A Taxonomy of Certificate Status Information Mechanisms[¶*]

J. S. ILIADIS[1], D. SPINELLIS[2], S. KATSIKAS[2], B. PRENEEL[3]

[1]Research Unit, University of the Aegean
Athens, 30 Voulgaroktonou St., GR-11472, Greece. e-mail: jiliad@aegean.gr

[2]Dept. of Information and Communication Systems, University of the Aegean
Samos GR-83200, Greece. email: {dspin, ska}@aegean.gr

[3]Dept. Electrical Engineering-ESAT, Katholieke Universiteit Leuven
K. Mercierlaan 94, B-3001 Heverlee, Belgium. email: bart.preneel@esat.kuleuven.ac.be

Abstract

A number of mechanisms have been proposed for generating and disseminating information on the status of certificates. Their operation is different, if not contradicting sometimes, and advantages and disadvantages depend on the requirements of the underlying PKI. PKI designers and implementors should perform a small scale study before deploying such a mechanism in a specific PKI, in order to select the most suitable mechanism for their environment. This paper presents a method for categorising Certificate Status Information mechanisms, depending on their elementary functionality. This taxonomy can be used as a guide for selecting CSI mechanisms to be used in large-scale PKI deployment efforts.

Keywords
certificate revocation, certificate status, certificate revocation lists, taxonomy

---

[¶] In *Information Security Solutions Europe ISSE 2000*, Barcelona, Spain, September 2000. European Forum for Electronic Business.

1. INTRODUCTION

The adoption of public-key cryptography as a basis for electronic commerce and other security-related information technology applications has brought to the surface a number of issues related to the deployment of a large public-key infrastructure (PKI). Public-key certificates — public keys signed by a trustworthy entity — are subject to subsequent revocation. Certificates can be revoked if e.g. a user's private key has been compromised, the authority that issued the certificate ceases to certify a given user, or the authority's certificate is compromised. Timely availability of correct revocation information is essential to build trust in digital signatures and applications built on digital signatures and PKIs. The validity of certificates (Certificate Status Information — CSI) must therefore be communicated by the *CSI provider,* the entity that is responsible for maintaining CSI (a certification authority or a party authorised to act on its behalf), to the *dependent entity,* the entity that relies on data in the certificate to make decisions. The mechanisms used for providing CSI have important implications on the management, security, and performance of the PKI that relies on them. As public-key certificates enter the mainstream, they will be issued and used in large numbers by less sophisticated users, and be relied on for a large number of important transactions. For these reasons, a PKI must be based on a CSI distribution mechanism, that provides flexibility and transparency in the management side, timeliness and scalability on the performance side, while guaranteeing high levels of trust and security. In this paper we present a method for categorising Certificate Status Information mechanisms, depending on their elementary functionality. We present the elementary functions a Certificate Status Information mechanism comprises of, and provide a way to analyse the capabilities of a CSI mechanism according to these functions. This taxonomy can be used as a guide for selecting CSI mechanisms to be used in large-scale PKI deployment efforts.

This paper is organised as follow: in Section 2 we present the methodology we use to identify the elementary functionality and produce a taxonomy of CSI mechanisms. In Section 3 we present a short overview for every mechanism we examine, accompanied by a table that outlines the elementary functionality of the mechanism according to our methodology. In Section 4 we present a comparative evaluation of the mechanisms we examine, based on the aforementioned taxonomy. Finally, in Section 5 we present our concluding remarks.

2. METHODOLOGY

CSI mechanisms are operated by one or more entities. These entities undertake the task of identifying the need to generate new CSI, generating and storing CSI in a way that makes it available to entities who depend on CSI (dependent entities). The latter are entities who use the status information during the process of certificate validation, in order to authenticate or authorise the respective certificate holder.

The elementary functions a **CSI provider** performs are the following:

1. The identification of the need for revoking a certificate. The CSI provider has to provide to external entities the capability of communicating revocation requests. When the CSI operator receives such information, he proceeds with verifying their validity and, if they are valid, generates the respective Certificate Status Information. Specific mechanisms and procedures have to be in place, both for the external entities to communicate revocation requests to the provider and for the provider to verify the validity of such requests.
2. The generation of Certificate Status Information. A CSI provider uses the information collected in the previous stage in order to generate CSI. The CSI format used in this stage might not be the one that will be used for CSI dissemination to the dependent entities. The purpose of generating CSI at this stage is not to disseminate it to the dependent entities, but to maintain a trusted repository of CSI. The characteristics of the CSI

generation function include the frequency of CSI generation, and the format and size of the produced CSI.

