A Model of Certificate Revocation[*]

David A. Cooper
Computer Security Division
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930
david.cooper@nist.gov

Abstract

This paper presents a model for the distribution of revocation information using certificate revocation lists (CRLs). This model is used to highlight inefficiencies in the "traditional" method of distributing certificate status information using CRLs. Two alternative CRL-based revocation distribution mechanisms, over-issued CRLs and segmented CRLs, are then presented. The original model is then expanded to encompass each of the alternative mechanisms and these expanded models are used to demonstrate the advantages of the alternative mechanisms to the "traditional" method. Finally, the paper offers some suggestions for choosing the best CRL-based revocation distribution mechanism for any particular environment.

1. Introduction

Public key infrastructures (PKIs) are being fielded in increasing size and numbers, but operational experience to date has been limited to a relatively small number of environments. As a result, there are still many unanswered questions about the ways in which PKIs will be organized and operated in large scale systems. Some of these questions involve the ways in which individual certification authorities (CAs) will be interconnected. Others involve the ways in which certificate status information will be distributed. In a 1994 report, the MITRE Corporation suggested that the distribution of revocation information has the potential to be the most costly aspect of running a large scale PKI [1].

The MITRE report assumed that each CA would periodically issue a certificate revocation list (CRL) that listed all of the unexpired certificates that it had revoked. Since the MITRE report was published, several alternative certificate status distribution mechanisms have been proposed. Each of these mechanisms has its own relative advantages and disadvantages in comparison to the other schemes. NIST is working to create mathematical models of some of the proposed certificate status distribution mechanisms. These models could be used to determine the circumstances under which each of the mechanisms is most efficient.

[*]This paper appears in the *Proceedings of the Fifteenth Annual Computer Security Applications Conference*, pages 256-264, December 1999.

Most of the proposals have involved variations of the original CRL scheme. Examples include the use of segmented CRLs and delta-CRLs [2]. However, some schemes do not involve the use of any type of CRL (e.g., on-line certificate status protocols [4] and hash chains [3]).

This paper presents models for various methods of distributing certificate status information using CRLs. In modeling the various methods, it is assumed that relying parties request CRLs only when needed to perform a validation (i.e., no pre-caching of CRLs) and that they have perfect caches (i.e., no CRLs are deleted from the cache until they have expired). For each option, the request rate is computed for CRLs as a function of time. As this paper shows, request rates vary over time. However, in building a repository for a PKI, it is necessary to build one that is capable of handling incoming requests, even when the request rate is at its peak, without unreasonable response times. As such, this paper focuses on distribution methods that minimize peak loads on repositories as opposed to methods that minimize average loads.

The paper begins by presenting a model for the "traditional" method of distributing certificate status information using CRLs. This model is then used to point out some of the deficiencies inherent in the "traditional" method. Finally, models for two alternative CRL-based certificate status distribution methods are presented and the manner in which these methods overcome some of the deficiencies in the "traditional" method is described.

2. A model for the traditional method

The traditional method of distributing certificate status information involves a CA periodically issuing a CRL,

