Instant Revocation

Jon A. Solworth*

University of Illinois at Chicago
solworth@rites.uic.edu

Abstract. PKI has a history of very poor support for revocation. It is both too expensive and too coarse grained, so that private keys which are compromised or otherwise become invalid remain in use long after they should have been revoked. This paper considers Instant Revocation, or revocations which take place within a second or two.
A new revocation scheme, *Certificate Push Revocation (CPR)* is described which can support instant revocation. CPR can be hundreds to thousands of times more Internet-bandwidth efficient than traditional and widely deployed schemes. It also achieves significant improvements in cryptographic overheads. Its costs are essentially independent of the number of queries, encouraging widespread use of PKI authentication.
Although explored in the context of instant revocation, CPR is even more efficient—both in relative and absolute terms—when used with coarser grain (non-instant) revocations.

Key words: Public Key Infrastructure, Revocation

1 Introduction

A *Public Key Infrastructure (PKI)* provides a means of binding a public key with the user who is the *key holder*. This binding is in the form of an identity certificate, signed by a Certificate Authority (CA).

Using the key holder's identity certificate, a third party can verify that a statement's digital signature was produced by the key holder: The public key is used to verify that the signature is for the statement and the binding identifies the user.

The party that relies on a signature, called the *relying party*, generally bears the risk if the signature is defective. This risk is reduced by ensuring that the CA is trustworthy and the identity certificate information is current. Trustworthiness comes from relying on the software and procedures for issuing certificates; perhaps the single most important component is that the CA protects its private key from disclosure or misuse. To safeguard the CA's private key, it is desirable to keep it offline, and carefully log its use. This limits how often the CA can

* This work was supported in part by the National Science Foundation under Grants No. 0627586 and 0551660. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the author and do not necessarily reflect the views of the National Science Foundation.

sign certificates and hence argues for longed lived certificates, typically on the order of a year. To prevent a relying party from using stale information, it is necessary to check that the information is still valid—that is, that the binding has not been *revoked*. (In general, an identity certificate can be revoked because the private key was exposed or because the binding is no longer valid.)

Unfortunately, the *revocation problem* has long been a difficultly in PKI, both in obtaining timeliness and efficiency of revocation mechanisms. The cost of revocations is the dominant cost for a PKI [23]. Rivest, for example, suggested short lived certificates [20], but this requires frequent signings and hence exposes the CA's private key to attack, especially if it can no longer be kept off line. Gutmann called revocation a Grand Challenge problem in PKI [6] and discusses several issues with it [5]. At a recent security architectures workshop[1], Sandhu stated PKI's critical need is for instant revocation, which he described as revocations which take place within a couple of seconds. Instant revocation is interesting since it states the problem as a requirement (recency of revocation information) rather than a mechanism for achieving it (eg. on-line queries).

We therefore consider the problem of *instant revocation*, which, following Sandhu, is defined as a revocation mechanism which can invalidate an identity certificate in no more than a second. In addition to invalidation, the revocation must be propagated to its destination to be used by the relying party. To achieve a total revocation delay of two seconds, the propagation delay must be no more than a second[2].

We introduce a new PKI revocation scheme, which we call *Certificate Push Revocation (CPR)*. Our focus here is on the design of the *Validity Authority (VA)* which revokes certificates, the *cache* which hold copies of revocations, and the relying party. (Normally, the VA would be would be part of the CA, but we separate them here to emphasize VA function.) While CPR was developed for instant revocation, it is even more efficient (and some requirements can be relaxed) when used for longer revocation intervals. CPR relies on efficiently pushing the updates towards the relying party, rather than a combination of pull and push as in other schemes.

The instant revocation problem is complicated because authentication takes place in a distributed environment amongst different organizations which may have limited trust in each other. Hence, the authentication should be robust, that is support deniability resistance[3], so that the relying party can prove after the fact that the certificate was authenticated. This is the strongest guarantee

[1] ACM 1st Computer Security Architectures Workshop, Panel on Distributed Authentication, Panelists Angelos Keromytis, Ravi Sandhu, and Sara "Scout" Sinclair.

[2] As network latency requirements seems inherent in *any* instant revocation scheme, and are not PKI specific, we do not consider them further here. Note that even on-line checks experience a round trip delay.

[3] We use *deniability resistance* to mean that with a few assumptions (the private key is under the sole control of the key holder and the crypto is not broken) then the key holder must have signed the statement. Of these assumptions, the "sole control" is the most likely to be violated. Nonetheless, deniability resistance has the minimum set of assumptions and therefore is the best basis for dispute resolution.

to the relying party that a VA can make, since the VA cannot later deny the status that it provided.

There are further requirements, since the above timing requirement can be trivially met with existing techniques—for example, a digital signature certifying the status can be computed in well under a second. The scheme therefore must also be efficient in its use of resources—in particular, it must be economical in its use of Internet bandwidth.

Hence, investigators traditionally measure the efficiency of certificate revocation schemes in term of their Internet bandwidth. To analyze and optimize revocation schemes accurately, it is important that the costs be accurately captured in the model. However, the network metric traditionally used is bits/day, although bandwidth is typically priced in terms of *peak* bandwidth usage, as the bandwidth provider must build out network hardware resources to address this peak demand. Hence, if there is significant difference in peak demand vs. average demand, average bit rates do not accurately reflect costs. Therefore we analyze, and our goal is to minimize, peak Internet bit rates. In particular, local network costs are inexpensive and hence ignored (although we show that local bandwidth requirements are modest).