3. The storage of CSI. This function relates to the mechanisms used by the CSI provider in order to protect the generated CSI, and render it readily available to the dependent entities. These mechanisms may be part of the CSI mechanism itself; alternatively, external mechanisms could be used in order to support the protection and availability of the generated CSI.

On the **dependent entity** side, the elementary CSI functionality comprises of:

1. The location function. Dependent entities need trusted information in order to locate CSI regarding a certificate they wish to validate. Also, the dependent entities need to know the mechanisms they have to use in order to communicate with the CSI provider at the specified location.
2. The retrieval CSI. This function concerns the mechanisms dependent entities use in order to retrieve trusted copies of CSI they are interested in.
3. The validation of CSI and of the respective certificates. Dependent entities verify the integrity and authenticity of CSI they receive. Also, they validate it, verify that the CSI they received contains information on the certificate they wish to validate. Finally, they validate a certificate, based on the retrieved CSI.

3. TAXONOMY OF CSI MECHANISMS

3.1 Certificate Revocation List

Certificate Revocation Lists were the first CSI mechanism to be standardised [ISO9594], [Hous99]. CRLs are issued by CAs periodically, for the dependent entities to know whether they have in their possession the latest revocation information available. Therefore, the dependent entities are not capable of possessing fresh revocation information on a need-to-know basis. Another problem this mechanism faces is that the CRL grows as certificates get revoked. This can lead to a very large CRL which will be difficult to communicate to dependent entities and be installed by them in their local storage media, which may have a restricted storage space.

In order to deal with the aforementioned problems, two solutions have been proposed [ISO9594], [Hous99]: Distribution Points and Delta-CRLs. Distribution Points provide the means to partition a CRL. The *cRLDistributionPoints* X.509v3 certificate extension is optional. If it exists, it must point to a valid URI (*DistributionPointName*) from which a specific CRL can be downloaded; this CRL is the one that will contain the revocation information for that specific certificate, once it is revoked.

Delta-CRLs intend to address the problem of using up too much of the available network resources when communicating the CRL either as a whole, or even in parts through Distribution Points. Delta-CRLs provide the means for constructing incremental CRLs. Whenever new revocations have taken place, the (new) CRL that dependent entities have to retrieve contains only this new certificate revocation information. A Delta-CRL must contain a *DeltaCRLIndicator*, so the dependent entities will know which CRL it is that a specific Delta-CRL complements.

CRLs do not specify a mechanism for collecting revocation requests from the aforementioned external entities. CSI providers must provide such a mechanism for requesting revocation of a certificate or notifying the CSI provider (or the CA) of information that relates or could lead to the revocation of a certificate. The mechanism specified in [Adams99] could be used for these reasons. However, it is not mandated by the CRL mechanism itself.

Entities external to the CSI provider can request revocation of a certificate. These entities inform the provider whenever they have reasons to believe a specific certificate needs to be revoked. These entities can be either the certificate holders themselves or other entities. The CSI provider (usually the CA that issued the certificate) has to validate these revocation requests and proceed with revocation of the specific certificate.

CRLs are issued periodically. In the time between the issuance of two consecutive CRLs, the CSI that has been collected and validated from submitted revocation requests is stored in a format and repository the CSI provider finds suitable. Dependent entities never receive CSI in this format, and never access directly the repository where CSI is stored in this format, therefore they do not need to know either how to interpret CSI in this format or how to access the aforementioned repository.

Dependent entities receive CSI in the format of a CRL [Hous99], [ISO9594]. This is a standardised format, and X.509.v2-aware dependent entities should be able to interpret it. The CSI provider uses the information in his internal, protected repository of CSI in order to generate a CRL.

Protection of the CSI, when it is stored in the non-standard format and protected repository, must be ensured by the CSI provider using protection mechanisms he considers to be suitable. Protection of the CSI, when it is CRL-formatted, is achieved by the digital signature of the CSI provider on the data contained in the CRL.