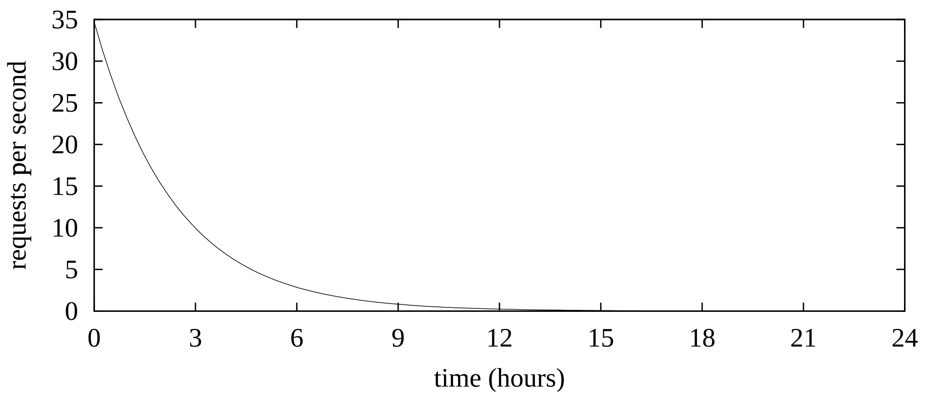

Figure 1. Unsegmented CRL

which it posts to a repository. The CRL includes all unexpired certificates issued by the CA that have been revoked. Each CRL includes a **nextUpdate** field that specifies the time at which the next CRL will be issued. Any relying party requiring certificate status information, that does not already have the current CRL, retrieves the current CRL from the repository. In order to enhance performance, copies of the CRL may be distributed to several sites. The model in this paper, however, assumes that all relying parties obtain CRLs from the same repository[1].

In order to examine the load on the repository from CRL requests, the rate at which requests for CRLs are made over time must be determined. With the traditional method, every relying party needs the most current CRL in order to perform a validation. Once a relying party has retrieved the most current CRL, it will not need to request any more information from the repository until a new CRL has been issued. This is the case with the traditional method since each CRL specifies the status of all unexpired certificates issued by the CA and each CRL includes a **nextUpdate** field stating when the next CRL will be issued. As a result, over the period of time in which a CRL is valid (i.e., the most current), each relying party will make at most one request to the repository for a CRL. This request will be made the first time after the current CRL is issued that the relying party performs a validation.

In order to compute the overall CRL request rate, the probability density function for validation attempts for a single relying party must be known. In particular, if a CA issues a new CRL at time 0, it must be determined, for any time t, the probability that a relying party will perform its first validation after time 0 at time t. If the number of relying parties is reasonably large, then it can be assumed that the times at which validation attempts are made are independent of each other (i.e., they occur at randomly distributed times). Based on this assumption, an exponential interarrival probability density [5] can be used to model the timing of validation attempts. In this model, the probability that a relying party's first validation attempt will occur in the interval $[t \ldots t + dt]$, in the limit $dt \to 0$, is

$$ve^{-vt}dt \tag{1}$$

where v is the validation rate (i.e., average number of certificates per unit time that a relying party attempts to validate). Since each relying party downloads a CRL at the time of its first validation attempt after time 0, this equation also represents the probability that any given relying party will send a request to the repository for a CRL in the interval $[t \ldots t + dt]$. If this equation is multiplied by the number of relying parties, N, and divided by dt, the result is the request rate for CRLs from the repository at time t:

$$R(t) = Nve^{-vt} \tag{2}$$

Figure 1 shows the request rate for a CRL, issued using the traditional method, over the course of 24 hours. The graph in figure 1 was drawn assuming that a CRL was issued at time 0 and that no other CRLs were issued during the period of time shown in the graph. It was also assumed that there are 300,000 relying parties each validating an average of 10 certificates per day.

3. Over-issued CRLs

The graph in figure 1 makes the problem with the traditional method of issuing CRLs apparent. The problem is that the CRLs cached by every relying party expire at the same time. Immediately after the CRLs expire, and a new CRL is issued, every relying party will need to obtain a CRL

[1]If more than one repository is used, then the load on each repository could be approximated by dividing the number of relying parties by the number of repositories.