Ideally, a revocation scheme should achieve the following goals:

1. Support instant revocation,
2. Efficiently use peak network bandwidth for the VA,
3. Efficiently use network bandwidth for the relying party,
4. Efficiently use computational resources at the VA,
5. Efficiently use computational resources at the relying party,
6. VA costs should increase minimally as revocation checks increase, and
7. VA vulnerability to attack should be minimized.

Item (1) is a requirement for PKI in high value operations. Items (2)-(5) are necessary to make the scheme economically viable[4]; Item (6) encourages use, thus increasing the value of the PKI facility; item (7) is desirable to protect the VA (and its high value keys) from attack.

CPR achieves all 7 of these goals. In contrast, existing techniques clearly fail to efficiently achieve the two-second goal. Existing techniques either have high cost per use (e.g., a digital signature) or costs which depend on the time granularity (e.g., hash chains) which make them more expensive to use for instant revocation. Techniques which attempt to aggregate unrelated information (such as revocation lists) increase transmission size and thus network costs.

CPR is also suitable for coarser grained revocation. We believe that, when applicable, it sets records as best in class for (1), (2), (3), (4), and (6); in some cases the improvements are by orders of magnitude.

The rest of the paper is organized as follows: Section 2 gives the background, Section 3 describes related work, and Section 4 characterizes VA attributes.

[4] Peak rate is used for the VA and average rate for the relying party. The VA's sole mission is to support PKI (and hence its peak network bandwidth is solely for the support of revocations), while the relying party caries on a variety of tasks and hence it is somewhat less sensitive to peak rates.

Section 5 presents CPR. Section 6 shows the performance of CPR against OCSP and CRS and describes how to combine CPR with other schemes to support low-bandwidth, intermittently connected computers. Section 7 describes security considerations and then we conclude.

2 Background

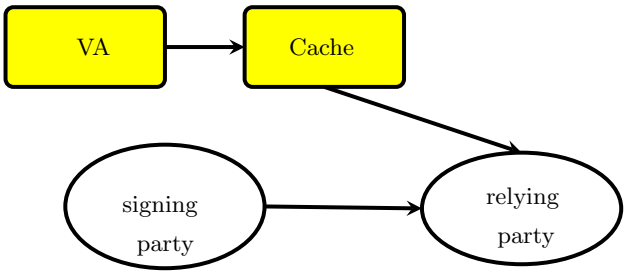

Fig. 1. Parties to a revocation scheme

Revocation schemes have followed a common architecture, using a VA and, optionally, a set of caches. They vary in the algorithms and protocols used to provide up-to-date information. Revocation schemes are centered around the following 4 components:

VA holds the secrets (such as private keys and/or hash chain seeds) necessary for revocation and thus produces deniability resistant validity information.

Caches hold replicas of the information produced by the VA and answer queries as to the validity (revocation status) of certificates. Caches do not hold secrets and need not be trusted. Hence, for schemes in which secrets are necessary to answer queries, the VA answers queries directly and the cache is eliminated.

Signing party is the user whose private key is used to produce a digital signature and who is the subject of authentication.

Relying party is the user who receives a digitally signature, determines who it corresponds to, and validates it.

In general, each of the above entities is administered by a different organization, and hence the network traffic between them is typically routed over the Internet. These parties and the communications between them are shown in Figure 1.

Although both VA and cache costs are borne by the same entity, they are managed differently because their security needs are different; the VA must be trusted for integrity and availability while caches collectively need only supply sufficient availability. This enables the caches to be outsourced. The cost of a revocation scheme is measured by the cost of Internet traffic from VA to cache

(providing the basis of authentication) plus the *queries* to the caches (or VA) as to the validity of specific certificates.

The queries can be made by the relying party or the signing party. With traditional revocations periods, such as a day, cache costs are reduced by sending information to the signing party. Given a single query, the signing party can send the validity information to multiple relying parties until the revocation period expires. However, signer caches has negligible advantage with instant revocation and thus the signing party saves network bandwidth when the cache transmits revocation information to the relying party. Moreover, this optimization does not increase cache costs. It is also more secure, as it prevents the signing party from sending dated information when she knows that her entry is about to be, or already has been, revoked (for example, just after she is fired).

3 Related work

Initial designs for PKI centered on Certificate Revocation Lists (CRLs) which are lists of bad certificates digitally signed by a VA. CRLs are patterned after off-line bad credit card number lists which were published by credit card issuers and used by retailers; a credit card not in the list was assumed to be good. To reduce the CRL publication and use costs, two CRL variants are used: *segmented CRLs* (partitioned by certificate serial number, alternatively see [12]) or *delta CRLs* (containing the changes since the previously published CRL). As revocations are typically assumed in the literature to be about 10% [15], CRLs constitute a non-trivial share of the VA plus CA databases. CRLs are inefficient due to their size and frequent publication (needed to ensure timeliness) [15, 20].