Table 1: Certificate Revocation Lists	
Collection of revocation requests	No mechanism specified; [Adams99] could be used.
Generation of CSI	CA generates primary CSI in proprietary format. This CSI is used for the periodic issuance of CSI to be disseminated (CRL).
Storage of CSI	CSI provider maintains repository of CRLs.
CSI location function	URIs pointing to CRLs contained in certificates.
CSI retrieval function	Format: CRL.
CSI and certificate validation function	Signature of CA on CRL. CA identification information in CRL. Certificate serial numbers contained in CRL (negative CSI assertions).

3.2 Freshest CRL

Adams et al [Adams98] propose the use of Delta-CRLs, issued on top of partitioned CRLs (CRL Distribution Points). Their method uses two certificate extensions: the standardised *crlDistributionPoint* X.509v2 CRL extension and a custom extension which they call Freshest Revocation Information Pointer (FRIP). The latter points to a CRL or Delta-CRL which has as a base the partitioned CRL pointed at by the *crlDistributionPoint*. The aforementioned CRL or Delta-CRL, which is called Freshest CRL (FCRL), can be issued very frequently and not have a fixed update granularity. Therefore, the dependent entities that need very fresh CSI could get hold of the latter, at the expense of downloading another CRL. One of the advantages of this method is that the implementation requires no major changes in the mechanisms already used by the CAs. The *crlDistributionPoints* certificate extension is a standardised extension [ISO9594] and the Freshest Revocation Information Pointer is a custom extension that could be added to the certificates issued by the specific CA.

This mechanism differs from the standardised CRL CSI mechanism only in the location function. There is additional location information that permits to the dependent entities to locate a Freshest CRL, that is a Delta-CRL issued on top of a Distribution Point CRL.

Table 2: Freshest CRL	
Collection of revocation requests	No mechanism specified; [Adams99] could be used.
Generation of CSI	CA generates primary CSI in proprietary format. This CSI is used for the periodic issuance of CSI to be disseminated (CRL)
Storage of CSI	CSI provider maintains repository of CRLs
CSI location function	URIs pointing to CRLs contained in certificates
CSI retrieval function	Format: CRL
CSI and certificate validation function	Signature of CA on CRL CA identification information in CRL Certificate serial numbers contained in CRL (negative CSI assertions)

3.3 Redirect CRL

Adams *et al.* [Adams98] suggest the use of another CRL custom extension, the Redirect Pointer. This could be used in combination with Distribution Points to allow for dynamic re-partitioning of the CRL. If a CRL fragment contained in a Distribution Point grows to be unmanageably large, then the CSI for a subgroup of the certificates contained in that CRL fragment could be moved into another, possibly new, Distribution Point. Adams *et al*. propose to install a pointer as a custom CRL extension in the original CRL fragment that points to the new CRL fragment and specifies the scope of certificates covered by that new CRL fragment. This would ensure that the dependent community will be able to continue receiving CSI without any disruption. The advantage this method presents is the dynamic re-partitioning of the CSI space, which can become a necessity if the devices used to store the CSI (regarding a specific, restricted group of certificate holders) have a limited storage capability.

This mechanism differs from the standardised CRL CSI mechanism only in the location function. In this mechanism, CSI location information is not contained in total inside the certificate itself. Also, this mechanism introduces a minor change in the generation of CSI.

Table 3: Redirect Certificate Revocation List	
Collection of revocation requests	No mechanism specified; [Adams99] could be used.
Generation of CSI	CA generates primary CSI in proprietary format. This CSI is used for the periodic issuance of CSI to be disseminated (CRL) CSI space is partitioned dynamically.
Storage of CSI	CSI provider maintains repository of CRLs.
CSI location function	URIs pointing to CRLs contained in certificates. Certain CRLs may contain URIs to other CRLs (segmentation of CSI space).
CSI retrieval function	Format: CRL.
CSI and certificate validation function	Signature of CA on CRL. CA identification information in CRL. Certificate serial numbers contained in CRL (negative CSI assertions).

3.4 Enhanced CRL Distribution Points

Hallam *et al.* [Hall98] had proposed a separation of the location function from the validation function, using a CRL extension called Status Referrals. This extension can be used to convey information regarding the newly issued CRLs. A CRL that contains *StatusReferrals* extensions does not contain certificate status information. Such a CRL is used in order to provide the dependent entities with information on the location of the CRLs they are interested in.

Hallam *et al.* [Hall98] also proposed the use of the *cRLScope* extension as a mechanism for implementing the validation function. Once a dependent entity locates a CRL through the use of *StatusReferrals* extensions, that entity can decide whether the located CRLs contain the required CSI. Multiple *PerCAScope* entries could be used in order to provide for Indirect CRLs, or even as another mechanism for implementing Redirect CRLs.