cRLNumber = 1
thisUpdate = Mon. 12:00am
nextUpdate = Tues. 12:00am

cRLNumber = 2
thisUpdate = Mon. 6:00am
nextUpdate = Tues. 6:00am

cRLNumber = 3
thisUpdate = Mon. 12:00pm
nextUpdate = Tues. 12:00pm

cRLNumber = 4
thisUpdate = Mon. 6:00pm
nextUpdate = Tues. 6:00pm

cRLNumber = 5
thisUpdate = Tues. 12:00am
nextUpdate = Wed. 12:00am

cRLNumber = 6
thisUpdate = Tues. 6:00am
nextUpdate = Wed. 6:00am

Figure 2. Over-issued CRLs

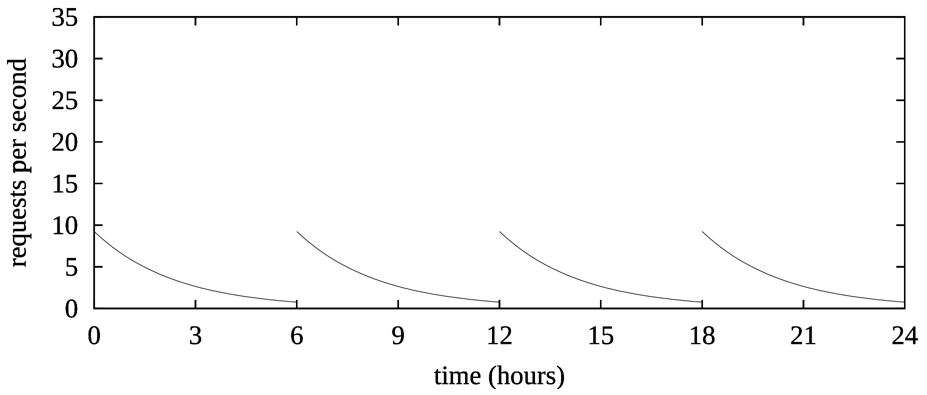

Figure 3. Request rate for over-issued CRLs

from the repository in order to perform a validation. As a result, there is a relatively high request rate when a new CRL is issued followed by an exponential decline in the request rate. If CRLs are valid for a reasonably long period of time, then there will be periods of time in which the repository is practically unused. In the example depicted in figure 1, the request rate begins at 34.72 requests/second and then drops to 0.24 requests/second after 12 hours. After 24 hours, the request rate is only 1.6×10^{-3} requests/second.

One way to reduce the peak load on the repository is to spread out requests for CRLs. This can be accomplished by issuing CRLs in such a way that the CRLs in relying parties' caches do not all expire at the same time.

With the traditional method, a new CRL is not issued until the time specified in the **nextUpdate** field of the previously issued CRL has been reached. As a result, a relying party with a CRL in its cache will not request a new CRL from the repository until the time specified in the **nextUpdate** field of the cached CRL has been reached. If a new CRL is issued before the previous one expires (i.e., before the **nextUpdate** time of the previous CRL has been reached), then some relying parties will retrieve this new CRL while other relying parties continue to use the previously issued CRL. If each issued CRL is valid for the same length of time, then the relying parties that retrieved the new CRL will still have valid CRLs in their caches when the original CRL expires.

The idea of issuing a new CRL before the previously issued CRL has expired is not entirely new. Many people have considered the idea of issuing an "emergency" CRL whenever a certificate has been revoked as a result of a key compromise. If the validity period of this CRL is the same as that of the "regularly scheduled" CRLs, then the issuance of the "emergency" CRL may lead to a temporary reduction in the peak request rate when the next "regularly scheduled" CRL is issued. However, in order to maintain a consistently low peak request rate, CRLs must consistently be issued at a rate greater than that required by the validity periods of the CRLs.

Figure 2 shows an example of over-issued CRLs. In this figure, each CRL is valid for 24 hours, but a new CRL is issued every 6 hours. Figure 3 shows the request rate for CRLs over the course of 24 hours assuming CRLs are issued as in figure 2 and that there are 300,000 relying parties each validating an average of 10 certificates per day. By comparing figure 3 to figure 1, it can be seen that the result of over-issuing is to spread out the requests for CRLs, thus significantly reducing the peak request rate (from 34.72 requests/second to 9.25 requests/second). Of course, even in figure 3 the request rate varies over time. If CRLs were issued more frequently, requests would be spread out even more evenly and the peak request rate would be further re-

