ON THE DISSEMINATION OF CERTIFICATE STATUS INFORMATION

by

Ioannis Iliadis

A thesis submitted in partial fulfilment of the requirements for the degree of

MSc in Information Security

Supervisor: Prof. Chris Mitchell

Royal Holloway and Bedford New College
University of London

1999

ACKNOWLEDGMENTS

I would like to thank my friends and colleagues from the University of the Aegean, for giving me the opportunity to work at an area I enjoy. I would also like to thank the State Foundation for Scholarships for providing me with financial support for my postgraduate studies; it has been much appreciated. Also, I would like to thank my supervisor, Prof. Chris Mitchell, for his support on my MSc dissertation. Finally, I would like to thank my family for helping me get there.

ABSTRACT

There has been an increasing interest in the deployment of Public Key Infrastructures, the past few years. Security issues emerge from the operation of Certification Authorities, as well as the operation of other PKI-related security service providers. Most of them have been addressed and efficient solutions have been found. One of the areas which has to be studied further is the generation and dissemination of information regarding the status of a digital certificate.

In this dissertation, we present a set of evaluation criteria for mechanisms that are used to generate and disseminate Certificate Status Information (CSI). We evaluate the proposed CSI mechanisms according to the aforementioned criteria, and identify the security and performance issues that emerge from their use.

Finally, we develop a prototype specification for a CSI dissemination mechanism, which we call Alternative Dissemination of Certificate Status Information (ADOCSI). This mechanism uses the functionality offered by Software Agents in order to disseminate CSI, and also uses some of the properties and functionality offered by the other CSI mechanisms. We believe that ADOCSI addresses some of the issues that emerge from the use of the other Certificate Status Information dissemination mechanisms.

Ioannis Iliadis
jiliad@aegean.gr
jiliad@bcs.org.uk

Table of Contents

1. DISSEMINATION OF CERTIFICATE STATUS INFORMATION 1

1.1 INTRODUCTION 1
1.2 CONTENTS OF REPORT 1

2. TAXONOMY OF CERTIFICATE STATUS INFORMATION MECHANISMS 2

2.1 CERTIFICATE REVOCATION LIST 2
2.2 FRESH REVOCATION INFORMATION 4
2.3 REDIRECT CRL 4
2.4 ENHANCED CRL DISTRIBUTION OPTIONS 4
2.5 POSITIVE CSI 5
2.6 CERTIFICATE REVOCATION STATUS 6
2.7 ONLINE CERTIFICATE STATUS PROTOCOL 7
2.8 FRESHNESS-CONSTRAINED REVOCATION AUTHORITY 8

3. EVALUATION OF CERTIFICATE STATUS INFORMATION MECHANISMS 10

3.1 EVALUATION CRITERIA 10
3.1.1 Type of Mechanism 10
3.1.2 *Efficiency 10*
3.1.3 *Security 11*
3.2 EVALUATION OF CSI DISSEMINATION MECHANISMS 12
3.2.1 Certificate Revocation List 12
3.2.2 Fresh Revocation Information 15
3.2.3 Redirect CRL 16
3.2.4 Enhanced CRL Distribution Options 16
3.2.5 Positive CSI 17
3.2.6 Certificate Revocation Status 19
3.2.7 Online Certificate Status Protocol 21
3.2.8 Freshness-constrained Revocation Authority 23

4. TOWARDS AN ALTERNATIVE MECHANISM 26

4.1 AGENTS 28
4.2 ADOCSI 29
4.2.1 AMP Location Function 31
4.2.2 CSI Location, Validation and Retrieval 33
4.2.3 Certificate Path Validation 34
4.2.4 Interface Agent 34
4.3 ADOCSI SECURITY 35
4.3.1 Unauthorised modification or replacement of Agents 35
4.3.2 Unauthorised modification of information contained in the Agents 36
4.3.3 AMP masquerade 36
4.3.4 Denial of Service 37
4.3.5 Replay attacks 37
4.3.6 Malicious Agents 37
4.4 COMPARATIVE EVALUATION 38
4.4.1 Type of Mechanism 38
4.4.2 Efficiency 40
4.4.3 Security 42

5. FUTURE WORK 45

6. CONCLUSIONS 48

References 50
Appendix 53
Glossary 57

List of Tables

Table 1: CRL compliance with criteria....14
Table 2: Freshest Revocation Information compliance with criteria....15
Table 3: Redirect CRL compliance with criteria....16
Table 4: Enhanced CRL Distribution Options compliance with criteria....17
Table 5: Positive CSI compliance with criteria....18
Table 6: CRS compliance with criteria....20
Table 7: OCSP compliance with criteria....22
Table 8: Freshness-constrained Revocation Authority compliance with criteria....24
Table 9: Evaluation of CSI mechanisms based on the type of mechanism....39
Table 10: Evaluation of CSI mechanisms based on the efficiency of the mechanism....40
Table 11: Evaluation of CSI mechanisms based on the security of the mechanism....43
Table 12: Certificate chains....46
Table 13: Certificate and CRL Extensions....53
Table 14: Enhanced CRL Distribution options....55
Table 15: OCSP referral to CRL....56
Table 16: Certificate identification in OCSP....56
Table 17: Key Usage....56

List of Figures

Figure 1: CSI dissemination....2
Figure 2: ADOCSI infrastructure....31

1. DISSEMINATION OF CERTIFICATE STATUS INFORMATION

1.1 Introduction

Certification Authorities have to provide a "secure and prompt revocation service" [Euro98a], [Euro98b]. The implementation of this service is most often being done with the use of Certificate Revocation Lists (CRL) and online certificate status protocols. These mechanisms could prove to be inadequate in the not so distant future, when the use of certificates will possibly be more widespread. The time granularity problem in CRLs prevents users from having up-to-date information regarding the status of a certificate. Furthermore, the increasing size of CRLs is also going to be a problem, even if Delta-CRLs and CRL distribution points (see sections 2.1 and 3.2.1) are in place. Online certificate status protocols could also prove to be inadequate. This is due to the fact that the number of Certification Authorities (CAs) that emerge, is increasing rapidly. The entities that depend on the use of certificates in order to authenticate certificate holders or verify the signature of the latter on documents, will soon not be able to track all these CAs and have an updated list of Universal Resource Identifiers (URI) [Berne94] which they could use to locate certificate status information.

The aforementioned mechanisms for disseminating certificate status information (CSI) as well as the other ones that have been proposed (see section 2) depend on reliable networks. These mechanisms also take for granted that the software applications which make use of certificates, care enough to check for the status of a certificate before accepting it and that they also support the mechanism used by a specific CA in order to provide entities who depend on certificates with CSI [Fox98].

It is clear that as time goes by, locating up-to-date information regarding the status of a certificate is going to become a tedious task for the aforementioned dependent entities. The dynamic nature of the information that concerns the status of a certificate and the complexity involved with locating CSI and validating a specific certificate is only going to render the aforementioned task more difficult. It is the author's opinion that Public Key Infrastructures (PKI) should provide an easier and more transparent path to the dissemination of certificate status information.

In this dissertation we present the prototype of a mechanism that makes extensive use of Software Agents [Gene94] and of the existing CSI mechanisms in order to provide solutions to some of the problems pertinent to the dissemination of certificate status information.

1.2 Contents of report

This dissertation is divided in the following sections: in section 2 we present a taxonomy of the proposed mechanisms for the dissemination of certificate status information (CSI). In section 3 we develop a set of evaluation criteria for the aforementioned mechanisms and we proceed with evaluating these mechanisms based on these criteria. In section 4 we present an alternative mechanism for disseminating the certificate status information and we provide a comparative evaluation of that mechanism against the other mechanisms that have been proposed. The proposed, alternative mechanism makes use of the properties and functionality of the mechanisms we reviewed in section 2, but also meets certain criteria that those mechanisms do not. In section 5 we refer to the future improvements that can be done on the prototype, alternative CSI mechanism and on the dissemination of certificate status information in general. Finally, in section 6 we present our conclusions.