This mechanism can reduce the unneeded downloading of CRLs that have not been updated yet and enables the user as well to find out whether a CRL has been issued ahead of time or not, without actually downloading the CRL itself.

Table 4: Enhanced CRL Distribution Points	
Collection of revocation requests	No mechanism specified; [Adams99] could be used.
Generation of CSI	CA generates primary CSI in proprietary format. This CSI is used for the periodic issuance of CSI to be disseminated (CRL) CSI space is partitioned dynamically
Storage of CSI	CSI provider maintains repository of CRLs
CSI location function	URIs pointing to CRLs contained in certificates A CRL may contain only URIs (StatusReferrals) to other, newly issued CRLs
CSI retrieval function	Format: CRL
CSI and certificate validation function	Signature of CA on CRL CA identification information in CRL Verification of scope of CRL Certificate serial numbers contained in CRL (negative CSI assertions)

3.5 Positive CSI

Rivest [Rivest98] argues that CRLs are not needed at all in order to convey CSI. He claims that CRLs are probably the wrong mechanism to use for disseminating CSI because they contain negative statements instead of positive ones and because it is the issuer and not the dependent entity that sets the requirements on the freshness of the CSI.

Positive statements on the status of a certificate can be constructed and communicated in many ways [Micali96], [Naor98]. Rivest [Rivest98] proposes another mechanism for communicating positive CSI. According to this mechanism, it is the certificate holder who should revoke his own certificate, by signing with his own private, compromised key a "suicide note" (SN). There should be a network of "Suicide Bureaus" (SB), which gather suicide notes from every possible source, and either replicate the information they hold or have a means to refer queries to each other.

When a dependent entity needsCSI it can ask for fresh CSI from the certificate holder; the latter, in turn, should ask a SB for a "certificate of health", stating that ‘no evidence has been received that the key has been lost or compromised’ [Rivest98]. The dependent entity could set in this case requirements on the freshness of the "certificate of health" provided by the SB to the certificate holder and by the latter to the dependent entity.

According to Rivest, the dependent entity should be able to revoke a certificate by itself (e.g. in case the dependent entity is a service provider and it notices that there is more than one entity that holds that certificate and makes, possibly illegitimate, use of it. In order to enable the dependent entity to revoke a certificate of an authenticating entity, the latter could be asked to sign a "suicide note" before having the right to use the service. Thus, the dependent entity could send the suicide note to SBs whenever the dependent entity believes that the certificate is being used illegitimately by more than one entities or has been compromised, without having to communicate with the authenticating entity and without the latter having to produce the suicide note at that time.

Table 5: Positive CSI	
Collection of revocation requests	Suicide Notes (SNs).
Generation of CSI	CSI providers does not have to generate primary CSI (SNs). The CSI revocation request serves as primary CSI too. SNs are used in order to generate certificates of health, upon request .
Storage of CSI	Suicide Notes. Cache repository of certificates of health (provisional).
CSI location function	The authenticating entity provides the dependent entity with CSI.
CSI retrieval function	Format: Certificates of health.
CSI and certificate validation function	Signature of SB on certificate of health. SB identification information on certificate of health. Certificate of health concerns and identifies only one certificate. Positive CSI assertion.

3.6 Certificate Revocation Status

The use of X.509v2 CRLs results in a communication overhead mainly from the CRL repository to the dependent entities. The Certificate Revocation Status (CRS) [Micali96] is a revocation mechanism that attempts to address that. In CRS, the CA has to include in every certificate two random or pseudorandom 100-bit values YES (Y) and NO (N). Initially, the CA has to decide on the CSI update granularity and calculate the number of CSI updates it will perform for the certificate it is going to issue, within the certificate's validity period. The CA produces two random or pseudorandom numbers Y_0 and N_0. If the number of CSI updates that are going to be performed for that certificate is i, the CA calculates Y by applying a hash function F to Y_0 i consecutive times. N is derived from N_0 by applying F once to N_0. Therefore, the certificate contains the following two values, in addition to its usual contents:

1. $N=F(N_0)$
2. $Y=F^i(Y_0)$

The CA communicates with the CSI repository regularly (the update granularity is predefined and CA-specific), and sends the following data:

1. a list of all the serial numbers of certificates that have been issued and are not expired yet, signed by the CA
2. for each such certificate, the CA also sends a 100-bit value K, where $K=N_0$ if the certificate has been revoked and $K=F^{i-j}(Y_0)$, if the certificate has not been revoked; j represents the number of CSI updates that have been performed since the issuance of the certificate.