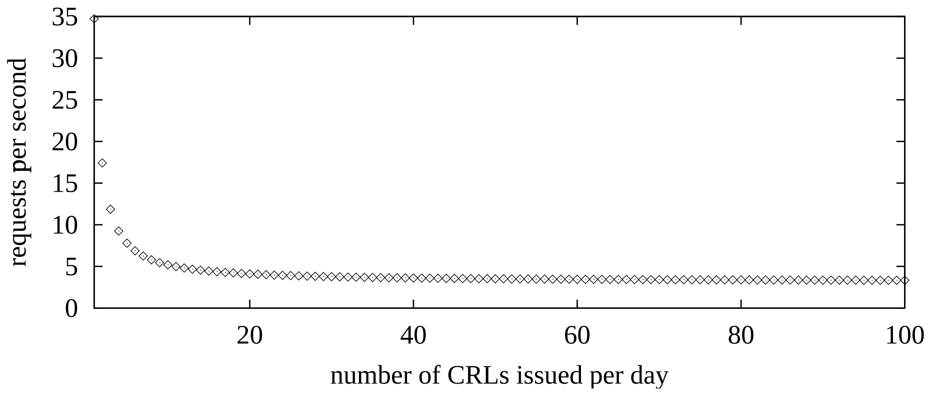

Figure 4. Peak request rate as a function of number of CRLs issued per day

duced. In the limit, if CRLs were issued continuously, the request rate in this example would be 3.16 requests/second. Figure 4 shows the peak request rate as a function of the number of CRLs that are issued per day. As can be seen, most of the advantages of over-issuing in this example scenario can be obtained by issuing CRLs once an hour.

3.1. A model of over-issued CRLs

The graphs in figures 3 and 4 are only illustrative of the benefits of over-issued CRLs. The degree to which over-issuing can reduce the peak request rate for CRLs depends on the validation rate of relying parties and the length of time that CRLs are valid. In general, higher validation rates and longer CRL validity periods lend themselves to greater relative improvements in the peak request rate as a result of over-issuing. In this section, a model for over-issued CRLs, which can be used to determine the benefits of over-issuing in any particular scenario, is presented.

Suppose that the period of time between the issuance of two CRLs is called an interval. In order to compute the request rate for CRLs over the course of an interval, the probability that a relying party will request a CRL during that interval needs to be determined.

As was described above, a relying party will only request a CRL from the repository in a given interval if it performs a validation in that interval and it does not have an unexpired CRL in its cache. If O represents the number of CRLs that are valid at any given time ($O = 4$ in figure 2) and P_{val} is the probability that a relying party will perform a validation in any given interval, then the probability that a relying party will request a CRL in interval n is P_{val} times the probability that the relying party did not request a CRL in any of the previous $O-1$ intervals:

$$P_{I,n} = P_{val}\left[1 - \sum_{j=n-O+1}^{n-1} P_{I,j}\right] \tag{3}$$

Once the system has reached a steady state, the probability that a relying party will request a CRL in an interval will be the same in each successive interval (i.e., $P_I = P_{I,n} = P_{I,n-1} = P_{I,n-2} = \ldots$). So, in the steady state

$$P_I = P_{val}[1 - (O-1)P_I] \tag{4}$$

Equation (4) can be solved for P_I:

$$P_I = \frac{P_{val}}{(O-1)P_{val} + 1} \tag{5}$$

If an interval begins at time 0, then the probability that a relying party will request a CRL from the repository between times t and $t+dt$, in the limit $dt \to 0$, is the probability that the relying party performs its first validation attempt of the interval between times t and $t + dt$ multiplied by the probability that the relying party does not have a valid CRL in its cache that it retrieved in a previous interval. The probability that the relying party performs its first validation attempt of the interval between times t and $t + dt$ is $ve^{-vt}dt$ (see equation (1)). The probability that the relying party does not have a valid CRL in its cache can be computed as the probability that the relying party will request a CRL during the interval divided by the probability that the relying party will perform a validation in the interval (i.e., equation (5) divided by P_{val}). Thus, the probability that the relying party will request a CRL between times t and $t+dt$ is