CRLs are patterned after an offline technique, so its natural that there be an on-line version. In order to provide deniability resistance, its necessary for the relying party to have positive proof that a certificate has not been revoked. To provide this property, an *On-Line Certificate Status Protocol (OCSP)* signs each revocation request [18]. OCSP provides high timeliness, but the signature increases query network bandwidth and CPU costs. The OCSP scheme we consider here does not use caches, an alternative to OCSP-based scheme uses multiple private keys to enable some secrets to be moved to the caches [10].

Micali proposed an authentication scheme called *Certificate Revocation System (CRS)* [14–16] which combines Lamport's hash chains with certificates. Lamport's scheme uses a cryptographic hash $H(x)$, applied n times to a *hash seed* s, $H^n(s)$ to support n authentications [11]. The hash function H and the top of the hash chain $H^n(s)$ are public information while s is secret. Authentication i is proved by providing $a_i = H^{n-i}(s)$ which can be verified by showing that $H^n(s) = H^i(a_i)$. It is computationally infeasible to determine s from $H(s)$. CRS uses a hash value for each revocation period. Hash chains have two main advantages over signing; hash values are an order of magnitude smaller, and about 10,000 times more efficient to compute, than a signature.

Hashing is also useful to summarize an arbitrary amount of data, such as some part of the VA database. If only part of the data is used or changed, then

an efficient alternative to a simple hash is a tree of hash values—called a Merkle Tree [13]. As with a single hash, the root of the Merkle Tree summarizes the whole data. The tree is constructed recursively from the children; given two children containing values d_0 and d_1, the parent contains $H(d_0|d_1)$ where '|' is concatenation. Given S values, stored at the leaves, $\log S$ hash values are needed to compute the root of the tree and thus verify the value.

Certificate Revocation Trees (CRTs) uses a tree whose leaves are ranges of values [9]. A certificate is valid if its serial number falls within one of the ranges of the CRT and is otherwise revoked. Every new time period, a signed list of revoked certificates is sent to the cache, which updates (and rebalances) its tree. The drawback of this scheme is that certificate validations requires logarithmic (in the number of certificates) hash values—using a Merkle Tree—plus a signature. Hence, it reduces VA-to-cache costs while increasing query costs.

To reduce VA to cache costs versus CRS, Goyal proposed using hash chains (based on CRS) to authenticate groups of certificates [4]. The hash chain is attached to a validity certificate which has a validity bit for each certificate in the range. If one of the (previously valid) certificates became invalid, a new certificate would be issued with a new hash chain. This scheme increases revocation-to-cache costs while decreasing query costs.

We do not consider certificate chains in our performance comparisons. All of the schemes (including our own) handle certificate chain processing, but such considerations drive up the authentication cost of other schemes while having no effect on our own. The cost of chains can be reduced for other schemes using synthetic certificates [21].

We consider only revocation here, and not the reason for revocation (i.e., status), although CPR could be extended to do so. For a fuller discussion of PKI revocation and status issues see [8, 7].

An alternative approach for revocation is to revoke public keys rather than the certificates which contain them. For example, security mediated PKI does this with a mediator which the relying party queries [24, 25]

4 Characterization of VAs

We next describe the parameters which have the greatest impact on the performance of revocation schemes. Each certificate is assumed to have a serial number; serial numbers are consecutively issued by the CA[5]. Two of the most significant parameters of the VA are the *number of certificates*, denoted N and the *annual revocation rate*, denoted R. From these, the size of the active range of serial number, $N_r = N/R$ can be derived. Q is the number of authentication (queries) per user per day. We are interested in peak performance, the peaks are characterized relative to average authorizations and revocations:

- P_a is the ratio of authentications in the busiest second to that of the average second, and

[5] It is possible to extend this scheme to allow, for example, multiple series of sequence numbers.

– P_r is the ratio of revocations in the busiest second to that of the average second.

We consider two VA sizes, the small VA with $N = 10,000,000$ and the large VA with $N = 100,000,000$. These values are summarized in Table 1. To be conservative, we have chosen P_a ranges to be relatively modest (since that affects only traditional schemes) while chosen P_r ranges to be more aggressive (since that affects only CPR).

Parameter	Meaning	Small VA	Large VA
N	Certificates	10,000,000	100,000,000
R	Annual revocation fraction	.1	.1
N_r	Size of serial number range	N/R	N/R
Q	Number of authentications per certificate per day	1–64	1–64
P_a	Peak–ratio of authentications	1–10	1–10
P_r	Peak–ratio of revocations	1–100	1–100

Table 1. Certificate Authority statistics

The schemes described here use cryptographic hashing, digital signatures, and Merkle (Hash) Trees. For the purposes of presenting performance, we use 2048-bit RSA signatures and SHA-1 hashing. The cost of cryptographic operations are shown in Table 2 (for further information see `http://www.cryptopp.com/benchmarks-amd64.html`). The table's numbers are for a current but inexpensive processor.

5 Certificate Push Revocation (CPR)

CPR provides the relying party with sufficient information to validate *any* certificate. Figure 2 shows the parties to CPR. Rather than the cache being owned by the VA as in other schemes, in CPR the cache is owned by, and located at, the relying party[6]. This does 3 things

– The VA does not pay *any* cache costs, resulting in substantial savings. These savings include elimination of cache hosting and, most importantly, queries;
– The cache cost (including queries) is essentially zero, since it is co-located with the relying party which performs revocation checking; and

[6] We distinguish our scheme from others in which the cache must be trusted.