The entity requesting CSI from the CSI repository will retrieve K. That entity also retrieves Y,N from the certificate and calculates:

1. $F^j(K)=Y$
2. $F(K)=N$

If (1) applies then the certificate has not been revoked, while if (2) applies then the certificate has been revoked. If neither of these two apply, the dependent entity should request from the CSI repository the signed list of all the serial numbers of certificates that have been issued and are not expired yet. If the certificate in question is in that list, the dependent entity should conclude that the CSI repository has not sent him the correct number K which has been sent to the repository by the CA. Depending on the integrity and authentication mechanisms used for the communication between the dependent entity and the CSI repository, the dependent entity should draw its conclusions about the reason why neither of the aforementioned conditions (1) and (2) applied.

The main advantage of this mechanism is that it significantly reduces the communication costs between the CSI repository and the dependent entity, by employing a mechanism for the CSI dissemination which contains positive statements regarding the status of a certificate. Furthermore, the advantage this mechanism presents over others is that positive statements are employed and that the CSI repository does not have to be trusted by the dependent entity.

An addition to this mechanism [Micali96] is to have the CA give also *full revocation certificates* to the CSI repository. These certificates could contain a revocation timestamp of the certificate and the revocation reason. If the dependent entities would like to have more information on the revocation of a specific certificate they could request for a *full revocation certificate* from the CSI repository.

Table 6: Certificate Revocation Status	
Collection of revocation requests	No mechanism specified.
Generation of CSI	CA generates primary CSI in proprietary format. This CSI is used for the periodic issuance of CSI to be disseminated (K values and CA-signed list).
Storage of CSI	Y_0, N_0 values and CA-signed list.
CSI location function	No mechanism specified.
CSI retrieval function	Format: Value K and CA-signed list (if needed)
CSI and certificate validation function	Integrity/Authenticity mechanism for value K not direct; if K does not return expected results, CA-signed list provides integrity/authenticity verification of K Each value K concerns only a specific certificate

3.7 Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP) [Myer99] is a protocol proposed by the IETF PKIX Working Group that allows dependent entities to query for CSI in a more timely fashion than CRLs. OCSP could be used in conjunction with CRLs. It provides an extension that can be used as a pointer to a CRL, in case more timely CSI is unavailable at a certain point of time.

The responses to CSI queries returned by OCSP are digitally signed. The authority that runs the OCSP service can either be the CA itself, another entity that is designated by the CA as a

CSI provider (CA Designated Responder) or an entity trusted by the dependent entity to provide CSI (Trusted Responder). A CA Designated Responder must possess a specially marked certificate, issued by the CA, which authorises it to provide CSI to requestors. The requests for OCSP service can be signed by the requestors themselves.

OCSP includes the CSI location function inside the certificate itself. CAs that support the use of OCSP for disseminating CSI should include in the certificates they issue the AuthorityInfoAccess extension [Hous99], as a pointer to the location of the authority that provides OCSP service for the specific certificate.

The possible OCSP responses are the following three:

1. "good", meaning that the certificate in question is not revoked. In the current version of [Myer99], it is mentioned that this response does not indicate that the certificate has ever been issued or that the OCSP response was produced within the validity interval of the certificate. Further CSI will be provided through the use of response extensions, which have not yet been specified in [Myer99]. Therefore, at its current status, OCSP provides only negative CSI, like CRLs.
2. "revoked", this indicates that the certificate has been revoked or has been suspended ('suspension' in OCSP terminology is equivalent to the certificateHold revocation reason code in CRLs).
3. "unknown", this response indicates that the OCSP service is not aware of the certificate in question.

Table 7: Online Certificate Status Protocol	
Collection of revocation requests	No mechanism specified.
Generation of CSI	CA generates primary CSI in proprietary format. This CSI is used by the OCSP service provider (direct access or replicated) to generate OCSP Responses, upon request.
Storage of CSI	No CSI is stored (except for the primary CSI, if it is replicated locally to the OCSP service provider).
CSI location function	URI pointing to OCSP service provider is contained in certificate; if timely CSI is not available, OCSP Response points to a CRL.
CSI retrieval function	Format: OCSP Responses. Authentication of dependent entities (provisional).
CSI and certificate validation function	OCSP Responses signed by OCSP service provider. OCSP Responses contain CSI for a specific certificate only. OCSP do not contain complete CSI.

