12] A. Arnes, M. Just, S. J. Knapskog, S. Lloyd, and H. Meijer, "Selecting revocation solutions for PKI," in *NORDSEC '00*, 2000.

[13] D. Walleck, Y. Li, and S. Xu, "Empirical analysis of certificate revocation lists," in *Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security*, 2008, pp. 159–174.

[14] C. Ma, N. Hu, and Y. Li, "On the release of CRLs in public key infrastructure," in *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*. Berkeley, CA, USA: USENIX Association, 2006.

[15] N. Hu, G. K. Tayi, C. Ma, and Y. Li, "Certificate revocation release policies," *J. Comput. Secur.*, vol. 17, pp. 127–157, April 2009.

[16] G. E. P. Box and G. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[17] A. Wasef and X. Shen, "EDR: Efficient Decentralized Revocation Protocol for Vehicular Ad Hoc Networks," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 9, pp. 5214 –5224, nov. 2009.

[18] K. P. Laberteaux, J. J. Haas, and Y.-C. Hu, "Security certificate revocation list distribution for VANET," in *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, ser. VANET '08, 2008, pp. 88–89.

[19] P. Papadimitratos, G. Mezzour, and J.-P. Hubaux, "Certificate revocation list distribution in vehicular communication systems," in *Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*, ser. VANET '08, 2008, pp. 86–87.

[20] J. L. Muñoz, O. Esparza, C. Gañán, and J. Parra-Arnau, "PKIX certificate status in hybrid MANETs," in *Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks (WISTP)*, ser. Lecture Notes in Computer Science, vol. 5746. Springer, 2009, pp. 153–166.

[21] B. Fox and B. A. LaMacchia, "Certificate Revocation: Mechanics and Meaning," in *International Conference on Financial Cryptography (FC98)*, no. 1465, Feb. 1998, pp. 158–164.

[22] R. Rivest, "Can we eliminate certificate revocation lists?" in *International Conference on Financial Cryptography*. Springer-Verlag, 1998, pp. 178–183.

[23] P. McDaniel and A. Rubin, "A response to can we eliminate certificate revocation lists," in *International Conference on Financial Cryptography 2000 (FC00)*. Springer-Verlag, Feb. 2000.

[24] J. J. Haas, Y.-C. Hu, and K. P. Laberteaux, "Design and analysis of a lightweight certificate revocation mechanism for VANET," in *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking*, ser. VANET '09. New York, NY, USA: ACM, 2009, pp. 89–98.

[25] B. of Transportation Statistics U.S. Department of Transportation, "Number of U.S. aircraft, vehicles, vessels, and other conveyances," 2009, [Online] Available: http://www.bts.gov/publications/national_transportation_statistics/html/table_01_11.html.

[26] D. N. Cottingham, I. J. Wassell, and R. K. Harle, "Performance of IEEE 802.11a in vehicular contexts," in *Proc. IEEE VTC*. Spring, 2007.

[27] R. Merkle, "A certified digital signature," in *Proceedings of Advances in Cryptology (CRYPTO' 89)*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 1990, vol. 435, pp. 218–238.

[28] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP," Internet Engineering Task Force, RFC 2560, Jun. 1999. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2560.txt

[29] "ITU-T X.509 : Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks," International Telecommunication Union, 2005.

[30] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, pp. 770–772, November 1981.

[31] C. Gañan, J. L. Muñoz, O. Esparza, J. Loo, J. Mata-Díaz, and J. Alins, "Efficient Certificate Status Information distribution mechanism," *Mobile Information Systems*, pp. 1–31, 2013, (in press). [Online]. Available: http://dx.doi.org/10.3233/MIS-130167

[32] S.-Y. Wang and C.-L. Chou, "NCTUns tool for wireless vehicular communication network researches," *Simulation Practice and Theory*, vol. 17, pp. 1211–1226, 2009.