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).

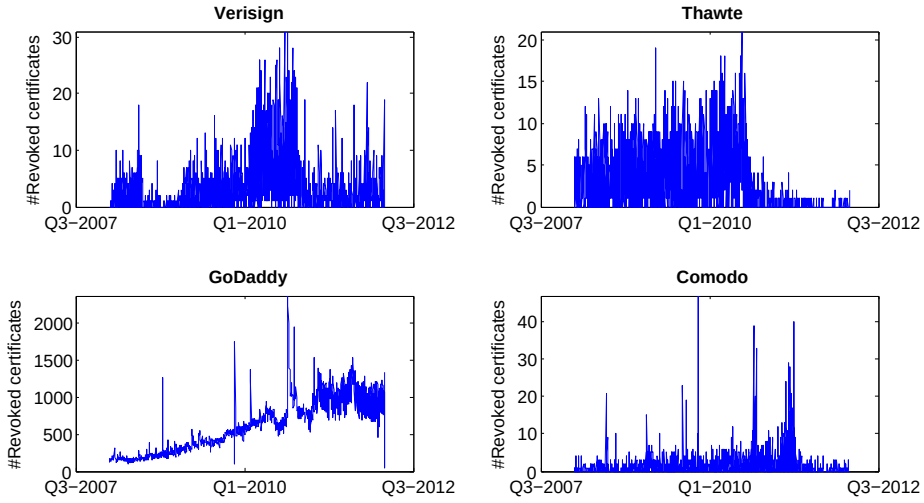


Fig. 1: Number of daily revoked certificates evolution for each CA.

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.

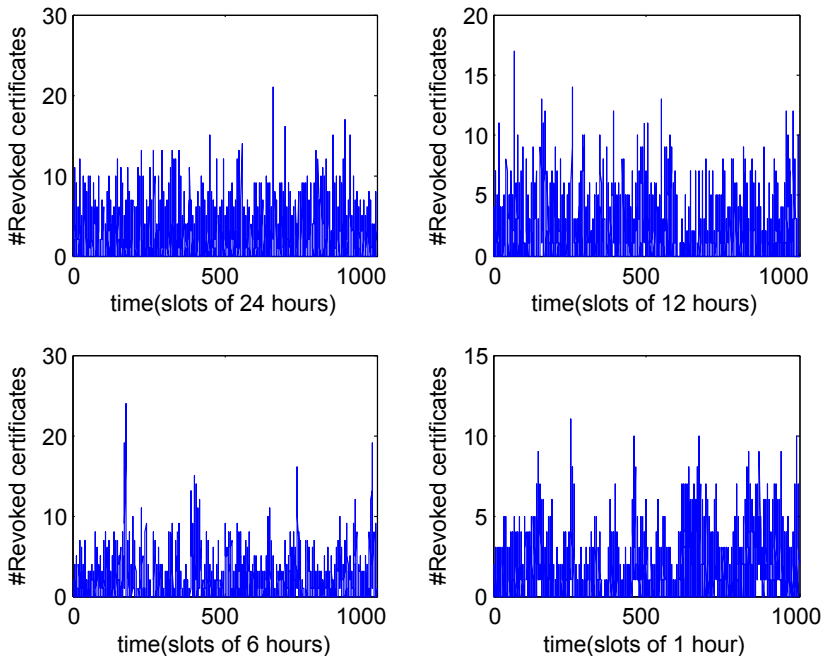


Fig. 2: Revocation Bursts over Four Orders of Magnitude.

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.