Dependent entities that request CSI from an OCSP service provider must be able to check the revocation status of the certificate of the service provider himself. According to [Myer99] CAs may choose to provide that functionality in the following three ways:

1. The CA could issue only short-lived certificates for OCSPs in order to avoid having them revoked.,
2. the CA may choose to specify an extension in the certificate the OCSP service provider that points to a CRL, or
3. the CA could choose not to specify any method for OCSP certificate status verification.

Dependent entities use the hash of the CA that issued a certificate for an entity, the hash of the public key contained in that certificate and the certificate serial number in order to form a CSI query for the OCSP service provider. Using hashes, the amount of information communicated is less and at the same time there is only a negligible chance of two sets of identification

information (the hashes we have mentioned above) to collide, if the hash function has a sufficiently large range and is collision-resistant. Using the aforementioned hashes, only an entity that already holds the certificate in question can create the appropriate hashes and request for CSI from the OCSP service provider.

3.8 Freshness-constrained Revocation Authority

Stubblebine [Stub95] proposes a revocation service where revocation can be definite, and where the repositories of revocation information need not be trusted. According to this service, the role of the Certification Authority (CA) is separated from the role of the Revocation Authority (RevA). The CA issues long-term certificates, which contain freshness constraints on the CSI the dependent entities will use in order to validate the certificates. Such a certificate also contains a pointer to the RevA that is responsible for issuing CSI regarding the specific certificate. The RevA issues frequently timestamped certificates which are used in order to provide the dependent entities with positive assertions regarding the validity of the certificate. The dependent entities themselves impose their own CSI freshness requirements, when certificate holders use their certificates in order to authenticate themselves. Another level of CSI freshness constraints can be imposed if there are higher-level CAs, Policy CAs (PCA), that issue certificates for the lower-level CAs. Policy CAs may impose their own CSI freshness requirements on the end-entities certificates, contained in the certificates of the lower-level CAs.

When a certificate holder attempts to authenticate himself based on his public key, the dependent entity will check the CSI freshness requirements imposed by the issuing CA. In addition, the dependent entity will impose its own freshness requirements, it will locate the RevA that corresponds to the certificate of the authenticating entity and retrieve a short-lived RevA certificate that meets these freshness requirements. The combined CSI freshness requirements, the timestamp on the certificate issued by the RevA and the current time are the information the dependent entity will use in order to verify the validity of the certificate presented by the certificate holder.

This method allows for flexible balancing of the authentication costs and level of protection on a per transaction basis. Furthermore, CSI (the short-lived, timestamped certificate) need not be communicated from a trusted repository. The RevA can replicate the frequently issued, short lived certificates to non-trusted repositories. Therefore, this method is efficient even when the network infrastructure is not reliable. Moreover, timestamped CSI is more flexible compared to CSI that contains expiration dates since the former can be used in environments with different CSI freshness requirements.

Table 8: Freshness-constrained Revocation Authority

Collection of revocation requests	No mechanism specified.
Generation of CSI	RevA generates primary CSI in proprietary format. This CSI is used to generate short-lived RevA certificates, upon request.
Storage of CSI	No CSI is stored except for the primary CSI. Cache repository of short-lived RevA certificates (provisional).
CSI location function	URI pointing to RevA is contained in the certificate.
CSI retrieval function	Format: Short-lived RevA certificates, communicated from the RevA.
CSI and certificate validation function	CSI (short-lived certificates) are signed by RevA. Each short-lived certificate provides CSI for a specific certificate only.

The aforementioned method allows the delegation of the revocation service to an authority other than the CA, but at the same without requiring, expecting or depending on the RevA to specify revocation policies. CSI freshness requirements are specified both from the hierarchy of the CAs and from the dependent entities themselves.

4. COMPARATIVE EVALUATION

A mechanism for collecting revocation requests or notices is specified only in Positive CSI. Certainly, the mandatory use of such a mechanism may be restricting the way certificate holders (or other entities) can communicate revocation requests to the appropriate authorities. In Positive CSI no entities other than the certificate holder himself or entities to which the certificate holder had explicitly given a Suicide Note (e.g. entities that provide services to the certificate holder) can request the revocation of the aforementioned certificate. If a certificate holder does not issue a priori a Suicide Note, and looses his private key (or access to it) he will not be able to revoke his certificate.