2. TAXONOMY OF CERTIFICATE STATUS INFORMATION MECHANISMS

In this section we present a taxonomy of the proposed mechanisms for conveying Certificate Status Information (CSI). The mechanisms we present are either de-facto or de-jure standards; others have been proposed by individuals or the industry and they are currently under review. In section 3.1 we develop a set of criteria which will help us evaluate the mechanisms of the following section.

In Figure 1, a generic scheme for CSI dissemination is depicted. Certificate holders send their certificates to entities that depend on the use of those certificate (dependent entities) in order to authenticate the certificate holders (authenticating entities) or verify their signatures (signers) on documents they have sent to the former. Dependent entities have to verify the validity of the certificate they have been presented with. Verification of the validity of the aforementioned certificates is being performed with the use of CSI dissemination mechanisms which are provided by the Certification Authorities that issued the respective certificates. Dependent entities communicate with these CAs and retrieve CSI from their CSI repository, using the CSI dissemination mechanism or mechanisms that the specific CAs support.

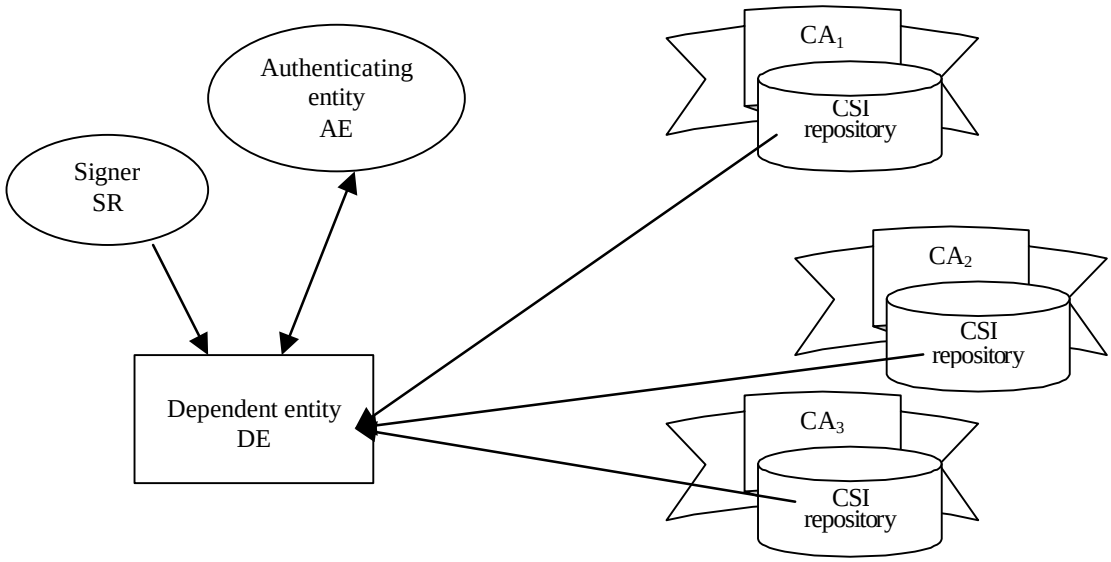

Figure 1: CSI dissemination

The first CSI dissemination mechanism we review is the X.509v2 Certification Revocation List (CRL) [ISO9594].

2.1 Certificate Revocation List

The first mechanism proposed for conveying certificate status information has been the Certificate Revocation List (see Table 13, Appendix) [ISO9594], [Hous99]. According to that, whenever a certificate needs to be revoked by a CA, a timestamped pointer to that certificate is added to a list; this list contains pointers to all the certificates that have previously been revoked by that CA. Conforming to key management standards [ISO11770] the timestamps in X.509v2 CRL contain the date and time when the CA revoked the certificate (revocationDate) or the date and time of known or suspected key compromise (InvalidityDate). This list is signed by the CA itself, in order to let dependent entities verify its authenticity.

The CRLs are issued by CAs periodically, for the dependent entities to know whether they have in their possession the latest revocation information available. However, they tend to be issued infrequently in order to minimise the network resources needed in order to communicate them to the dependent entities, either by pull or push methods. Therefore, the dependent entities are not capable of possessing fresh revocation information on a need-to-know basis. The frequency of CRL issuance is certainly a matter of risk analysis, therefore it is decided in each case, according to the applying conditions.

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 (see Table 13, Appendix). 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) where a specific CRL can be downloaded from; this CRL is the one that will contain the revocation information for that specific certificate, once it is revoked.

The aforementioned CRL partitions can be created on the basis of certificate serial numbers (e.g. a CRL Distribution Point could contain only the revocation information for the certificates with serial number within the domain 510-620) or on the basis of different revocation reasons (e.g. a CRL in a specific Distribution Point could contain only the certificates that were revoked because of key compromise and another Distribution Point could contain only the certificates that were revoked because of a certain change in the affiliation of the certificate holder).

If the X.509v2 CRL extension issuingDistributionPoint contains the indicator indirectCRL [ISO9594], [Hous99] with a Boolean value of True, then the CRL may contain revocation information deriving from CAs other than the CA that issued the CRL. A pointer to the CA from whom each revocation entry in that CRL derives, is stored in the respective certificateIssuer CRL entry extension. The same CA is assumed to be the certificate issuer for all the subsequent CRL entries that do not contain the certificateIssuer CRL entry extension.

Delta-CRLs are there to face 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 will have to retrieve, will contain only those new certificate revocation information. A Delta-CRL must contain a DeltaCRLIndicator (see Table 13, Appendix), so the dependent entities will know which CRL it is that a specific Delta-CRL complements.

We should note that a newer CRL (or Delta-CRL, or a CRL at a specific Distribution Point) does not only contain newer *negative revocation* information. Certificates can also be suspended, by using the certificateHold value in the CRL entry extension CRLReason, and by adding the holdInstructionCode field in that entry as well. The way the holdInstructionCode is to be interpreted by the dependent entities depends on their policy. The CA may decide at a later time to remove a specific certificate from the *on hold* state. In order to do that, the CA will have to remove the entry for that certificate in the CRL If Delta-CRLs are used, the CA will have to add another entry for that certificate in the next Delta-CRL, containing the removeFromCRL reason code in the CRLReason CRL entry extension. Thus, this specific CRL or Delta-CRL will contain *positive revocation* information. The CSI contained in a CRL is, in general, *negative*; that is, the entries in a CRL are used to disseminate the information that a specific certificates has been revoked. However, the CSI contained in a CRL can also be *positive* as in the case we have just presented.

2.2 Fresh Revocation Information

Adams et al [Adams98] propose the use of Delta-CRLs, issued on top of partitioned CRLs (CRL Distribution Points). Their method includes using 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, if it meets the needs of a specific community of dependent entities, then the implementation requires no major changes in the mechanisms already used by the CAs. The crlDistributionPoints certificate extension is a standardised extension [X.509] and the Freshest Revocation Information Pointer is a custom extension that could be added to the certificates issued by the specific CA. However, having that custom extension interpreted properly by the application clients and servers used by the dependent entities remains an issue.

2.3 Redirect CRL

Adams et al [Adams98] also suggest the use of a 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 installing 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 in order to store the CSI (regarding a specific, restricted group of certificate holders) have limited storage. However, as we have mentioned in Fresh Revocation Information as well (section 2.2), the economy of such a solution has to be studied, since certificate-handling software applications should be made aware of that custom extension and the way it should be handled.

2.4 Enhanced CRL Distribution Options

There are two functions that a dependent entity has to use when it looks for CSI: the CSI location function and the CSI validation function. These two functions are usually implemented in one mechanism only; a pointer stored in the certificate is pointing to the CRL where CSI regarding that certificate may appear. Therefore, the dependent entity can locate the CRL from the information stored within the certificate itself and at the same time it can verify that the CSI regarding that certificate is within that CRL. According to Hallam et al [Hall98], there are benefits from separating the location function from the validation function. These include the following:

1. if a location function is not included in the certificate, it can contain last update information regarding the CSI. Therefore, the dependent entities will be able to decide whether to download or not the latest version of the CSI, depending on if they already have it or not,

2. if the validation function is not included in the certificate, the validation function can be used to provide dynamic allocation of the CSI thus allowing for partitioning and

load-balancing of that information according to the needs and requirements presented at a specific time.

