Revocation: Options And Challenges

Michael Myers

VeriSign, Inc.
mmyers@verisign.com **

Abstract. Public keys can be trusted if they are digitally signed by a trusted third party. This trust is most commonly conveyed by use of a digital certificate. However, having once established trust in a public key, means must exist to terminate that trust should circumstances dictate. The most common means to do so is through revocation of the corresponding digital certificate. This paper identifies and discusses options that may be considered by those undertaking to address the revocation of digital certificates.

1 Introduction

Who do you trust? And why? And will you be protected if something goes wrong? These are tough questions facing the builders of PKI today as we move beyond the initial broad releases of products and services and into more value-added realms. As we do so, one fact is glaringly obvious: while the current crop of commercial products and services today have done a fair job of turning on the PKI machine, there is yet no means in place to shut it off; there is no revocation.

This paper adopts the somewhat constrained view that a successful infrastructure must provide well-understood and broadly deployed capabilities. A freeware toolkit that parses an X.509 CRL does not form an infrastructure. Rather, infrastructure is formed when capabilities are uniformly deployed in large scale such that services or derivative products can be developed with an assumption of the existence of these capabilities.

In view of this claim, we are today faced with applications that implement but a portion of standards-compliant logic necessary for a globally scaleable public key infrastructure. Correspondingly, PKI service providers, self-certifying enterprises and various industry consortia are foreclosed from a complete solution. The time is rapidly approaching when revocation capabilities will become essential if the marketplace often predicated on PKI is to emerge.

2 Characteristics

This paper examines several means by which digital certificates may be revoked. To do so it's important to establish the metrics with which these various approaches can be analyzed. We claim there are at least four such characteristics:

** The opinions expressed in this paper are those of the author and not necessarily those of VeriSign Inc.

1. Population size and symmetry;
2. Timeliness of revocation information;
3. Connectivity and bandwidth utilization; and
4. Responsiveness to security-critical needs.

2.1 Population Size and Asymmetry

The absolute size of a population of potentially revocable certificates can strongly influence the approach taken. Absolute size can range across orders of magnitude. At one end of this spectrum we have closed communities with, say, fifty members, to in the worst case the entire Internet. Quite obviously a solution that meets the needs of the former in all likelihood may fail to adequately address the latter. Conversely a solution intended to address Internet-scale populations may in many cases require more resources and complexity than is necessary for more modestly sized enterprises.

One must also take into account the effects of population asymmetry. Efficiencies can be gained when the set of potentially revocable certificates is considerably smaller than the set of relying parties. While other architectures may emerge over time, there exist today two well-known instances of this situation: client-server type services and signed objects. In both cases the number of relying parties would exceed by orders of magnitude the number of certificates issued.

2.2 Timeliness of Revocation Information

How soon after a certificate is revoked does a relying party wish to know of the revocation? Within a week? A day? Immediately? The degree of timeliness relates to the interval between when a Certification Authority made a record of the revocation and when it made that information available to its relying parties. At first it would seem that CAs should strive to make this information available as soon as possible. All other things being equal, this is certainly a legitimate requirement. However the mechanism used to convey this information may consume an appreciable level of bandwidth. While it might be prudent to publish CRLs on an hourly basis, in all likelihood CRL(n+1) will contain no more information than CRL(n).

2.3 Connectivity and Bandwidth Utilization

Does the approach require the relying party to be online, or can the relying party ascertain a certificate's reliability using cached data? Clearly there exists scenarios where a relying party will be attempting to validate a certificate in off-line modes of operation. Online mechanisms further create a mission-critical component in the overall security design. However, online mechanisms can be applicable to a wide variety of electronic exchanges that already assume connectivity. This dimension to the problem can inform the designers of online mechanisms of the need to facilitate off-line caching of prior results.

2.4 Security Considerations

In the overwhelming majority of cases a certificate will expire without ever having been revoked. It is however those few circumstances when a certificate needs to be revoked that causes one to carefully consider the above mentioned characteristics. Without a doubt the most troubling scenario involves the compromise of a private key. Without an effective compromise recovery capability, a security solution based on PKI is at risk of general system compromise.

3 Certificate Revocation Options

3.1 Certificate Revocation Lists

The Certificate Revocation List has been a fixture of PKI standards for several years. The mechanism is well understood and is well supported across the relevant standards. The concept does however suffer from some widely recognized shortcomings.