Operation	Time	Cycles
SHA-1 (Hashing)	0.546 μs	999 + 10.6 cycles/byte
RSA 2048 Signature	5,950.000 μs	10,890,000
RSA 2048 Verification	150.000 μs	280,000

Table 2. Crypto++ benchmarks on AMD Opteron 2.4 GHz processor

– The relying party, which is taking the risk, can make tradeoffs which minimize costs.

Because the cache has no secrets (integrity depends only on VA operations) moving it to the relying party does not reduce security. The cost to the relying party is very low.

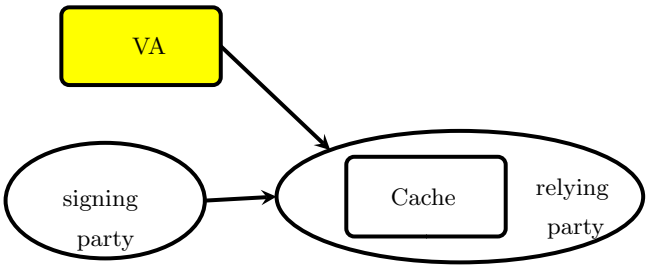

Fig. 2. Parties to CPR

Locating the cache at the relying party enables it to perform many authentications per second for not much more than the price of a single authentication. But more importantly, it is the relying party which needs the assurance that the information is valid (and thus will invest appropriately to protect it).

Finally, the relying party can chose when to take or mitigate risks depending on its business model. For example, it might choose a lower cost (and availability) system at the price of increased fraud. Different businesses are subject to different risks, for example a web-based retailer may choose after-the-fact verification, while an investment brokerage firm may refuse to perform a transaction without up-to-date verification. Indeed, such a tradeoff is inherent [2, 3].

Because the cache is maintained at the client site, and CPR has many caches, we need to consider cache failure. Of course, other schemes need to recover from cache failure, but papers have traditionally ignored this cost because it is small in traditional schemes. Hence, we'll measure it carefully for CPR and assume it's cost is zero for other schemes.

Moreover, to ensure that we count all costs and provide maximum security for the VA, access to the VA is limited to the minimum necessary: that which performs revocation and addition of new certificate serial numbers. To enable recovery under this scenario, the VA continually broadcasts recovery information. If a cache fails, it simply listens to this broadcast channel until it is up to date.

Instant revocation relies on posting revocations once a second. In CPR, all revocations for the past second are posted to the cache. Revocations are the slowest changing part of the system, and so having updates depend on revocations is likely to be more efficient than other techniques (this is how CRT achieves low VA-to-cache costs). For example, assuming an annual revocation rate of 10%, the small VA will have 0.032 revocations per second while the large VA will have 0.317 revocations per second. This enables the *certificate validity vector*—a

vector which contains a bit for each certificate indicating whether that certificate has been revoked—to be kept up-to-date at the relying party. Furthermore, although each cache holds the entire revocation database, it is a relatively small data structure, for example, for the Large VA it is less than 20 megabytes.

Our protocol is exceeding simple. Every second it sends out:

1. The summary (i.e. the root of the Merkle tree) of certificate validity vector,
2. The revocations in the last second, and
3. Part of the certificate validity vector for some range of certificates.

Item (1) is used to secure revocations. Item (2) is used to update certificate validity vector. Item (3) is used for recovery after a cache failure. (Recovery after a VA failure is provided by simply sending out the "updates" once the VA comes up).

Revoker We first describe the revoker, which contains items (1) and (2). The revoker data structure is shown in Table 3. We have tried to be conservative in its design (no more than the number of bits specified are required), although if a few thousand more bits are needed it is of little consequence as only one revoker is issued per second.

Name	Bits	Purpose
time	64	Number of seconds since epoch
min	32	Minimum certificate serial number
max	32	Maximum certificate serial number
hash	160	Merkle hash root of certificate validity vector
s_{rv}	32	size of the rv
rv	$32s_{rv}$	Revocation vector
signature	2048	RSA 2048 bit signature
	$2368 + 32s_{rv}$	total bits

Table 3. Revoker

It contains the time in seconds since some epoch as a 64-bit integer; to ensure that the relying party has the latest update, it needs a reasonably accurate clock[7].

The min and max are the minimum and maximum certificate serial numbers. The max increases when new certificates are issued. The min increases when the lowest certificate serial number for the VA is either revoked or expires.

The revoker "points" to a *certificate validity vector,* a bit vector of length $\max - \min + 1$ with bit i being 1 if certificate $\min + i$ is not revoked and 0 otherwise. As certificates are issued in order, and about 10% are revoked during the year, the bit vector is of length N_r.

[7] Time synchronization protocols such as Network Time Protocol (NTP) easily provide synchronization to a small fraction of a second [17]. Other time sources include GPS and radio receivers; obtaining an accurate time source is inexpensive. Moreover, the relying party can always use a slightly older revoker (i.e., the last one received).