The mechanism that had been proposed by Hallam et al [Hall98] for the implementation of the location function, separated from the validation function, is called Status Referrals. StatusReferrals is a CRL extension and it can be used for conveying 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 propose the use of the cRLScope extension (see Table 13, Appendix) 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 (see section 2.3).

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.

2.5 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.

According to Rivest [Rivest98], it should be the dependent entity that should set the freshness requirements of CSI, dynamically and depending on each case where certificates are used. Furthermore, it must be the certificate holder (either an authenticating entity or a signer) that has to provide the dependent entity with the CSI requested, in order to alleviate the burden of locating and validating the CSI from the dependent entity who is probably already "busy", trying to allocate resources and serve requests for services from authenticating entities.

The need for the use of such mechanisms does not exist if the certificate issuing authority is able to issue short-lived certificates, rendering thus useless the CSI mechanisms, since the CSI contained in the certificate itself will always be fresh. However, this is seldom the case with certification authorities, which usually are forced to issue long-lived certificates mainly due to lack of resources which would render them capable of frequently re-issuing certificates. In the case of a usual long-lived certificate, the dependent entity has to retrieve, or even locate, by itself the necessary CSI in order to verify the validity of the certificate of an authenticating entity or a signer. Rivest suggests that there can be a period within which a dependent entity does not need to have CSI in order to verify the validity of a certificate. This can be done if the life-cycle of a (non-compromised) certificate includes three phases ("guaranteed", "probable" and "expired") instead of just two ("probable" and "expired"). CAs could include statements inside the certificate that mark those three phases. In such a case, a dependent entity may not need to have CSI in order to authenticate a certificate holder if the certificate is within the "guaranteed" phase.

However, checking whether a certificate has been compromised (and therefore revoked) requires the communication of CSI from the authenticating entity to the dependent entity. 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 that 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.

In this case, when a dependent entity wishes for CSI 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 the there are more than one entities that hold that certificate and make, 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.

2.6 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. 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 has been defined already), 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 $K=Fi\text{-}j(Y_0)$, 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. $F_j(K)=Y$ (1)
2. $F(K)=N$ (2)

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) did not apply.

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.

2.7 Online Certificate Status Protocol

The Online Certificate Status Protocol (OCSP) [Myer99] is a protocol proposed by the IETF Network Working Group that allows the dependent entities to query for CSI in a more timely fashion than CRLs. OCSP can be used in order to provide timely CSI, and it could be used in conjunction with CRLs. It does provide an extension (see Table 15, Appendix) 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, or another entity that is designated by the CA as a CSI provider (Trusted Responder or CA Designated Responder). These entities must possess a specially marked certificate, issued by the CA, which authorises them to provide CSI to requestors. 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 CSI responses given by the aforementioned authorities are always signed in order to let requesters verify the authenticity of the CSI. However, the signing of each OCSP response 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 authority that provides the OCSP service open to replay attacks, where someone could replay OCSP responses before their expiration date but after a certificate has been revoked.

Furthermore, the requests for OCSP service can be signed by the requesters themselves. This is a useful feature, because entities that offer the OCSP service could have requesters authenticate before they deliver the CSI they request. One of the uses for such an authentication could be to allow entities that offer OCSP service to charge for it. It is considered as a natural consequence to charge for CSI, if PKI is finally used extensively in the commercial world [Fox98].

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 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.

OCSP responses can contain three values concerning time:

1. thisUpdate, which indicates the time at which the CSI communicated to the requester was known to be correct,

2. nextUpdate, which indicates when the next update of CSI is expected to be available,

3. producedAt, which indicates the time at which the CSI communicated to the requester was signed by the entity that runs the OCSP service.

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. In that case, the CA should include the 'nocheck' extension in the OCSP service provider certificate in order to notify requesters that they should not check for the revocation status of that certificate, as there will not be any revocation information at all, because of its short lifetime,

2. The CA may choose to specify an extension in the certificate the OCSP service provider, that points to a CRL or other CSI-providing service point, in order to facilitate the requester locate the CSI regarding the OCSP service provider certificate,

3. The CA could choose not to specify any method for checking the CSI of the OCSP service provider, therefore the requesters will have to locate the relevant CSI by themselves, if they decide they need to locate such information.

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 (see Table 16, Appendix) 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 no chance of two sets of identification information (the hashes we have mentioned above) to collide, if the hash function is collision-free. Hashes are used for another reason as well. In certain cases, certificates may contain private information and in some countries the legislation concerning private information could prohibit the unrestricted distribution of such information without any specific security measures. 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.

2.8 Freshness-constrained Revocation Authority

Stubblebin [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), who 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, the dependent entity will check the CSI freshness requirements imposed by the issuing CA, the higher-level CAs and the high-level PCA. These requirements apply to the certificate of the certificate holder in question. In addition, the dependent entity will impose its own freshness requirements and will expect from the certificate holder to provide a short-lived certificate, issued by the RevA, that fulfils 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.

It is obvious that this method allows for flexible balancing of the authentication costs and level of protection on a per transaction basis. Furthermore, the CSI, that is the short-lived, timestamped certificate, need not be communicated from a trusted repository. The RevA can deploy agents that will 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.

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.

3. EVALUATION OF CERTIFICATE STATUS INFORMATION MECHANISMS

In this chapter we evaluate the available mechanisms for certificate revocation, and for disseminating the certificate status information (CSI) to users. We will first develop the criteria for the aforementioned evaluation and then proceed with the evaluation itself. This evaluation will provide the information we need in order to form an alternative method for disseminating the certificate status information. This method, hopefully, will be able to deal with the problems the other methods face.

3.1 Evaluation Criteria

The criteria we will use in order to evaluate the revocation mechanisms are the following:

3.1.1 Type of Mechanism

M1. *Transparency*. Transparency in locating the CSI repository (CSI location function) and verifying that the CSI contained in that repository is the one the dependent entity is looking for (CSI validation function),

M2. *Offline revocation*. Whether a user can revoke his certificate by himself without having to contact the respective CA,

M3. *Delegation of revocation.* Delegation of the revocation by the CA to another authority, either another CA or another authority that operates only as a Revocation Authority (RevA) and not as a Certification Authority,

M4. *Delegation of the CSI dissemination.* Delegation of the CSI dissemination by the CA to another authority; the latter may be trusted by the dependent entities or not, depending on the mechanics of the CSI dissemination in each case,

M5. *Delegation of the certificate path validation.* Delegation of the certificate path validation from the dependent entity to another entity; the dependent entity should be provided with the means to verify the origin of the validation result; in addition to that, the entity that performs the certificate path validation should be trusted by the dependent entity,

M6. *Referral capability*. If the CSI repository does not contain the CSI the dependent entity is looking for, the repository could refer the dependent entity to another CSI repository that may contain the aforementioned CSI,

M7. *Revocation Reasons*. The certificate path validation function should take into consideration the reasons for the revocation of a certificate (e.g. reasonCode in X.509v2 CRLs, see Table 13) in order to assert on the validity of a certificate in a certificate path (see section 5).

3.1.2 *Efficiency*

E1. *Timeliness of CSI*. Dependent entities should be able to locate and receive CSI on time, for them to use such information in authenticating entities or verifying the signatures of entities,

E2. *Freshness of CSI*. How 'fresh' is the status information delivered to dependent entities,

E3. *Bounded revocation*. New CSI should become available to the dependent entities within a bounded time period,

E4. *Emergency CSI capability*. Facility to generate emergency CSI to the dependent entities (e.g. in case the certificate of an entity that many people use and trust has been revoked),

E5. *Economy*. Economy should be examined both from the point of view of the authorities that control the use of certificates (e.g. Certification Authorities, Registration Authorities, Revocation Authorities) and from the point of view of the entities that use certificates (e.g. dependent entities, certificate holders). The CSI dissemination mechanism should not cause any obstacles, delays or disruptions in the normal working practice that the dependent entities and certificate holders follow. The CSI dissemination mechanism should not require from the dependent entities or certificate holders to have a clear and profound understanding of the mechanism itself, and interact to a great extent with the mechanism in order to make it operate properly. Finally, the CSI dissemination mechanism should not hinder the operation of the other functions performed by the authorities that control the use of certificates; these authorities should not have to change the way they issue certificates, register new entities or revoke certificates in order to accommodate for the operational needs of the CSI dissemination mechanism.