$$\frac{ve^{-vt}dt}{(O-1)P_{val} + 1} \tag{6}$$

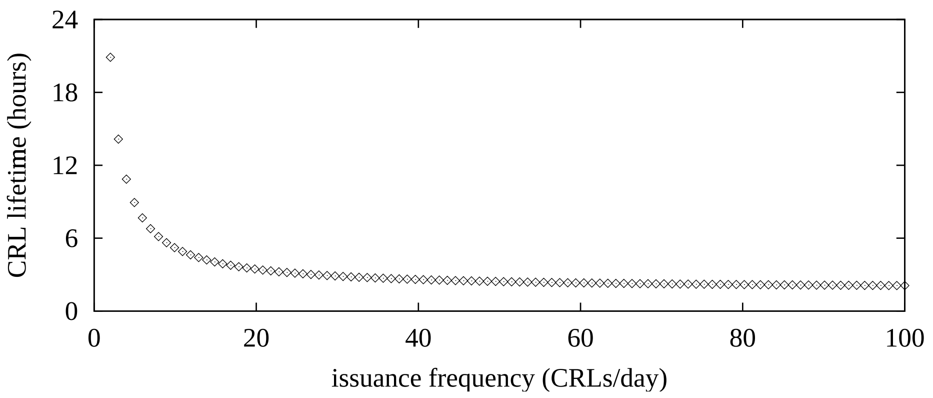

Figure 5. Minimum CRL lifetime as a function of CRL issuance frequency

If equation (6) is multiplied by the number of relying parties, N, and divided by dt, the result is the request rate for CRLs from the repository at time t:

$$R_I(t) = \frac{Nve^{-vt}}{(O-1)P_{val}+1} \tag{7}$$

Since validations follow an exponential interarrival probability distribution, the probability that a relying party will perform no validations during any given interval is $e^{-vl/O}$ where l is the length of time that a CRL is valid (i.e., an interval is of length l/O). Thus, $P_{val} = 1 - e^{-vl/O}$. So, the request rate for CRLs over the course of an interval is

$$R_I(t) = \frac{Nve^{-vt}}{(O-1)\left(1-e^{-vl/O}\right)+1} \tag{8}$$

and the peak request rate is

$$R_I(0) = \frac{Nv}{(O-1)\left(1-e^{-vl/O}\right)+1} \tag{9}$$

When O equals 1, equation (9) simplifies to the request rate for CRLs issued in the traditional manner, $R_I(0) = Nv$. At the other extreme, if CRLs are issued continuously, the request rate will be

$$\lim_{O\to\infty}\left[\frac{Nv}{(O-1)\left(1-e^{-vl/O}\right)+1}\right] = \frac{Nv}{vl+1} \tag{10}$$

Equation (10) represents the theoretical minimum peak request rate that can be achieved by over-issuing CRLs. While it is impossible to achieve this minimum rate, as illustrated in figure 4, one can get a peak rate close to the theoretical minimum with a moderate amount of over-issuing.

Equation (9) was derived under the assumption that the goal in choosing a revocation distribution mechanism is to minimize the peak request rate for CRLs from the repository while maintaining a given CRL lifetime, l. An alternative approach, however, is to assume that the repository can handle a given peak request rate and then use the equation to determine how short the CRL lifetime can be made. If, in equation (9), lf is substituted for O (where f represents the number of CRLs that are issued per unit time), then the equation can be solved for l, the minimum CRL lifetime that can be achieved if the peak request rate must be at most R:

$$l = \frac{Nv - Re^{-v/f}}{(1-e^{-v/f})Rf} \tag{11}$$

The graph in figure 5 shows an example of how CRL lifetime can be reduced as over-issuing is increased. In this figure, it is assumed that there are 300,000 relying parties each validating an average of 10 certificates per day and that the repository can handle at most 20 requests per second. As the graph shows, the more frequently CRLs are issued (i.e., the more they are over-issued) the more up-to-date relying parties can keep their caches while still not overloading the repository. In this particular example, if CRLs are only issued twice a day, the CRL lifetime must be set to at least 21 hours in order to prevent the repository from being overloaded. On the other hand, if CRLs are issued continuously, the CRL lifetime can be reduced to 1 hour and 46 minutes, thus ensuring that status changes are propagated more quickly.

4. Segmented CRLs

Another way to improve performance over the traditional method of distributing certificate status information is to segment CRLs. While segmenting CRLs may not reduce the peak request rate for CRLs, it will usually reduce the size of each CRL. This may allow a repository to service requests for CRLs at a faster rate.

In this section, a model for segmented CRLs is presented which is used to determine the effects of segmentation on request rates. In some cases, certificates may be allocated to CRL segments in a way that attempts to minimize the number of CRL segments that a relying party will need to download. The model in this paper, however, assumes that certificates are allocated to CRL segments at random. It is then assumed that each validation attempt is equally likely to require access to any of the CRL segments.