Contrary to Positive CSI, the other mechanisms allow any entity to notify the appropriate authorities of reasons or facts that should lead to the revocation of a certificate. However, the lack of a specific mechanism for authenticating and validating revocation requests presents disadvantages as well. If such a mechanism does not exist, the authorities that have the responsibility of collecting and validating revocation requests are vulnerable to Denial of Service attacks, since any entity can send (possibly malformed) revocation requests to these authorities, and the latter will have to examine them, in order to validate them and proceed with revocation of a certificate or reject the revocation request.

The location function in mechanisms 1, 2, 7, and 8 (CRL, FCRL, OCSP, Freshness-constrained RevA) is embedded in the certificate of the authenticating entity. Therefore, the dependent entity knows beforehand where is the CSI located, and the approximate time and respective cost of communicating with that location and retrieving the appropriate CSI. Therefore, the dependent entity can take into consideration the time and cost of retrieving CSI (which could be high, especially in mechanisms 1 and 2) and decide to retrieve it or not, depending on whether the the benefit of obtaining that information outweighs the cost of retrieving it.

Mechanisms 3 and 4 (RCRL, Enhanced DP CRL) support a feature that addresses, among others, the problem of high CSI communication costs: dynamic re-partitioning of CSI space. However, in mechanism 3, the dependent entity has to trust that the CSI provider will have partitioned the CSI to a satisfying extent, in order to reduce the communication cost. In any case, the dependent entity cannot estimate the time and cost of retrieving CSI with mechanism 3, before retrieving it. Mechanism 4 is slightly more effective in this point; this mechanism prevents the dependent entity from retrieving CSI it has already retrieved, thus reducing the communication costs from repeated retrieval of the same CSI.

In CRS, dependent entities have no specific means to locate the CSI they are interested in. Although such a mechanism is not clearly defined in CRS, the location functions of the other mechanisms we examine could be used in this case as well (e.g. URI in certificate).

In Positive CSI, the dependent entity does not need to locate CSI. The dependent entity requests CSI from the authenticating entity and it is the latter who locates and retrieves the requested CSI on behalf of the dependent entity.

Mechanisms 1-4 (CRL, FCRL, RCRL, Enhanced DP CRL) involve large communication costs (CSI provider to dependent entities) at the beginning of the period when new CSI becomes available. On the contrary, the rest of the mechanisms spread these communication costs over time. The only exception is CRS, in case the untrusted CSI provider is acting maliciously and does not provide dependent entities with the correct K values; in this case, dependent entities will be retrieving the CA-signed list of serial numbers of issued-and-not-yet-expired certificates, therefore communication costs may rise significantly.

Mechanisms 5, 7, and 8 (Positive CSI, OCSP, Freshness-constrained RevA) involve dynamically preparing CSI, signing it and sending it to the dependent entity who requested it.

Table 9: Comparative Evaluation of Certificate Status Information Mechanisms

	CRL	**FCRL**	**RCRL**	**Enhanced DP CRL**	**Positive CSI**	**CRS**	**OCSP**	**Freshness constrained RevA**
Collection of revocation requests	No mechanism specified	No mechanism specified	No mechanism specified	No mechanism specified	Suicide Notes to Suicide Bureaus	No mechanism specified	No mechanism specified	No mechanism specified
Generation of CSI	proprietary ↓ CRL	proprietary ↓ CRL	proprietary ↓ CRL Dynamic re-partitioning	proprietary ↓ CRL Dynamic re-partitioning	SN ↓ cert. of health	proprietary ↓ K values and CA-signed list	proprietary ↓ OCSP Responses	proprietary ↓ short-lived certificates
Storage of CSI	CRL	CRL	CRL	CRL	SN	Y_0, N_0, and CA-signed list	No CSI stored	No CSI stored
CSI location function	URI in certificate	URI in certificate	URI in certificate and URI in CRL	URI in certificate and URI in CRL	Authenticating entities provide CSI	No mechanism specified	URI in certificate	URI in certificate
CSI retrieval function	CRL	CRL	CRL	CRL	Certificate of health	K, signed list	OCSP Responses	RevA certificates
CSI and certificate validation function	CA signature, ID info Certificate serial numbers	CA signature, ID info Certificate serial numbers	CA signature, ID info Certificate serial numbers	CA signature, ID info Certificate serial numbers CRL scope	SB signature, ID info Positive CSI on a specific certificate	Indirect integrity, authenticity CSI on a specific certificate	OCSP signature, ID info CSI on a specific certificate Incomplete CSI	RevA signature, ID info CSI on a specific certificate