Name	Bits	Purpose
time	64	Number of seconds since epoch
startVec	32	starting certificate serial number of the vector
sizeVec	32	size of the validity vector
valVec	$N_r/T_{recovery}$	$1/T_{recovery}$ fraction of the validity vector
signature	2048	RSA 2048-bit signature

Table 4. Certificate Validity Segment

Finally, to enable the cache to update its certificate validity vector, the revocations that occurred in the last H seconds are sent (H can be set to 1 or larger, and is a parameter related to fault tolerance). The cost of additional revocations (due to $P_a > 1$ or $H > 1$) is quite small because the average number of revocations is very small and because a 32-bit serial number is significantly smaller than a 2048-bit signature (both are fields of the revoker).

The algorithm at the cache is as follows:

- Each second the revoker is received and used to update the local copy of the certificate validity vector (by turning off bits corresponding to the revoked certificates and adjusting the bit vector to reflect the new valid range);
- The root of the Merkle tree is computed over the bit vector;
- The computed hash is valid if it equals the hash in the revoker and the signature in the revoker is valid; and
- The bit vector can then be used to check an arbitrary number of revocations in the next second.

Certificate Validity Segment The certificate validity segment, used to construct the certificate validity vector at the cache, is described here.

Every second a certificate validity segment containing $1/T_{recovery}$ of the certificate validity vector is transmitted; in $T_{recovery}$ seconds the entire certificate validity vector is transmitted. Hence, smaller $T_{recovery}$ results in faster recovery after a failure, but at a higher bit rate.

The fields of the certificate validity segment are shown in Table 4. The time field is as in the revoker. The current portion of the certificate validity vector is described by:

startVec the starting certificate serial number of the vector,
sizeVec the size of the vector, and
valVec the bits of this segment of the certificate validity vector.

Its possible to save some bits (the signature and some time) by combining the certificate validity segment with the revoker. However, the certificate validity segment is used only for recovery while the revoker is used to validate signatures, so there is an advantage of making the revoker small as it reduces average relying party bandwidth requirements.

		Auth/sec		bits/sec		CPU cores	
Q	P_a	Small VA	Big VA	Small VA	Big VA	Small VA	Big VA
1	1	115.74	1,157.41	244,444.44	2,444,444.44	0.69	6.89
4	1	462.96	4,629.63	977,777.78	9,777,777.78	2.75	27.55
16	1	1,851.85	18,518.52	3,911,111.11	39,111,111.11	11.02	110.19
64	1	7,407.41	74,074.07	15,644,444.44	156,444,444.44	44.07	440.74
1	5	578.70	5,787.04	1,222,222.22	12,222,222.22	3.44	34.43
4	5	2,314.81	23,148.15	4,888,888.89	48,888,888.89	13.77	137.73
16	5	9,259.26	92,592.59	19,555,555.56	195,555,555.56	55.09	550.93
64	5	37,037.04	370,370.37	78,222,222.22	782,222,222.22	220.37	2,203.70
1	10	1,157.41	11,574.07	2,444,444.44	24,444,444.44	6.89	68.87
4	10	4,629.63	46,296.30	9,777,777.78	97,777,777.78	27.55	275.46
16	10	18,518.52	185,185.19	39,111,111.11	391,111,111.11	110.19	1,101.85
64	10	74,074.07	740,740.74	156,444,444.44	1,564,444,444.44	440.74	4,407.41

Table 5. OCSP network bandwidth and CPU costs

6 Performance

The performance of revocation schemes is measured here by two metrics: The primary performance metric is Internet network bandwidth and secondarily, the cost of performing cryptographic operations.

In particular, we consider the effect of peak rate authentications and revocations[8]. This will almost certainly be significantly higher than the average rates, and the VA infrastructure must be sized to accommodate them. For example, financial markets are typically opened only a limited number of hours per day and in addition tend to see the greatest volume at opening and closing; moreover, trading volume varies widely on different days. In retail, peak sales are often related to Christmas or New Years shopping, depending on country. It is the peak load for which these systems must be sized, for example Amazon.com sizes its systems to work with extreme reliability even for its peak load during the 2 weeks prior to Christmas. It would seem that minimum peak to average ratios would be at least 5 although much higher peaks would be necessary in many scenarios. Sizing for peak loads significantly drives up costs; ignoring peak factors significantly distorts costs.

CPR outperforms traditional schemes under peak rates. But CPR's advantages do not depend on peak rates—it significantly outperforms other schemes under average rate measurements as well ($P_a = P_r = 1$). Our technique's advantages increase as the peak authentication ratio increases.

6.1 OCSP

The cost of OCSP depends on Q (average number of authentications per user per day) and P_a. In Table 5, values are given for P_a equals 1, 5, and 10 and values

[8] Although CPR performance is independent of query rate, other schemes such as OCSP and CSR are not.