As in section 2, the probability density function for a single relying party with respect to a single CRL segment (e.g., segment 1) must be determined. If CRLs are not over-issued, and a new CRL for segment 1 was issued at time 0, then a relying party will request segment 1 from the repository in the interval $[t \dots t + dt]$ if and only if it attempts to validate a certificate in the interval $[t \dots t+dt]$ that requires the use of segment 1 and it has not validated any certificates in the interval $[0 \dots t]$ that required the use of segment 1.

First, the probability that a relying party will not have requested segment 1 in the interval $[0 \dots t]$ will be determined. Since an exponential interarrival probability for validation attempts is assumed, it is known from the Poisson law [5] that the probability that n validation attempts will be made during an interval of length t is

$$\left[\frac{(vt)^n}{n!}\right] e^{-vt} \tag{12}$$

If there are s CRL segments, then there is a probability of $1/s$ that segment 1 will be needed to perform any given validation attempt. Thus, the probability that segment 1 will not be needed for any of n validation attempts is

$$\left(1 - \frac{1}{s}\right)^n \tag{13}$$

Equations (12) and (13) can be combined to determine the probability that any given relying party will not request segment 1 during the interval $[0 \dots t]$:

$$\sum_{n=0}^{\infty} \left(1 - \frac{1}{s}\right)^n \left[\frac{(vt)^n}{n!}\right] e^{-vt} = e^{-vt/s} \tag{14}$$

Next, the probability that a relying party will need segment 1 during the interval $[t \dots t + dt]$ (in the limit $dt \to 0$) must be determined. The probability that one validation attempt will be made in the interval $[t \dots t + dt]$ is[2] $ve^{-v\,dt}dt = v\,dt$. Since the probability that any given validation attempt will require the use of segment 1 is $1/s$, the probability that segment 1 will be needed in the interval $[t \dots t + dt]$ is

$$\frac{v\,dt}{s} \tag{15}$$

Combining equations (14) and (15) and multiplying the result by the number of relying parties results in the total expected number of requests for segment 1 in the interval $[t \dots t + dt]$:

$$N'_s(t) = \frac{Nve^{-vt/s}dt}{s} \tag{16}$$

Dividing both sides of equation (16) by dt and multiplying by the number of segments results in the total request rate:

$$R_s(t) = \frac{s\,N'_s(t)}{dt} = Nve^{-vt/s} \tag{17}$$

Equation (17) shows how CRL request rates change with the amount of segmentation. Since $R_s(0) = Nv$, it is clear that the peak request rate is not affected by the amount of segmentation. Increasing the amount of segmentation only affects the rate at which the request rate drops off after a group of CRL segments have been issued. This can be seen by comparing figure 1 to figure 6.

5. Over-issued segmented CRLs

The reason that the peak request rate for segmented CRLs (equation (17)) is the same as the peak request rate for unsegmented CRLs (equation (2)) is that they both suffer from the same problem. In both models, all CRLs expire at the same time. As a result, there is a moment in time in which every relying party's cache is empty. At this moment, every validation attempt will result in a request for revocation information from the repository. Since the validation rate is Nv, this is the peak request rate.

Sections 3 and 4 have presented two techniques for improving the performance of repositories, with section 3 demonstrating how peak request rates can be reduced and section 4 demonstrating how the sizes of CRLs can be reduced without increasing the peak request rate. To a certain degree, the advantages of these two techniques can be combined. However, as the amount of segmentation increases, the degree to which over-issuing can reduce the peak request rate decreases.

There are two basic ways that over-issuing can be combined with segmentation. The first is to issue all CRL segments at the same time, but issue the CRL segments more

[2] Since the interval $[t \dots t + dt]$ is infinitesimally small, it can be assumed that the probability of more than one validation attempt occurring is 0.