However, signing each piece of CSI requested by dependent entities is a computational overhead and it could facilitate Denial of Service (DoS) attacks. Pre-computed responses that have a short validity period could be a solution to this problem, but they render the CSI provider open to replay attacks, where someone could replay CSI responses before their expiration date but after a certificate has been revoked.

The requests for OCSP service can be signed by the requestors themselves. This is a useful feature, because entities that offer the OCSP service could have dependent entities authenticate before they deliver the CSI they request. One of the uses for such an authentication could be to allow OCSP CSI providers to charge for the service they offer. If

PKIs will be used extensively in the commercial world, a natural consequence will be that CSI can be charged for [Fox98].

5. CONCLUSIONS

The techniques and mechanisms that can be used for providing CSI services cover a large spectrum of design, implementation, management, performance, and security options. PKI designers and implementors need to evaluate and decide on a CSI mechanism when deploying a PKI. Different tradeoffs may exist depending whether the PKI serves a closed user group or addresses a wider community.

We hope that the taxonomy we presented can provide a basis for evaluating CSI mechanisms and selecting specific ones to deploy in PKIs. Based on the methodology we used to derive this taxonomy, other CSI mechanisms can be reviewed, compared, and evaluated. In the future we would like to expand the comparative evaluation of the mechanisms examined in our taxonomy to include specific metrics and prescriptive selection guidelines.

6. REFERENCES

[Adams98] Adams C., Zuccherato R., A General, Flexible Approach to Certificate Revocation, Entrust Technologies

[Adams99] Adams C., Farrell S., Internet X.509 Public Key Infrastructure Certificate Management Protocols, Request for Comments 2510, 1999, available at http://www.ietf.org/rfc/rfc2510.txt

[Fox98] Fox B., LaMacchia B., Certificate Revocation: Mechanics and Meaning, In Proceedings of Financial Cryptography 98, LNCS 1465, New - York, Springer - Verlag

[Hall98] Hallam-Baker P., Ford W., Enhanced CRL Distribution Options, IETF PKIX Working Group, Internet Draft, 7 August 1998, available at http://www.ietf.org/internet - drafts/draft-ietf-pkix-ocdp-01.txt

[Hous99] Housley R., Ford W., Polk W., Solo D., Internet X.509 Public Key Infrastructure Certificate and CRL Profile, IETF Network Working Group, Request for Comments 2459 (Category: Standards Track), January 1999, available at http://www.ietf.org/rfc/rfc2459.txt

[ISO9594] ISO/IEC 9594-8 (1994), Open Systems Interconnection - The Directory: Authentication Framework. The 1994 edition of this document has been amended by the Draft Amendments [Dram96] and a Technical Corrigendum [Cor95]

[Micali96] Micali S., Efficient Certificate Revocation, Technical Memo 542b, Laboratory for Computer Science, Massachusetts Institute of Technology, March 1996

[Myer99] Myers M., Ankney R., Malpani A., Galperin S., Adams C., X.509 Internet Public Key Infrastructure Online Certificate Status Protocol, IETF Network Working Group, Request for Comments 2560 (Category: Standards Track), January 1999, available at http://www.ietf.org/rfc/rfc2560.txt

[Naor98] Naor M., Nissim K., Certificate Revocation and Certificate Update, In Proceedings 7th USENIX Security Symposium, Jan 1998, San Antonio, Texas

[Rivest98] Rivest R., Can We Eliminate Revocation Lists?, In Proceedings of Financial Cryptography 1998, available at http://theory.lcs.mit.edu/~rivest/revocation.ps

[Stub95] Stubblebine S. G., Recent-Secure Authentication: Enforcing Revocation in Distributed Systems, In Proceedings IEEE Symposium on Research in Security and Privacy, pages 224-234, May 1995, Oakland

ACKNOWLEDGEMENTS

This work was partially funded by the European Commission (Directorate General III, contract #ETD/99/502536). Bart Preneel is a research associate, sponsored by the National Fund for Scientific Research, Flanders (Belgium).