		Auth/sec		bits/sec		CPU cores	
Q	P_a	Small VA	Big VA	Small VA	Big VA	Small VA	Big VA
1	1	115.74	1,157.41	22,222.22	222,222.22	5.46	54.6
4	1	462.96	4,629.63	88,888.89	888,888.89	5.46	54.6
16	1	1,851.85	18,518.52	355,555.56	3,555,555.56	5.46	54.6
64	1	7,407.41	74,074.07	1,422,222.22	14,222,222.22	5.46	54.6
1	5	578.70	5,787.04	111,111.11	1,111,111.11	5.46	54.6
4	5	2,314.81	23,148.15	444,444.44	4,444,444.44	5.46	54.6
16	5	9,259.26	92,592.59	1,777,777.78	17,777,777.78	5.46	54.6
64	5	37,037.04	370,370.37	7,111,111.11	71,111,111.11	5.46	54.6
1	10	1,157.41	11,574.07	222,222.22	2,222,222.22	5.46	54.6
4	10	4,629.63	46,296.30	888,888.89	8,888,888.89	5.46	54.6
16	10	18,518.52	185,185.19	3,555,555.56	35,555,555.56	5.46	54.6
64	10	74,074.07	740,740.74	14,222,222.22	142,222,222.22	5.46	54.6

Table 6. CRS network bandwidth and CPU costs

for Q of 1 to 64. When $Q = 1$ and $P_a = 1$, the result is 116 authentications per second for the small VA and 1157 per second for the large VA. As each authentication requires 2112-bits, (a 32-bit certificate serial number, a 32-bit time, and a signature). Minimum bit rates for this scheme are about 244,000 bits/second for the small VA and 2,444,000 bits/second for the large VA. They increase linearly with the number of queries per day. For the large VA, they exceed over a billion bits per second in a system which has $Q = 64$ and $P_a = 10$.

The cost of computing signatures is also significant. While very low authentication and peak rates can be performed with just a few processing cores[9], high rates result in the need for hundreds or thousands of cores (or custom cryptographic hardware). Furthermore, extra expense is incurred for these cores to suitably protected the highly sensitive private key of the VA.

6.2 CRS

CRS achieves a significant savings over OCSP because hash values are so much smaller than signatures (160 bits vs. 2048 bits) and because the time is implicit in CRS (by the hash value's location in the hash chain). CRS's scheme gains a factor of 11 in network bandwidth. The corresponding CRS performance in shown in Table 6. Nevertheless, CRS requires significant network bandwidth of up to 142,000,000 bits/second.

There are also significant savings in CPU cores for CRS vs. OCSP, although the time to compute the hash chains needed is non-trivial; 5.46 processor cores for the small VA and 54.6 cores for the large VA. This under counts the number of cores, since it does not count run-time computation needed to recompute these hash values.

[9] Most desktop, notebook, or server processor chips today have 2-4 cores, or independent CPUs, and thus able to execute 2-4 independent programs simultaneously.

6.3 CPR

In CPR, the VA transmits a revoker and certificate validity segment every second. The Internet Protocol's multicast, enables the same packet to be delivered to an arbitrary number of Internet destinations. Two multicast groups are used, one for revoker and another for certificate validity segment traffic. A relying party can join and leave these multicast groups independently. When a relying party is running, it is always part of the revoker multicast group; it is only part of the certificate validity segment multicast group when recovering from a failure. The latter is needed for making CPR reliable.

We consider first the revoker. Its only variable component is the size of the rv containing revoked serial numbers (see Table 3). The average number of revocations/second is .032 for the small VA and .317 for the large VA. The peak number of revocations sent is $P_r \cdot H$ times the average number of revocations. We note that bandwidth usage is not too heavily related to this value for the ranges of P_r and H considered here. Moreover, CPR is totally independent of the number of queries.

The results for a variety of values of H and P_r are shown in Table 7. For example, when $H = 10$ and $P = 100$ (a very conservative choice), the average number of bits during peak load is 3,382 for a small VA and 12,512 for a large VA. These are the only updates needed during normal operation.

		avg. bits/second				avg. bits/second	
H	P_r	Small VA	Large VA	H	P_r	Small VA	Large VA
1	1	2,369.01	2,378.14	10	1	2,378.14	2,469.44
1	5	2,373.07	2,418.72	10	5	2,418.72	2,875.20
1	10	2,378.14	2,469.44	10	10	2,469.44	3,382.40
1	50	2,418.72	2,875.20	10	50	2,875.20	7,440.00
1	100	2,469.44	3,382.40	10	100	3,382.40	12,512.00
5	1	2,373.07	2,418.72	20	1	2,388.29	2,570.88
5	5	2,393.36	2,621.60	20	5	2,469.44	3,382.40
5	10	2,418.72	2,875.20	20	10	2,570.88	4,396.80
5	50	2,621.60	4,904.00	20	50	3,382.40	12,512.00
5	100	2,875.20	7,440.00	20	100	4,396.80	22,656.00

Table 7. Revoker bits

The cost of cryptography at the VA is extremely modest (it is even less at the relying party).

- Two RSA signatures—one signature for the revoker and one for the certificate validity segment (11.90 milliseconds),
- Update of the Merkle tree for a Small/Large VA (39/347 microseconds), using the maximum $P_r = 100$.

That is, the cryptographic cost is a small fraction of a single core.

The cost to the relying party is significantly smaller than to the VA, as it verifies a signature (150 μs) rather than signs (5,950 μs). Hence, the CPU cost/second at the relying party is less than 1 millisecond (339/647 μs).