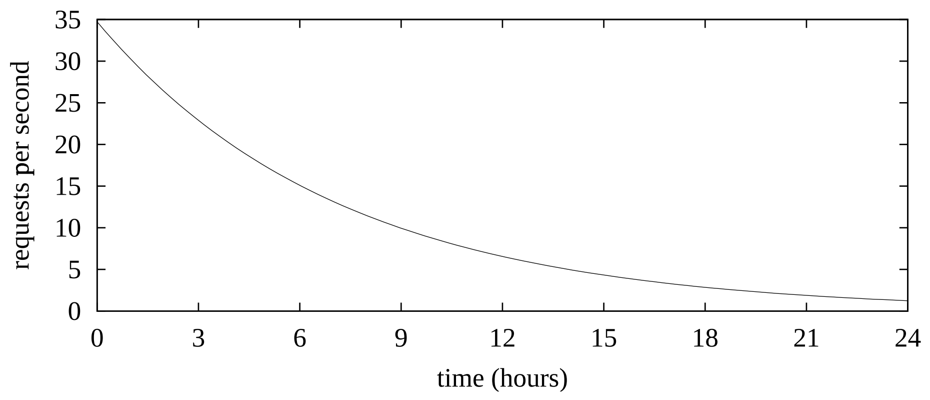

Figure 6. Request rate for three CRL segments

often than is required by the CRLs' validity periods. The other way is to issue each segment only as often as necessary, but to stagger the issuance of each segment so that the peak request rates for the different segments occur at different times.

In order to demonstrate the advantage of staggering the issuance of CRL segments, the scenario from figure 6 will be used as an example. The request rate for each segment at the time of issuance is 11.57 requests/second. However, after 8 hours the request rate has dropped to 3.81 requests/second and after 16 hours the request has dropped to 1.25 requests/second. So, if the issuance of the three CRL segments were staggered by 8 hours then the peak request rate would be only 16.64 requests/second as opposed to a peak rate of 34.72 requests/second if all three segments were issued at the same time (see figure 7). Unfortunately, the peak request rate does not continue to decline with increasing numbers of segments. With 4 CRL segments issued at 6 hour intervals, the peak request rate increases slightly to 17.15 requests/second. As the number of CRL segments approaches infinity, the peak request rate approaches the peak rate for an unsegmented CRL.

In general, the request rate for segmented CRLs issued at evenly separated intervals is

$$\frac{Nve^{-vt'/s}}{s}\sum_{i=0}^{s-1} e^{-ivl/s^2} = \frac{Nve^{-vt'/s}\left(1-e^{-vl/s}\right)}{s\left(1-e^{-vl/s^2}\right)} \tag{18}$$

where $t' = t \bmod \left(\frac{l}{s}\right)$.

If a CA is willing to over-issue each CRL segment, then the peak request rate can be reduced even further. Using the example from above, if three CRL segments are used and each segment is issued once every 8 hours, then the peak request rate will be only 14.83 requests/second. Furthermore, the more frequently the segments are issued, the more the peak request rate can be reduced. If the three CRL segments from the example above were issued continuously, then the peak request rate would be only 8.01 requests/second. As was described above, however, as the number of segments increases, the degree to which the peak request rate can be reduced also decreases. So, if revocation information is divided among a large number of segments in order to reduce the size of each CRL segment, then there may be no advantage, in terms of peak request rate, to over-issuing the CRL segments or to staggering the issuance of the segments.

In general, the request rate for over-issued segmented CRLs is

$$R_I(t) = \frac{Nve^{-vt/s}}{(O-1)\left(1-e^{-vl/sO}\right)+1} \tag{19}$$

where t is the amount of time since the latest CRL segments were issued. As in section 3, the lowest possible peak request rate can be determined by computing equation (19) in the limit as O approaches infinity. The resulting request rate is

$$R_I = \frac{Nvs}{vl+s} \tag{20}$$

6. Choosing a method for issuing CRLs

A number of factors must be taken into account when choosing the best method for distributing certificate status information using CRLs. In addition to the validation rates of relying parties and the validity periods of the CRLs, one must consider the expected number of revoked certificates and the environments in which relying parties will be operating.

All of the techniques described above rely on the re-use of cached information in order to reduce peak request rates.