First and perhaps most significantly, CRLs can grow arbitrarily large. One may delete expired certificates from a CRL, but the CRL remains a linear function of the population of certificates it covers. One means of reducing the size of a CRL is through partitioning. There exist proposals that partition CRLs according to some partitioning rule and include in the certificate the location of the partial CRL where that particular certificate would be listed should it be revoked. While this approach shows promise of managing CRL size, it remains to be seen if this capability will be implemented in anything other than a few proprietary systems.

Closely related to the sizing criticism, frequent distribution of CRLs may unnecessarily consume bandwidth. Assuming that a CRL listing 1000 certificates may run to about 50kb in size, periodic updates listing one additional certificate will transmit roughly 50 bytes of new information in addition to 50kb-50 bytes of redundant information.

CRLs do not provide a positive response; they do not speak to a certificate's existence. Another way to look at this aspect of CRLs is to ask: Does the absence of a certificate on a CRL imply the certificate exists? While in most cases a certificate processing system would have on hand the certificate in question, it is foreseeable that certificate processing systems can be developed and deployed that rely exclusively on a certificate identifier.

All that said, CRLs have their place in the global scheme of PKI. They form a least-common-denominator baseline. Even in the instances where alternative methods are used to enforce revocation locally (as will be discussed momentarily), CRLs may serve a role as a common interchange format across autonomous PKIs.

To place CRLs against the metrics defined earlier, they are most useful where:

1. Populations on the order of 10,000 certificates. An order of magnitude larger requires infrastructure capabilities that are both beyond the state of the art and further may not be operable in practice.

2. The number of relying parties is considerably larger than the number of certificates issued. Client-server architectures are one such system design pattern, as is the practice of signed code.
3. Timeliness is not the top priority. For the purposes of comparison, this would amount to a daily update vs. hourly or realtime updates.
4. High bandwidth environments that can easily handle the redundancy inherent in CRL distribution. CRLs are also useful in instances where the certificate processing system is not connected to the CRL distribution network, bearing in mind that the weight of this advantage is inversely proportional to the periodicity of CRL update.
5. CRLs provide basic mechanisms to deal with key compromise. Reason codes can be embedded into CRLs that can be used to partition key compromise CRLs from all other reasons.

In summary, CRLs are best used within an enterprise that is handling more or less typical PKI-secured traffic. To the extent that these characteristics defines a non-trivial subset (or market), CRLs can and must be considered a viable solution to the revocation problem.

3.2 Online Certificate Status Checking

It has been long known that timely reporting of unreliable public keys is crucial to the safety and security of a secure message handling infrastructure. In the absence of commercially available options, mission-specific solutions were developed that met this need in a DoD context. To move the matter forward into the commercial sector, the IETF's PKIX working has established that some means of online status checking can and should be anticipated over the evolution of PKI on the Internet. A recent proposal has been put forward to drive this solution to reality.

Status checking is particularly relevant to environments with severe time constraints. Federal reserve loans among major banks are a prime example. In that environment, interest is charged by the minute. To the extent that trades and transfers in this environment are secured using public key technology (and consequently public key certificates), there exists a compelling need for very timely status on the reliability of a certificate prior to accepting its use to validate a transfer. Does this type of solution bear risk of lost connectivity? Absolutely—to the same extent that those electronic transfers are themselves at risk of connectivity loss. Dimensions of timeliness, service availability and reliability would all serve to characterize the value of the service.

Online status is well suited to environments where:

1. Populations on the order of 10,000 certificates. An order of magnitude larger requires infrastructure capabilities that are both beyond the state of the art and further may not be operable in practice.
2. As with CRLs, efficiencies can be achieve in instances where the number of relying parties is considerably larger than the number of issued certificates.

3. Timeliness is of the highest priority, notwithstanding the ability to produce status responses that have a validity interval similar to that used in CRLs. This capability would allow for offline operations.
4. Online-oriented security protocols. In most client-server and server-server design patterns, connectivity is already an assumption. Online status can also be relevant to firewalls that are set up to assess the reliability of signed objects entering a secure enclave. In-band inclusion of status responses in security protocols can improve bandwidth utilization.
5. To the extent that online status is a direct reflection of CRLs, this mechanism can effectively address key compromise. A "push" model of online status responses can also be used to broadcast key compromise data in the instance where a wide-scale alert of such information is relevant to the overall security of the system.