E6. *Scalability*. When the number of the CSI dissemination mechanism authorities and users (e.g. CAs, RevAs, dependent entities, certificate holders) increases, new problems or difficulties in the operation of the mechanism should not emerge. Scalability relates more to resources, in contrast to economy which relates more to infrastructure and mode of operation,

E7. *Adjustability*. The dependent entities (or the CA, RevA as well) should be able to adjust the location or validation function operation in order to create a balance between performance and protection, depending on the requirements and the risk policy in each case. Ideally, the dependent entity should be able to do this, since it is the dependent entity that takes the risk.

3.1.3 *Security*

S1. *CSI disseminator authentication*. The dependent entities must verify the origin of the CSI they receive. The CSI disseminator is usually the RevA (the service is operated by the CA) itself. If authentication is not used, a malicious entity pretending to be a trusted CSI dissemination entity, could provide the dependent entities with false CSI that appears to be valid,

S2. *CSI integrity*. Verifying the integrity of the CSI, when it is stored in the CSI repository, transferred to the dependent entities and when it is stored in the dependent entities' local repository. Such verification must be possible lest a malicious entity modifies the CSI in transit, before the dependent entity receives it; should this happen, the dependent entity may not realise that the received CSI is old, partial or invalid in any way,

S3. *CA compromise*. The effects of a CA key being compromised should be minimised,

S4. *RevA compromise*. There should be a mechanism for the dependent entities to know whether the RevA has been compromised. This mechanism must not be the same with the one used by the dependent entities in order to receive CSI on certificates that belong to entities other than the RevA,

S5. *Contained functionality*. If RevA is compromised, it should not be possible for the entities that gained control of the RevA to issue new certificates,

S6. *Availability*. The CSI dissemination mechanism has to be resilient against unreliable networks.

3.2 Evaluation of CSI dissemination mechanisms

In this section we evaluate the mechanisms presented in section 2, using the criteria developed in section 3.1. Throughout the evaluation, we identify the problems each mechanism faces. In section 4.4 we present a comparative evaluation of the aforementioned mechanisms and the one we propose in section 4. One of our main concerns with the CSI dissemination mechanisms is the fact that they are not transparent as far as the user is concerned. That is, both the dependent entity and the application it is using, should care enough to go through the necessary steps in order to receive the appropriate CSI. Another concern is that, in most of the mechanisms presented, the dependent entity should have a good understanding of the revocation and CSI dissemination mechanism in order to interact with the mechanism and aid in the operation of the mechanism.

We will begin our evaluation with the X.509v2 CRLs, which were presented in section 2.1.

3.2.1 Certificate Revocation List

The first set of criteria we evaluate CRLs against are the "Type of Mechanism" criteria, presented in section 3.1.1. CRLs do not meet criterion M1; the dependent entity has to locate and retrieve on its own the latest CRL that may have CSI on the certificate that the entity has been presented with, by an authenticating entity or signer. CRLs do not meet criterion M2 either, since a certificate holder has to contact the CA (or another entity that has revocation authority over the user's certificate) in order to have his certificate revoked.

CRLs meet some of the *delegation* criteria. Criterion M3 is not met, as CRLs have to be signed by the private CA key. A level of indirection could be implemented here; the CA could use a different key for signing the certificates and a different key (also owned by the CA) in order to sign the CRL. However, that would add complexity to the mechanism, since the dependent entities would have to have a trusted copy of that key as well (if the CRL signing key has not been signed with the certificate signing key), and monitor the validity of that key through means other than the CRL, since if that key was compromised one would be able to distribute false CRLs. The only reason for having that level of indirection would be to minimise the "exposure" [Denni82] of the private CA key that is used to sign certificates. However that is not enough of an advantage in order to make CAs adopt this solution. Criterion M4 is met; CAs can delegate the CSI dissemination to other entities. Since the CRL is signed by the CA, there is no threat towards the integrity of the CRL; however, dependent entities should trust the entities they choose to retrieve the CRL from. If the CRL distribution entities are not trusted, they might not give to the dependent entities the latest CRL, thus hiding useful CSI from them. In order to deal with untrusted CRL distribution entities or trusted CRL distribution entities acting maliciously, CAs should include in the CRLs the date of the next release of the CRL, thus providing bounded revocation (criterion E3). If this happens, CRLs could be distributed even by untrusted parties since, the dependent entities will know when the next CRL is due, therefore the untrusted parties will not be able to deny that a new CRL has been issued.

Delegation of the certificate path validation is not supported in CRLs (criterion M5); the dependent entities have to download the CRL and validate a certificate path themselves, based on the information contained in the CRL. CRLs do have referral capability (criterion M6); CAs can use the 'Distribution Points' and 'Delta-CRL' (see section 2.1) in order to refer the CSI requester to another point of distribution for the CSI he is looking for (distribution points) or to fresher CSI (Delta-CRLs).

CAs could issue new CRLs very frequently, thus meeting the freshness criterion E2. Unfortunately, the cost for the dependent entities of downloading frequently the new CRLs is considerable; Distribution Points and Delta-CRLs can help in reducing that cost, though. However, issuing a CRL very frequently renders the CRL mechanism obsolete, since in that case it tends to offer the same functionality as an online certificate status protocol (like OCSP, see section 2.7) but at the cost of downloading much more information and placing the certificate path validation burden on the user. Therefore, having CRLs issued very frequently is not being done in practice, so CRLs do not meet criterion E2. There will probably be some certificates the dependent entity will not know they have been revoked, when using CRLs as a CSI mechanism, since revocation may have occurred within the CRL validity period, before the next CRL update time. Therefore CRLs do not meet criterion E1, about timeliness, as well.

Emergency CSI (criterion E4) can be issued, with the use of CRLs. All the CA has to do is issue a new CRL (or Delta-CRL) that contains the new CSI that have to be communicated urgently to the dependent entities. However, the dependent entities have no means of being informed that a new CRL has been issued before the expected time, except for standard means of "push" communication (e.g. e-mail, post etc). This certainly cannot be done, mainly because the CA does not know who the dependent entities may be. In conclusion, when using CRLs, there is no way Emergency CSI can be communicated, therefore CRLs do not meet criterion E4.

Even though the CRL scheme supports the use of revocation reasons (criterion M7) to be included in the CRL, no method has been standardised for taking those reasons into consideration for the certificate path validation (see section 5).

CRLs do not meet criterion E5 about economy. Their use by dependent entities requires that these entities take the time to locate and retrieve the latest CRLs, validate the certificate paths that point to the certificate which they want to verify. The dependent entities should also have a clear understanding of the use of CRLs; they must be able to locate all the certificates and CRLs they may need in order to gather the necessary CSI, which they must input to the certificate path validation function. Furthermore, they must be able to understand the mechanics of the certificate path validation function and evaluate the result of that function. Furthermore, if the availability of CRLs becomes an issue, the CAs will have to replicate the CRLs to repositories other than the CA repository (Directory). These repositories could possibly be untrusted ones, therefore the CAs might feel the need to verify regularly that the CRL is still being made available from a certain, untrusted repository where they have installed it. For all those reasons, the criterion about economy (E5) is not met.

Moreover, the scalability criterion (E6) is not met either, since the cost of communicating a large CRL to a large number of users can become quite significant, depending on the number of dependent entities that download the CRL and the number of revoked certificates. Furthermore, certain dependent entities might have self-imposed restrictions on the storage space they can use for CRLs (e.g. if smart cards are used as a CRL repository in a specific application).

Dependent entities cannot adjust the location or validation function (criterion E7) in order to match the risk policy in each case; dependent entities cannot control as well which CRL (or parts of) they download, since the only information the dependent entities have is a pointer to a specific CRL (or Distribution Point) contained in the certificate they wish to verify. Furthermore, certificate path validation based on CSI retrieved from CRLs is not a flexible or adjustable procedure. If the revocation reason codes were used then some adjustability could be

introduced, but considering those reasons in a certificate path validation still presents problems (see section 5).