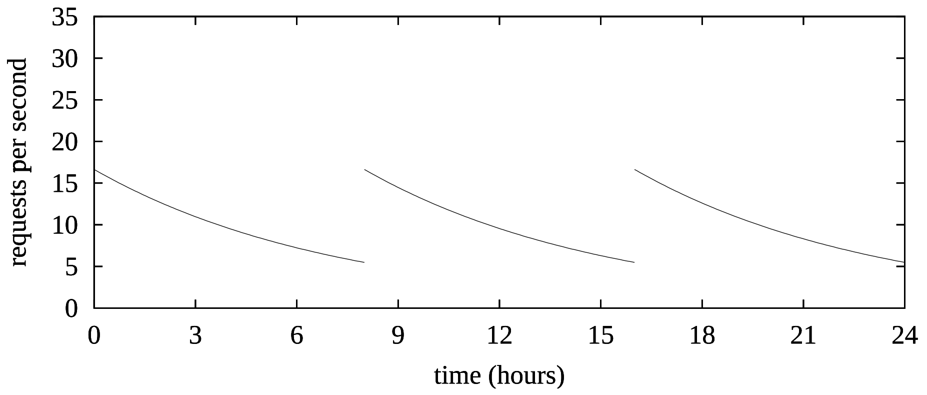

Figure 7. Request rate for three CRL segments with staggered issuance

If relying parties must obtain fresh revocation information from the repository each time a validation is performed, either because CRLs are valid for a very short period of time or the validation rates of the relying parties are relatively low, then none of the techniques described in this paper will be effective at reducing the peak request rate. As an example, if CRLs were valid for only 10 minutes instead of 1 day, then the lowest peak request rate that could be obtained by over-issuing for the example in section 3 would be 32.47 requests/second. This represents a mere 6.5% improvement over the traditional method as opposed to the 90.9% improvement that was possible when CRLs were valid for a day. The best solution in this case is to break up CRLs into as many segments as possible in order to minimize the time required to service a request.

If CRLs are valid for a sufficient length of time to make caching effective, then the best choice of certificate status distribution method will depend on the expected number of revoked certificates. If it is expected that very few certificates will be revoked, then segmentation will not be very effective in reducing the size of CRLs. This may be the case with an authority revocation list (ARL) since it is expected that certificates issued to certification authorities will rarely need to be revoked. For these environments, the best option is to issue unsegmented CRLs, but to over-issue them in order to reduce the peak request rate.

If there will be a large number of revoked certificates, then it may be more important to reduce the sizes of CRLs than to reduce the peak request rate. If this is the case, then segmentation should be used. If certificate status information is divided among a sufficiently large number of segments, however, then there is no need to over-issue the segments as over-issuing will not substantially reduce the peak request rate.

One final variable to consider when choosing a method for distributing certificate status information is the environment in which relying parties will be operating. If relying parties will be operating off-line, then segmentation will not be effective. Relying parties that operate off-line will not know which certificates they will be validating at the time that they obtain CRLs from the repository. As a result, if CRLs were segmented, these relying parties would need to obtain all of the segments. Over-issuing, on the other hand, will be very useful for these relying parties. If CRLs are over-issued, then relying parties will be guaranteed to always obtain relatively fresh information from the repository each time they request a CRL. Without over-issuing, some relying parties requesting certificate status information would receive CRLs that are about to expire. These CRLs would be of little or no use to the relying parties.

7. Conclusions and future work

This paper has presented a model for the traditional method of distributing certificate status information using CRLs along with models for two improved methods for using CRLs. As was shown, these improved methods can reduce peak loads on repositories, and thus improve response times, in almost all environments.

One method of certificate status distribution that this paper did not cover is the use of delta-CRLs. With delta-CRLs, one can have base CRLs whose lifetimes are relatively long, allowing for re-use, while at the same time using delta-CRLs with relatively short lifetimes in order to ensure that relying parties have up-to-date certificate status information. We are working to develop models for the use of delta-CRLs that will demonstrate how they can be used most effectively. This work will be described in a future paper.

References

[1] S. Berkovits, S. Chokhani, J. A. Furlong, J. A. Geiter, and J. C. Guild. *Public Key Infrastructure Study: Final Report.*

Produced by the MITRE Corporation for NIST, Apr. 1994.

[2] R. Housley, W. Ford, W. Polk, and D. Solo. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*. RFC 2459, Jan. 1999.

[3] S. Micali. Efficient certificate revocation. Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, Laboratory for Computer Science, Mar. 1996.

[4] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. RFC 2560, June 1999.

[5] A. S. Tanenbaum. *Computer Networks*. Prentice-Hall, Inc., second edition, 1989.