We next consider the certificate validity segment. The peak network bandwidth on this multicast channel is dependent on $T_{recovery}$. We consider recovery periods from one to five minutes. Our goal is to measure the bandwidth from two points of view, the cache and the VA. The bandwidth requirements depend on $T_{recovery}$ and are a bit more than a T1 line[10] (each 1.536 MBits/s) at 60 second recovery and only a few hundred thousand bits/second at a 5 minute recovery.

	bits/second	
$T_{recovery}$	Small VA	Big VA
60	187,361.19	1,854,127.85
120	94,768.59	928,201.93
300	39,213.04	372,646.37

Table 8. Instant revocation recovery information network costs

CPR is designed for instant revocation. However, it is the only scheme analyzed whose network cost *decreases* with longer revocation intervals (mostly due to fewer signatures). All of the other scheme's network cost depend only on the number of queries, and are therefore independent of the revocation interval. Hence, CPR's absolute and relative performance advantages *improve* with longer revocation intervals, making it advantageous to use at all revocation intervals.

6.4 Blended schemes

CPR is clearly very inexpensive for the VA, so we now consider its effect on the relying party. For a relying party with a high speed Internet connection, CPR's bandwidth use is modest, especially given ISP practices[11]. However, instant revocation may be impractical for computers connected intermittently or at low-speed. These schemes can also be used to avoid firewall limitations in large companies, by setting up a directory outside the firewall.

In general, cache schemes are either trusted or untrusted. A *trusted cache* means that the relying party cannot verify that the information provided by the cache is the same as that provided by the VA. For example, a trusted cache might be maintained by one's employer. An *untrusted cache* means that the cache must provide a deniability resistant proof that the cache information is the same as that provided by the VA.

[10] The standard unit of commercial Internet bandwidth.

[11] Cable/DSL ISP's typically have a *gap* between actual vs. advertised bandwidth, due to insufficient Internet bandwidth. However, CPR is very low cost in terms of Internet bandwidth due to multicast, and hence its bandwidth may be almost free (decrease the gap) rather than slowing down other connections.

There are three *no-cost cache extensions*—i.e., they do not increase VA bandwidth or computational costs—which enable intermittent or low-speed computers to be connected:

trusted cache Each query to a trusted cache can be answered with "revoked" or "non-revoked". This combines Gutmann's on-line query with deniability resistance from the trusted cache. An example of a trusted cache is the webDAV cache [1].

untrusted cache (ISP) A cache provided by the relying party's ISP does not consume additional Internet bandwidth (only bandwidth between the relying party and ISP)[12].

untrusted cache (signing party) Cache information can be provided by the signing party, assuming the signing party is not low bandwidth.

Untrusted caches for CPR can be implemented most easily with CRT-style query responses as no change is needed in VA-to-cache traffic. The query response includes the latest revoker, the 512-bit part of the certificate validity vector containing the certificate's validity bit and a logarithmic number of hash values (3072-bits plus the revoker). The relying party can then verify that the query response was provided by the VA[13].

	Costs				increased revoc period	
Technique	Crypto Cost	Network Cost	Deniability Resistance	Uses Cache	Reduced Networking	Reduced Crypto
OCSP	High	High	yes	no	no	no
CRS	Medium	Medium	yes	no	no	yes
CRT	Low	High	yes	yes	yes	yes
CPR	Low	Low	yes	yes	yes	yes
trusted dir.	Low	Low	no	yes	no	no

Table 9. Comparison of different schemes

If the above techniques are not sufficient for revocations, CPR can be combined with other schemes to reduce overall system costs. It is profitable to do so, because instant revocation minimizes query costs which dominate overall VA costs in traditional schemes; hence satisfying even a fraction of the queries by instant revocation will save Internet bandwidth.

Therefore, the VA can also do OCSP or CRS style revocation. This would increase VA query costs, which need to go out over the Internet. To a first order approximation, if the percentage of requests which can be answered by CPR, the

[12] This is advantageous to the ISP as it reduces Internet query costs. Fielding of such a service is analogous to providing DNS servers, which all ISP's do.

[13] Further optimization can be achieved by splitting the revoker into two components, one with the Merkle tree root and the other with the revoked certificate serial numbers.

ISP, or a trusted cache is p, then p is the savings in bandwidth over a non-blended approach. The differences in the various schemes are summarized in Table 9.

We believe that with the no-cost cache extensions above, the vast majority of revocations can be answered without the VA incurring Internet query costs.

7 Security considerations

Since the VA only countersigns certificates from the CA, its security implications are limited. A compromised VA can:

- Revoke certificates which are valid or
- Fail to revoke certificates which are invalid.

Such security violations can be detected by auditing at any relying party.

Any VA is going to depend on a correct stream of new certificate and revocation information. Similarly, any VA is subject to denial-of-service attacks on the VA or VA-to-cache path.

Other than that, the VA is the only trusted entity for the integrity of the system. The VA signs both the changes and the hash of the certificate validity vector. Hence, assuming that its private key is kept private, the signature and hashing scheme are not broken, and that its rather simple calculations are performed correctly, the integrity of the system is ensured.

Further, we assume that revocations are not confidential. Knowledge of certificate validity is public, and hence its disclosure does not violate security.

8 Conclusions and Future Work

We have presented, CPR, a revocation scheme which is capable of instant revocation and is efficient. This is the first practical PKI revocation scheme which meets this goal.