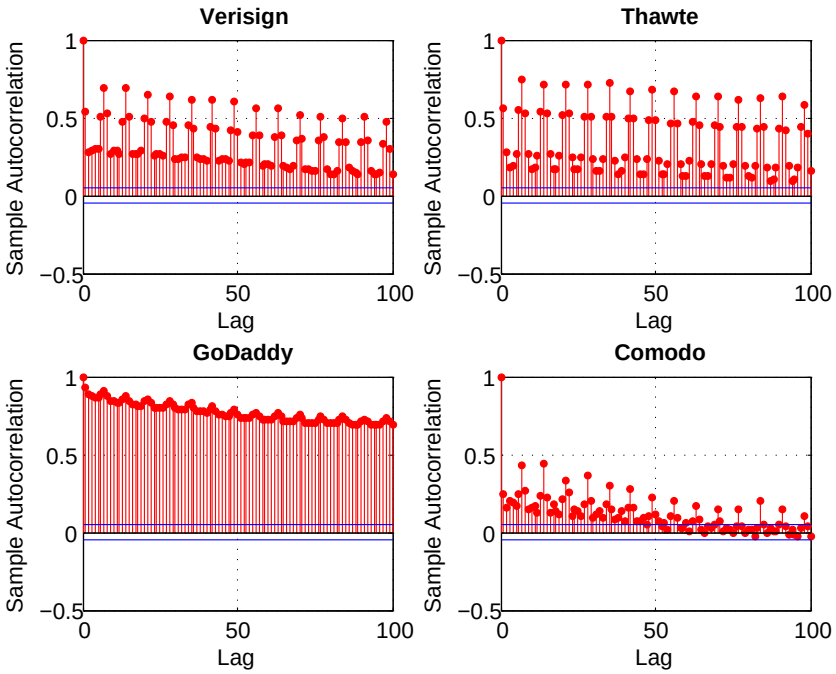
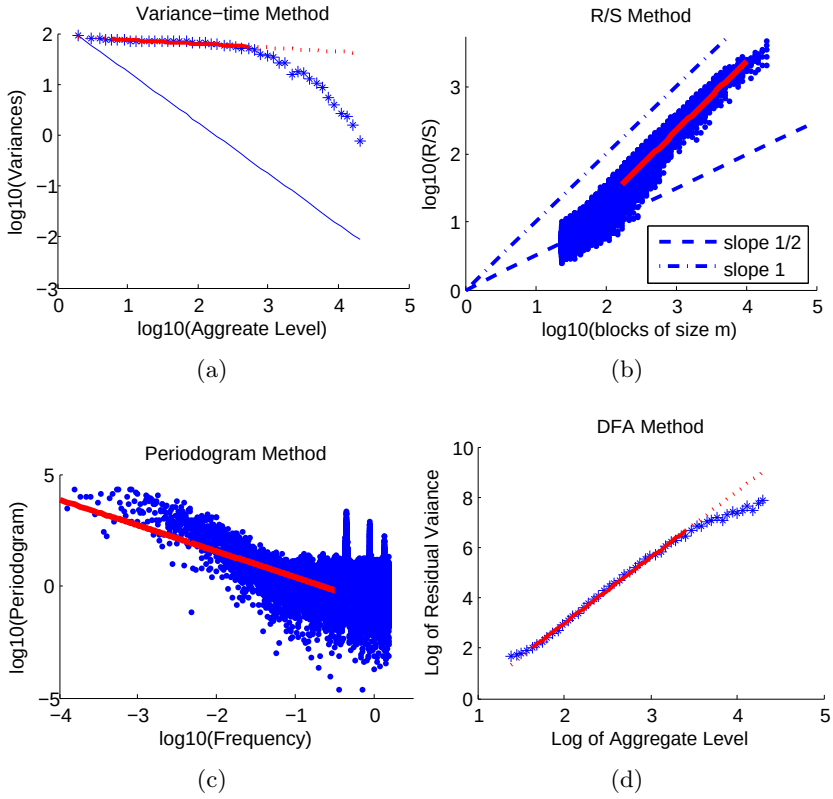


Fig. 3: Autocorrelation function of the revocation process per CA.

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.



(a) (b) (c) (d)

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 $(0.87, 0.95)$.

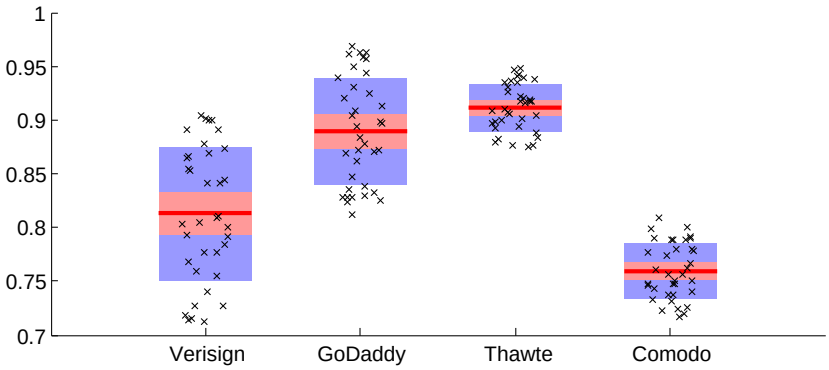


Fig. 5: Summary plot of estimates of the Hurst parameter H for all the CAs.

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.

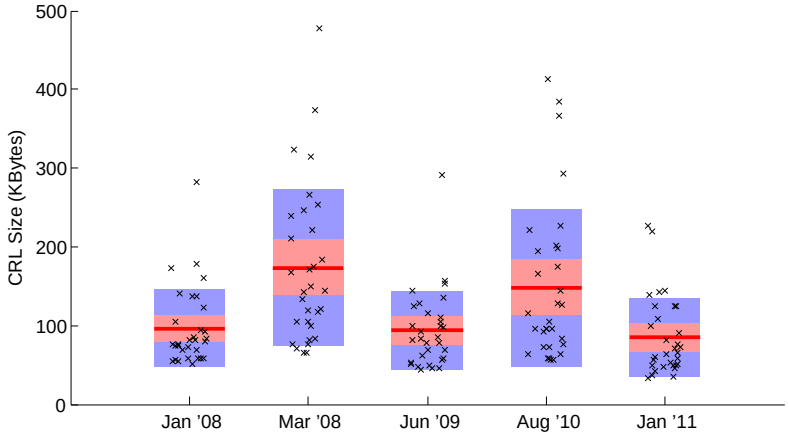


Fig. 6: Estimated daily size of Verisign's CRL.

[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:

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

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.

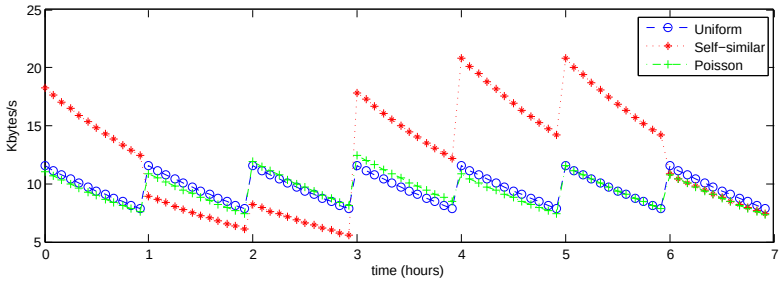


Fig. 7: Delta-CRL BW consumption.

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.