The role of the CSI disseminator, in the case of CRLs, is usually undertaken by the CA itself. It could also be a locally trusted authority (such as the Network Operations Centre in an organisation, providing cached copies of the CRLs that have been recently retrieved from entities within the organisation). Authentication of the origin of the CRL is possible through the signature that the CA has put on the CRL, therefore criterion S1 is met. For the same reason, the integrity of the CRL can be verified, therefore criterion S2 is met also.

If a CA is compromised, the respective CRLs issued by that CA should be considered void; all certificates issued by that CA should be considered as revoked, because the entities that hold the private CA key will be able to issue false certificates at will. Therefore, the effect of having a CA key compromised is maximum; any entity that compromises the CA private signing key can issue and revoke certificates at will, therefore CRLs do not meet criterion S3. The role of the Revocation Authority (RevA) is undertaken by the CA itself; however, the RevA (CA) may use a different private key for signing CRLs. In this case, the respective certificate must contain only the crlSign bit in the KeyUsage extension (see Table 17, Appendix). In this case, compromise of the RevA key does not enable the holder of the compromised key to issue new certificates. Since criterion S5 is met only in the case that the key used to sign the CRL is different to the one used to sign certificates, we assert that criterion S5 is partially met. Furthermore, in this case, the dependent entities should always check the validity of the RevA certificate, whenever they receive new CSI about other certificates. However, since there is no definite mechanism for verifying the RevA key validity, we shall have to assert that CRLs only partially meet criterion S4.

Criteria	Compliance
M1. Transparency	
M2. Offline revocation	
M3. Delegation of revocation	
M4. Delegation of CSI dissemination	√
M5. Delegation of the certificate path validation	
M6. Referral Capability	√
M7. Revocation reasons	
E1. Timeliness of CSI	
E2. Freshness of CSI	
E3. Bounded revocation	√
E4. Emergency CSI capability	
E5. Economy	
E6. Scalability	
E7. Adjustability	
S1. CSI disseminator authentication	√
S2. CSI integrity	√
S3. CA compromise	
S4. RevA compromise	√?
S5. Contained functionality	√?
S6. Availability	√

Table 1: CRL compliance with criteria

We should stress the fact that if CAs use the same private signing key for signing new certificates and signing the CRLs, then criteria S3, S4 and S5 are definitely not met; the dependent entities will not be able to know that that specific key has been compromised through

the standard CSI channels, the entities that compromised the CA/RevA private signing key can lead the dependent entities into believing false CSI to be true. The information that the CA private signing key has been compromised will have to reach the dependent entities through other channels, probably out-of-band ones. This could be the case if the CA is the root in a certification path, or if it does not belong at all in a certification path; in both cases the CA certificate is self-signed, therefore there is no one who can provide CSI for that certificate, at least through the standard CSI channels and mechanisms that CRLs provide.

CRLs do provide the means to meet with criterion S6; Replication of CRLs in many repositories can help increase the resilience of the mechanism against unreliable networks. We have to note though that, in case the entities that replicate the CRLs are not trusted, they could hide certain CRLs from entities that ask for them, thus delivering partial and therefore false CSI. This can be avoided by having bounded revocation (criterion E3), and using the extension nextUpdate in the CRL, in order to ensure that dependent entities are aware of that bounded revocation.

3.2.2 Fresh Revocation Information

This mechanism meets all the other criteria met by the X.509v2 CRL. In addition to that, it also meets the freshness criterion E2. This is achieved with the use of the Freshest Revocation Pointer. If the Freshest CRL repository has a high level of availability (criterion S6) then this mechanism also meets the timeliness criterion (E1). Furthermore, the Emergency CSI criterion (E4) is met, since the Freshest CRL update period is very short and not fixed.

Criteria	Compliance
M1. Transparency	
M2. Offline revocation	
M3. Delegation of revocation	
M4. Delegation of CSI dissemination	√
M5. Delegation of the certificate path validation	
M6. Referral Capability	√
M7. Revocation reasons	
E1. Timeliness of CSI	√
E2. Freshness of CSI	√
E3. Bounded revocation	√
E4. Emergency CSI capability	√
E5. Economy	
E6. Scalability	
E7. Adjustability	√
S1. CSI disseminator authentication	√
S2. CSI integrity	√
S3. CA compromise	
S4. RevA compromise	√?
S5. Contained functionality	√?
S6. Availability	√

Table 2: Freshest Revocation Information compliance with criteria

Moreover, this mechanism meets the adjustability (E7) criterion, since dependent entities can choose whether they need fresher CSI compared to what they get from the original CRL and if they do, the can download the Freshest CRL as well. The mechanism can be adjusted by the dependent entities on a per-transaction level. However, the level of adjustability is low

(dependent entities can only choose whether they will download the regular CRL or the Freshest one as well).

3.2.3 Redirect CRL

This mechanism meets at least the same criteria with X.509v2 CRLs. Moreover, the mechanism is more scalable (criterion E6) to X.509v2 CRLs. Redirect CRLs (RCRL) provide dynamic re-partitioning of the revocation space, thus they can be used in order to re-fragment the CRLs or Distribution Point CRLs whenever needed, without the dependent entity having to perform any additional actions in order to retrieve the CSI it is looking for. This fact allows for the maintenance of a CRL structure consisting of a 'central' CRL which contains Redirect Pointers to other CRLs; each CRL contains CSI for a user space specified in the 'central' CRL. This mapping can change over time; the dependent entity, once it visits the 'central' CRL, will be redirected to the CRL that contains the information it is looking for, without him having to perform any further actions. Therefore, the amount of information a dependent entity has to retrieve (and the respective communication costs) in order to retrieve the CSI it is looking for, can be low. The reduction of the communication and storage costs renders this mechanism scalable, at least more than standard CRLs.

Criteria	Compliance
M1. Transparency	
M2. Offline revocation	
M3. Delegation of revocation	
M4. Delegation of CSI dissemination	√
M5. Delegation of the certificate path validation	
M6. Referral Capability	√
M7. Revocation reasons	
E1. Timeliness of CSI	
E2. Freshness of CSI	
E3. Bounded revocation	√
E4. Emergency CSI capability	
E5. Economy	
E6. Scalability	√
E7. Adjustability	
S1. CSI disseminator authentication	√
S2. CSI integrity	√
S3. CA compromise	
S4. RevA compromise	√?
S5. Contained functionality	√?
S6. Availability	√

Table 3: Redirect CRL compliance with criteria

3.2.4 Enhanced CRL Distribution Options

This mechanism has the same properties with a RCRL (see section 2.3), therefore it meets the same criteria.

Criteria	Compliance
M1. Transparency	
M2. Offline revocation	

M3. Delegation of revocation	
M4. Delegation of CSI dissemination	√
M5. Delegation of the certificate path validation	
M6. Referral Capability	√
M7. Revocation reasons	
E1. Timeliness of CSI	
E2. Freshness of CSI	
E3. Bounded revocation	√
E4. Emergency CSI capability	
E5. Economy	
E6. Scalability	√
E7. Adjustability	
S1. CSI disseminator authentication	√
S2. CSI integrity	√
S3. CA compromise	
S4. RevA compromise	√?
S5. Contained functionality	√?
S6. Availability	√

Table 4: Enhanced CRL Distribution Options compliance with criteria

3.2.5 Positive CSI

This mechanism [Rivest98] offers some transparency in the CSI location procedure for the dependent entities but that is because it is the authenticating entities that have to locate CSI that meets the requirements set by the dependent entities and sent it to them. Therefore, the transparency criterion (M1) is not really met.