It is efficient in network traffic as it reduces external network traffic to little more than the rate of revocations, the slowest changing part of the revocation system. It shows improvements in Internet bandwidth of 100s to 1000s times over well known and widely used techniques. It is so efficient that it is trivial to add in redundancy so that the scheme is robust.

Additionally, it requires very little computational resources and can be 10s to 1000s of times more efficient than other schemes. For the case of low bandwidth, intermittently connected devices CPR can be blended with other techniques. Several of these blends do not increase VA costs over a pure instant revocation scheme.

CPR eliminates the VA's need to provide caches—and the queries made by signing or relying parties against them—by co-locating the cache with the relying party. It does this at very low cost to the relying party. Finally, it allows the relying party to make tradeoffs of availability vs. risk of fraud by using somewhat slightly older revocation information.

Although our goal was to meet requirements of instant revocation, the scheme is perfectly suitable for revocations which need not be as timely, as it is even more efficient for longer revocation intervals than it is for instant revocation.

We have started to build a CPR prototype and associated infrastructure using a simplified but powerful certificate system called sayAnyting [22]. These mechanisms are to be used, among other things, as part of an enterprise-wide authentication system [19].

References

1. David W. Chadwick and Sean Anthony. Using webDAV for improved certificate revocation and publication. In Javier Lopez, Pierangela Samarati, and Josep L. Ferrer, editors, *4th European PKI Workshop: Theory and Practice, EuroPKI*, volume 4582 of *Lecture Notes in Computer Science*, pages 265–279. Springer, 2007.
2. Armando Fox and Eric A. Brewer. Harvest, yield and scalable tolerant systems. In *Workshop on Hot Topics in Operating Systems*, pages 174–178, 1999.
3. Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News*, 33(2):51–59, 2002.
4. Vipul Goyal. Certificate revocation using fine grained certificate space patitioning. In *Financial Cryptography and Data Security Conference*, 2007.
5. Peter Gutmann. PKI: It's not dead, just resting. *IEEE Computer*, 35(8):41–49, 2002.
6. Peter Gutmann. Drawing lessons. In *3rd PKI workshop*, 2004. Invited talk.
7. John Iliadis, Stefanos Gritzalis, Diomidis Spinellis, Danny De Cock, Bart Preneel, and Dimitris Gritzalis. Towards a framework for evaluating certificate status information mechanisms. *Computer Communications*, 26(16):1839–1850, 2003.
8. John Iliadis, Diomidis Spinellis, Dimitris Gritzalis, Bart Preneel, and Sokratis Katsikas. Evaluating certificate status information mechanisms. In *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*, pages 1–8, New York, NY, USA, 2000. ACM.
9. Paul C. Kocher. On certificate revocation and validation. In *FC '98: Proceedings of the Second International Conference on Financial Cryptography*, pages 172–177, London, UK, 1998. Springer-Verlag.
10. Satoshi Koga and Kouichi Sakurai. Proposal and analysis of a distributed online certificate status protocol with low communication cost. *IEICE Transactions*, 88-A(1):247–254, 2005.
11. Leslie Lamport. Password authentification with insecure communication. *Commun. ACM*, 24(11):770–772, 1981.
12. Javier Lopez, Antonio Mana, José A. Montenegro, and Juan J. Ortega. PKI design based on the use of on-line certification authorities. *Int. J. Inf. Sec.*, 2(2):91–102, 2004.
13. R. Merkle. A digital signature based on a conventional encryption function. In *Crypto*, pages 369–378, 1987.
14. Silvio Micali. Efficient certificate revocation. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.
15. Silvio Micali. Efficient certificate revocation. In *Proceedings 1197 RSA Data Security Conference*, 1997.
16. Silvio Micali. NOVOMODO: Scalable certificate validation and simplified PKI management. In *1st PKI Workshop*, 2002.

17. David L. Mills. Network Time Protocol (version 3) specification, implementation and analysis. Internet Request for Comment RFC 1305, Internet Engineering Task Force, March 1992.
18. Online certificate status protocol, version 2. Working document of the Internet Engineering Task Force (IETF).
19. Manigandan Radhakrishnan and Jon A. Solworth. Netauth: Supporting user-based network services. In *Usenix Security*, 2008.
20. Ronald L. Rivest. Can we eliminate certificate revocations lists? In *Financial Cryptography*, pages 178–183, 1998.
21. Selwyn Russell, Ed Dawson, Eiji Okamoto, and Javier Lopez. Virtual certificates and synthetic certificates: new paradigms for improving public key validation. *Computer Communications*, 26(16):1826–1838, 2003.
22. Jon A. Solworth. What can you say? and what does it mean? In *Workshop on Trusted Collaboration*. IEEE, 2006.
23. Stuart Stubblebine. Recent-secure authentication: Enforcing revocation in distributed systems. In *Proceedings 1995 IEEE Symposium on Research in Security and Privacy*, pages 224–234, May 1995.
24. Gabriel Vanrenen, Sean W. Smith, and John Marchesini. Distributing security-mediated PKI. *Int. J. Inf. Sec*, 5(1):3–17, 2006.
25. Jong-Phil Yang, Kouichi Sakurai, and Kyung Hyune Rhee. Distributing security-mediated PKI revisited. In *EuroPKI*, pages 31–44, 2006.