3.3 Trusted Directory

Within an enterprise, one can effectively "revoke" a certificate on the basis of its absence in a trusted directory. Such can be the case when an employee leaves a company. Her account is deleted from the system, including its content of digital certificates. To the extent that applications are designed to check for certificates in the directory prior to relying on them, this enables an expedient solution to the immediate crisis of revocation capability. It does however invoke severe penalties. Simply put, the directory and all its components now become trusted elements and thus are prime targets for attack. This option thus transforms the proven reliability of a digital signature on a CRL to trusted software development, secure systems engineering, trusted operational procedures and trusted operating personnel. An independent, cryptographically trustworthy assertion— either as a CRL or a signed status response message—significantly reduces the costs, risks and complexity of this option.

The trusted directory approach is largely limited to a closed, well-connected enclave. It requires continuous connectivity to the directory for every certificate acceptance decision. In most cases a corporation's internal directory is not made available to external parties for reasons of privacy and security. While replication of internal directory content to external servers may reflect the existence of certificates, there nonetheless exists concern that the certificates themselves may contain proprietary information. Thus external parties will very likely have no means to determine if a certificate is revoked.

Despite these limitations it is nonetheless foreseeable that the trusted directory approach will be taken by some environments due to the expediency of its default behavior. With this eventuality in mind, the trusted directory approach:

1. Is as scaleable in population as the underlying directory technology reaches across inter-departmental and branch-office boundaries.
2. Can be responsive to timeliness concerns in that the absence of a certificate in its assumed directory entry would inhibit further reliance regardless of cause.

3. Requires constant connectivity to the directory if timeliness benefits are to be achieved. This approach however fundamentally fails to address the needs of the off-line user.
4. Can enforce compromise if connectivity can be assumed. The absence of a certificate in its assumed directory entry would inhibit reliance regardless of cause.

3.4 Short Lived Certificates

The presumption underlying this option is that in the absence of any other act, a certificate with a short validity interval will naturally bound the effects of revocation causes. Proposed intervals are typically on the order of weeks where current practice is on the order of years. There are some fundamental questions that need answering however:

Are new key pairs generated for each new certificate, or is the same key pair simply recertified? Public key validity renewal is today absent from commercial products as well, although sorely needed. Short certificates amplify the urgency of this requirement. In its absence one is led to conclude that new key pairs are generated for each new "ticket". This is a good deal of keying material to be generating, placing a greater reliance on key generation performance than has to date been asserted with well-defined models of PKI key management requirements.

Do the validity intervals overlap from certificate to certificate for the same subscriber? If not, then an enterprise PKI being sustained by this model must carefully consider the effects of delayed propagation of one's "next" certificate in the event of network failure. It's worth noting that short validity intervals reverse the freshness requirement, the need to distribute information to maintain freshness and the criticality of doing so reliably. This option does nothing to eliminate these essential requirements. If however validity intervals do overlap, will applications be savvy enough to disambiguate from among several equally valid subscriber certificates? Perhaps, but only at the expense of additional user interface and operational complexity.

This option also completely ignores the essential need to recovery from compromised private keys. It is not enough to simply require the owner of a compromised key to stop using it in favor of a fresh key pair. Outlying relying parties must be provided notice that the signatures they are processing—which validate with the still valid public key of the prior certificate—are no longer reliable.

As it relates to the proposed metrics, this option:

1. Responds effectively to population effects. Regardless of the size or symmetry of the population, the essential requirement to inhibit reliance on the certificate can be achieved to a first-order degree of compliance.
2. As it relates to timeliness metric, it fails to provide an asynchronous mechanism to notify relying parties of revocation events.
3. Suffers seriously from a need for frequent refreshes from centralized recertification services. In short, this option reduces public-key infrastructure to secret-key management.

4. Provides no means to deal with key compromise.

4 Conclusions

Independent of the technology and solution options there exists a less tractable problem of trust policy. Clearly the impact of state, national and international legislation bears respect in defining the extent of the revocation solution space. While the law historically describes actual usages of trade rather than create them, certificate issuance and acceptance policy has established a respected level of maturity. This body of knowledge would maintain that a CA not only has the right but may be held accountable for providing notice of revocation to its relying parties. Technologists considering options for revocation need to take into account their role as a enabler of such policies and practices.