The offline revocation criterion (M2) is met, since the certificate holder can revoke his certificate without contacting the CA or any other entity. However, since the CSI dissemination is being done by the Suicide Bureaus (SB), the certificate holder has to hand in his suicide note to the SBs so the CSI can be communicated to the dependent entities. Delegation of revocation is certainly met (criterion M3), since it is the certificate holder, not the CA or RevA that revokes the certificate, even if the certificate holder has delivered a suicide note to the RevA/CA upon receiving his certificate. Furthermore, an entity that provides services could revoke the certificate of the authenticating entity, if the latter has already provided the former with a "suicide note" (see section 2.5).

Delegation of certificate path validation (criterion M5) is also true; SBs issue "certificates of good health", depending on the suicide notes they possess, which they use in order to validate a specific certificate. It is the SBs that validate a certificate path and not the dependent entities, and it is the authenticating entities that have to locate the CSI, retrieve it from the SBs and communicate it directly to the dependent entities (criterion M4). SBs have to form a network in order to provide a common pool of CSI, thus they have to provide referral capabilities in searching their repositories; therefore, they meet criterion M6. However, they do not meet criterion M7, since they do not take into account the reasons for the revocation of a certificate.

If the validity period of the short-lived "certificates of good health" is very short, then CSI is communicated timely to the dependent entities. Suicide Bureaus can revoke a certificate (stop issuing "certificates of good health" for that certificate) as soon as they realise or as soon as they are notified that there is a reason to revoke that certificate. Moreover, dependent entities always get fresh (criterion E2) CSI from the SBs. Therefore, this mechanism meets the

timeliness criterion (E1). However, if the validity period of the certificates is not very short then this mechanism meets only partially the timeliness criterion. If a dependent entity decides to verify a certificate, based on a "certificate of health" it already has and which is not so fresh, then it may consider a revoked certificate to be still valid.

Bounded revocation (criterion E3) is not guaranteed, since suicide notes can be presented to the SBs at any time, coming either from the certificate holders or from entities to whom the certificate holders had given a suicide note (see section 2.5). However, for the same reason, the emergency CSI criterion (E4) is supported.

This mechanism is economic (criterion E5). The dependent entities do not have to process any other information than certificates in order to verify the validity of a certificate. Therefore, the infrastructure required in order to implement and operate this CSI dissemination mechanism is minimal. Moreover, the mechanism is simple to understand, both by the dependent entities and the certificate holders (authenticating entities or signers). The dependent entities ask the certificate holders for fresher certificates and the latter can provide that CSI simply by following a procedure that is approximately the same as the procedure they followed in order to get their regular, long-lived certificate. Finally, the authorities involved in this mechanism (CAs, SBs) do not have to implement another infrastructure or complement the existing one, in order to support this mechanism. The only process the aforementioned authorities have to go through, in order to provide CSI, is to issue certificates, again.

Criteria	Compliance
M1. Transparency	
M2. Offline revocation	√
M3. Delegation of revocation	√
M4. Delegation of CSI dissemination	√
M5. Delegation of the certificate path validation	√
M6. Referral Capability	√
M7. Revocation reasons	
E1. Timeliness of CSI	√?
E2. Freshness of CSI	√
E3. Bounded revocation	
E4. Emergency CSI capability	√
E5. Economy	√
E6. Scalability	
E7. Adjustability	
S1. CSI disseminator authentication	√?
S2. CSI integrity	√?
S3. CA compromise	√?
S4. RevA compromise	√?
S5. Contained functionality	√?
S6. Availability	√

Table 5: Positive CSI compliance with criteria

Scalability might be an issue that needs to be further studied in this mechanism. Rivest [Rivest98] assumes that it is possible for the SBs to share a "high-speed reliable network". If this mechanism is used for CSI dissemination concerning certificates of entities that belong to a small, closed user group then it could be possible to ensure network reliability to a great extent. However, if this mechanism is applied in a wide scale, then problems may arise from the lack of network reliability. Replication of "suicide notes" might not be able to take place as fast as

this mechanism assumes it will. If this happens, then some SBs will not be aware of revocations that other SBs will be aware of, therefore some SBs will not be able to provide the dependent entities with CSI in a timely manner.

This mechanism is not economic (criterion E5), since it requires an infrastructure that is not compatible with the one already deployed and used; furthermore, it requires the high availability of the SBs. However, this mechanism can be quite scalable once the infrastructure is there since the CSI communication are very small and the storage costs (at least at the dependent entity side) are minimal. Finally, the mechanism is highly adjustable since the dependent entity can set requirements regarding the freshness of the CSI the authenticating entity has to locate and deliver to the former.

This mechanism meets the adjustability criterion (E7) because dependent entities can set freshness requirements for the CSI they require from the authenticating entities.

No clear method has been defined in [Rivest98] for verifying the origin or the integrity of "certificates of good health". We assume that SBs also have certificates, thus the origin and integrity of "certificates of good health" can be verified through the digital signature of the SBs on the aforementioned certificates. Though, since no method has been clearly defined in [Rivest98], we assert that the CSI mechanism presented in [Rivest98] meets partially criteria S1, S2. The same applies for criterion S4; no method has been defined in order to verify the validity of the certificate of an SB. We assume that dependent entities can verify that by contacting directly the issuing CA, which will provide the dependent entities with a way of verifying the validity of SB certificates, other than through a Suicide Bureau. If a CA key is compromised then the entity that holds the compromised CA key will only be able to issue new certificates and not revoke old ones. The only exception would be if the CA performs key recovery as well and if the aforementioned entity managed to compromise the key recovery repository of the CA as well. If this is the case then old certificates can be revoked as well, therefore criterion S3 is partially met.

Criterion S5 is partially met because an entity that has compromised the RevA (SB) private key cannot issue new certificates. However, that entity can revoke valid certificates, by simply not issuing "certificates of good health" for those certificates. In addition to that, the aforementioned entity can issue "certificates of good health" for certificates that had been revoked, thus making them valid again. Criterion S6 is met, since referrals and replication can be used in order to ensure that the mechanism for disseminating CSI has a high level of availability.

3.2.6 Certificate Revocation Status

This mechanism is not transparent (criterion M1) since the user has to locate, download and validate the CSI, which is composed of the value K and the signed list of non-expired certificates. The mechanism also does not support offline revocation (criterion M2); it is the CA that revokes certificates, by issuing the N_0 number instead of another value (Fi-j(Y_0)). Revocation and dissemination of the CSI could be partially delegated (criteria M3, M4); an entity other than the CA could hold the values N0,Y0 for all the certificates and publish the value K at regular time intervals, while the CA should publish the updated list of non expired certificates with the same update granularity.

If the Y,N values are updated at regular intervals (e.g. per day) then this mechanism also supports bounded revocation (criterion E3), therefore a malicious entity that disseminates the CSI on behalf of the CA could not deny the existence of fresh information at a certain point of time. Fresh CSI (criterion E2) can be updated and retrieved by the dependent entities very often because the involved CSI communication and storage costs are very small, compared with other mechanisms we have already examined (criterion E6).

No functions are described [Micali96] for locating the CSI, therefore the mechanism -at its present state- does not ensure timely delivery of CSI (criterion E1). The is no support for

referral capability (criterion M6). The revocation reasons are not taken into consideration for the validation of CSI, therefore criterion M7 is not met. The certificate path validation function has to be performed by the dependent entity, after having received the Y,N values and, if needed, the signed list of no-yet-expired certificates; thus, neither criterion M5 is not met.

There is no emergency CSI capability (E4) since the CSI update granularity is determined at the generation of a certificate (see section 2.6). However, CSI update granularity should be the same for all certificates issued by one CA in order to avoid inconsistencies and having dependent entities confused.

This mechanism is not economic (criterion E5). The mechanism requires from users to locate and retrieve both the K value for every certificate they wish to verify and -if needed- the updated list that contains the issued but not yet expired certificates. Furthermore, the mechanism requires from the dependent entities a clear understanding of the mechanics of CSI dissemination; dependent entities should be able to decide on the reason why a K value that has been given to them was not equal either to Fj(Y) or F(N) (see section 2.6); in this case, the dependent entities will have to judge for themselves whether a certificate is valid or not, depending on whether there is an entry for that certificate in the list of issued and not yet expired certificates. CSI validation is not definite, at least not as much as in other mechanisms we have examined (e.g. CRLs) and human interaction may sometimes be needed to reach a conclusion on the validity of a certificate.

Criteria	Compliance
M1. Transparency	
M2. Offline revocation	
M3. Delegation of revocation	√?
M4. Delegation of CSI dissemination	√
M5. Delegation of the certificate path validation	
M6. Referral Capability	
M7. Revocation reasons	
E1. Timeliness of CSI	
E2. Freshness of CSI	√
E3. Bounded revocation	√
E4. Emergency CSI capability	
E5. Economy	
E6. Scalability	√
E7. Adjustability	
S1. CSI disseminator authentication	
S2. CSI integrity	√
S3. CA compromise	√
S4. RevA compromise	
S5. Contained functionality	√
S6. Availability	√

Table 6: CRS compliance with criteria

The mechanism is not adjustable (criterion E7); there is a predetermined set of information the dependent entities must retrieve in order to assert on the status of a certificate (value K and possibly signed list of non expired certificates). There is also no defined mechanism for authenticating the origin of part of the CSI (value K), therefore criterion S1 is not met, although the origin and the integrity of the list of not-yet-expired certificates can be verified through the CA signature on the list. There is though an indirect way of verifying the integrity

of the value K (criterion S2). One of the values Y, N stored in the certificate should be equal to either $F_j(K)$ or F(K) (see section 2.6); if they are not, they have been modified by unauthorised entities on purpose or by accident.

If the CA private key is compromised by a malicious entity M, then M could issue a list of issued and not-yet-expired certificates that does not contain certain certificates that have been issued and have not yet expired. However, if the generation and dissemination of the K values is being performed by a RevA R other than the CA, or if M did not also compromise the Y0, N0 values repository, the dependent entities could still get valid CSI regarding the certificates that had been issued before the CA key compromise. A CA key compromise by itself enables the entity that holds the compromised CA key to issue new certificates but not revoke old ones, therefore criterion S3 is met.

However, if M manages to launch a successful DoS attack at R, then R would not be able to provide a dependent entity with the requested K value for a certificate C. In this case the dependent entity would ask for the list of issued and not-yet-expired certificates. If M could lead the dependent entity in retrieving the list (signed with the compromised CA key) from his repository, then the dependent entity would be led into believing that the certificate C has never been issued by the CA (and that it was probably issued by some entity that compromised the CA key), therefore it is not valid. Therefore, if RevA is not available, an entity that compromised the CA key could lead dependent entities into believing that valid certificates are invalid.

If RevA is compromised (in case it is not the CA itself) then the entity M that compromised RevA could revoke old certificates but would not be able to issue new ones because that would require knowledge of the CA private key. Therefore, criterion S5 about contained functionality is met. However, no specific mechanism is described in [Micali96] for dependent entities to verify the that RevA has not been compromised, therefore criterion S4 is not met.

Finally, the availability of this mechanism could be adjusted to the necessary levels by replicating the frequently updated K values for certificates and signed lists of issued and not-yet-expired certificates in hosts other than the central CSI disseminator as well. These could be hosts that are locally trusted by a group of users.

3.2.7 Online Certificate Status Protocol

This mechanism does not support transparency (criterion M1), since the dependent entity must locate and retrieve CSI itself. The user cannot revoke his own certificate, therefore criterion M2 is not met either. However, criteria M3 and M4 are met, because both the revocation and the dissemination of CSI can be performed by a separate RevA (Trusted Responder or CA Designated Responder) as long as this authority has a certificate from the CA marked with the appropriate extension (see section 2.7).

Certificate path validation is being performed by the OCSP service provider, therefore criterion M5 not met. Referrals are supported (criterion M6); an OCSP service provider may contain only serviceLocator extensions [Myer99] that point to other OCSP service providers that contain the CSI that an entity requested from the former. The reasons for the revocation of a certificate are not taken into account for the certificate path validation function, therefore criterion M7 is not met.

If the OCSP service is being provided by the CA itself or if the OCSP provider has a direct, online link to the CA certificates and CSI repository, then the dependent entities will be able to receive information in a timely fashion. However, the OCSP service provider is not required to have all the time up-to-date CSI; that is, it may update the CSI it has, by regularly downloading all the latest CSI available at the CA CSI repository. Therefore timeliness is not always feasible. However, CSI delivered by an OCSP tends to be very fresh (criterion E2), because even in the case the OCSP service communicates offline with the CA CSI repository in order to make batch updates of his own CSI repository, this update is very frequent, therefore

the CSI given by the OCSP service provider is very fresh. This mechanism supports bounded revocation (criterion E3). If the OCSP provider has an online link with the CA CSI repository, then CSI is immediately available to the dependent entity; if that communication is offline, then the boundary is equal to the update granularity. Emergency CSI is supported (E4); a CA may inform the OCSP service provider before a regular update (if there is no online link to the CA CSI repository) about certificates for which the CSI needs to be available immediately to the dependent entities.

This mechanism supports delegation of the certificate path validation, therefore depending entities do not have to possess any knowledge and infrastructure in order to validate a certificate path. This is an economic feature of the OCSP CSI dissemination mechanism. However, this mechanism gives CSI on a per-certificate basis. The advantage of this feature is that dependent entities do not have to retrieve large amounts of CSI in order to verify the validity of a certificate, as is the case e.g. in CRLs. Therefore, the communication, processing and storage costs for the dependent entities are small, which makes this mechanism scalable (criterion E6). However, the mechanism requires that dependent entities have to be connected to the network whenever they wish to retrieve CSI about a specific certificate; continuous network access is not always feasible, either due to the connection costs involved or due to the lack of resources. Moreover, CAs have to issue certificates for the RevAs (OCSP service providers) as well, and dependent entities must verify those certificates by means other than the OCSP CSI

queries. Dependent entities -in this case- have to be motivated and knowledgeable enough to verify frequently (if not every time they ask for CSI) the validity of the RevA certificate. In conclusion, the OCSP mechanism does not meet the economy criterion (E5).

Criteria	Compliance
M1. Transparency	
M2. Offline revocation	
M3. Delegation of revocation	√
M4. Delegation of CSI dissemination	√
M5. Delegation of the certificate path validation	√
M6. Referral Capability	√
M7. Revocation reasons	
E1. Timeliness of CSI	√?
E2. Freshness of CSI	√
E3. Bounded revocation	√
E4. Emergency CSI capability	√
E5. Economy	
E6. Scalability	√
E7. Adjustability	
S1. CSI disseminator authentication	√
S2. CSI integrity	√
S3. CA compromise	
S4. RevA compromise	√
S5. Contained functionality	√
S6. Availability	√?

Table 7: OCSP compliance with criteria

This mechanism is not adjustable (E7); the dependent entities retrieve the only CSI that is available from the OCSP service provider. The dependent entities may not adjust any of the

parameters of the CSI location or validation function, and may not request CSI that meets certain requirements (e.g. freshness requirements).

The origin of the CSI and their integrity (criteria S1, S2) can be verified by the dependent entity through the signature the OCSP service provider has put on the CSI. The OCSP service provider has a certificate issued by the respective CA, and the dependent entities can use that in order to verify the aforementioned signatures. A specific method has been defined in [Myer99] for the dependent entities to be able to verify the validity of the certificate of the RevA (OCSP service provider), therefore criterion S4 is met. Furthermore, if the OCSP service provider private key is compromised, no new certificates can be issued; only old ones can be revoked, therefore criterion S5 is met also.

If the CA key is compromised, the OCSP service provider (according to [Myer99]) may provide negative CSI (revoked status) for all the certificates issued by that CA. However, even if this happens, the entity that holds the compromised CA private key may create another OCSP service provider certificate. Dependent entities will trust that certificate and the respective OCSP provider as well, they find out about the compromise of the CA key. A CA private key compromise enables the entity that compromised the key to issue new certificates and revoke old ones. If the CA key was not self-signed (that is, if it was signed by another CA) and CSI on the CA certificate are available at another CA repository, then dependent entities, if they are knowledgeable and motivated enough, will be able to find out rather quickly for the CA key compromise. If this does not apply, a CA key compromise renders all certificates, and the transactions that are performed with them, void; therefore, criterion S3 is not met. Finally, the availability of the mechanism cannot be ensured because a piece of CSI concerns only one certificate and is very fresh, thus has to be re-generated very often; this makes replication very difficult to implement. However, if referrals are used, the levels of the availability of the mechanism can be increased. Therefore, we assert that the mechanism meets the availability criterion (S6) partially.

3.2.8 Freshness-constrained Revocation Authority

This mechanism is not transparent (criterion M1) because the dependent entity has to locate the RevA and retrieve the CSI. The user cannot revoke his certificate by himself, a certificate is considered to be revoked if a RevA does not deliver to the dependent entity a certificate that meets the various freshness requirements. Therefore, offline revocation (criterion M2) is not supported. Delegation of revocation (criterion M3) is supported since the fresh certificates that compose the CSI are issued by a party (RevA) other than the CA. Delegation of CSI dissemination (criterion M4) is also supported since the CSI is not disseminated by the CA, but by the RevA, who is a different entity. Furthermore, RevA may replicate the CSI to other entities as well, possibly to entities that are locally trusted by a group of users. The certificate path validation is performed (criterion M5) by the RevA; based on the certificate path validation, RevA issues (or does not) a fresh certificate that verifies the validity of the certificate of the authenticating entity. Support for referrals is not included in this mechanism (criterion M6), neither does the mechanism consider the reasons for the revocation of a certificate when it validates a certificate path (criterion M7).

The CSI delivered by this mechanism can be as much fresh as the dependent entities, CAs and Policy CAs wish, therefore criterion E2 is met. This mechanism also meets the adjustability criterion (E7) since the dependent entities can opt to accept less fresh short-lived, RevA certificates in order to minimise the online communication with the RevA, who could deliver fresher certificates. However, the timeliness criterion (E1) is not met; RevA stops publishing new, fresh short-lived, RevA certificates the moment it realises that a certificate needs to be revoked (thus meeting the Emergency CSI criterion, (E4), but a dependent entity may rely on a less fresh short-lived, RevA certificate in order to validate a certificate, thus the dependent entity will not know at that time that the certificate has been revoked. Revocation is

bounded, the boundary being the time the RevA will realise that a certificate needs to be revoked (criterion E3).

This mechanism does not disrupt the normal working practice of the dependent entities, since the latter set freshness requirements and it is the certificate holders (authenticating entities or signers) that have to provide the appropriate CSI to the dependent entities. However, the certificate holders have to be knowledgeable enough in order to be able to locate a CSI source, ask for CSI to be produced that meets these requirements, retrieve it and communicate it to the dependent entities. There is no mechanism defined in order to do the aforementioned task in a transparent way, as far as the certificate holder is concerned, therefore interaction with the latter is necessary. Furthermore, if a RevA is using a different mechanism for producing and disseminating CSI, it will have to change to this mechanism (or support this mechanism, too), therefore the tasks the RevA has to carry out are increasing. Therefore, this mechanism does not meet the economy criterion (E5).

Criteria S1 and S2 are met; the origin and integrity of the CSI contained in the "validity" certificate can be verified through the RevA signature on that certificate. However, no method has been specified in [Stub95], in order to verify the validity of the RevA certificate (criterion S4). This has to be done manually, possibly by contacting the CA, due to the lack of such a method.

Criteria	Compliance
M1. Transparency	
M2. Offline revocation	
M3. Delegation of revocation	√
M4. Delegation of CSI dissemination	√
M5. Delegation of the certificate path validation	√
M6. Referral Capability	
M7. Revocation reasons	
E1. Timeliness of CSI	
E2. Freshness of CSI	√
E3. Bounded revocation	√
E4. Emergency CSI capability	√
E5. Economy	
E6. Scalability	
E7. Adjustability	√
S1. CSI disseminator authentication	√
S2. CSI integrity	√
S3. CA compromise	√
S4. RevA compromise	√?
S5. Contained functionality	
S6. Availability	

Table 8: Freshness-constrained Revocation Authority compliance with criteria

If the CA key is compromised, new certificates can be generated but existing ones cannot be revoked, unless a successful DoS attack is launched at the RevA as well, therefore criterion S3 is met. If the RevA key is compromised, new certificates cannot be issued but existing, valid certificates can be revoked (the RevA will stop issuing short-lived, RevA certificates for them) and existing certificates that had been revoked could appear to be valid again (the RevA will be issuing short-lived, RevA certificates for them). Since revoked, possibly compromised, certificates can be made valid, criterion S5 is not met. Finally, the availability of this system

will probably not be high, since there is a single RevA that issues the short-lived, RevA certificates and if the dependent entities require very fresh CSI, the RevA may have to issue continuously this kind of certificates. Digitally signing is a computationally intensive procedure and if the RevA has to serve a large number of CSI requests, the response could be slow or even temporarily unavailable. Therefore, the criterion about availability (S6) is not met. In addition to that, the re-issuance and distribution of short-lived, RevA certificates entails large communication, processing and storage costs for every party that is involved in the operation and use of the mechanism, therefore neither the scalability criterion is met (E6).

4. TOWARDS AN ALTERNATIVE MECHANISM

The contemporary applications where electronic transactions take place differ from the older applications where electronic transactions are involved (e.g. the banking sector) in a number of ways. Some of these differences are the following:

1. electronic transactions in contemporary applications do not take place between entities in a closed group (e.g. in contrast to bank customers and banks),

2. electronic transactions in contemporary applications do not occur within private, controlled networks (such as the banking network or other private, corporate networks); instead, public, unsafe and uncontrolled networks are used,

3. a dynamic, constantly increasing, group of entities have the capability to offer services over public networks; these entities do not always have the capability to perform a detailed security analysis of the way their services operate and the way their electronic transactions are carried out,

4. the users of contemporary networked systems are not necessarily experienced computer users; the majority of users of networked systems are not sensitive or cautious enough about security measures. Even when they are, there are other reasons that could lead them into taking the wrong security decisions on certain occasions [Spruit98].

Therefore certain security-related processes have to be automated, or at least be made more transparent, for those users to be able to offer and receive securely electronic services. This is the reason why the use of credit and debit cards has expanded. End-users of the credit and debit card system (cardholders) and the respective dependent entities (e.g. merchants) do not need to take any special security measures in order to take advantage of the credit and debit card system. All they have to do is adopt simple human procedures in order to facilitate the operation of the underlying security system.

We consider that this has to be done as well with Certification Authorities and the security services they offer. Authenticating entities, signers and dependent entities should not have to carry out special security procedures. There has to be a security infrastructure that allows them to profit from the use of certificates without them having to contribute to the operation of that security infrastructure, except for some simple procedures which would ensure that the security infrastructure works for their profit.

Considering the above, it is the author's opinion that transparency (criterion M1, see section 3.1.1) should be met by the CSI dissemination mechanisms that are used. However, as presented in section 3.2, none of the mechanisms for the dissemination of Certificate Status Information meets the transparency criterion. A CSI dissemination mechanism will be of use for both the certificate holders and the dependent entities if it can provide transparency in the dissemination of the CSI, both for certificate holders and dependent entities. Neither the certificate holders (authenticating entities or signers) nor the dependent entities should have to contribute manually to the location and validation of CSI. This is not the job of the certificate holders or of the dependent entities, as it is not the job of the merchants to locate or validate status information regarding credit cards. This is a security service that the security providers (credit card organisations in one case, Certification Authorities in the other case) should provide to users in a transparent way, for their profit. Certainly, in some cases at least, the final decision regarding CSI must come from the human operator but in most cases human operators and users of electronic services should concentrate on offering or using the electronic services they are interested in, and not on performing security-related tasks.

Certainly, one could argue that it is the responsibility and duty of the certificate holders to provide the dependent entities with the most current CSI [Rivest98]. Rivest also claims that it should be the responsibility of the dependent entities to set the requirements that the CSI they are presented with, by the certificate holders, has to meet. However, the persons that run the services offered by the dependent entities are not always security experts. They are certainly aware of the importance of security controls, much more than the certificate holder on an average, but that does not make them security experts. Furthermore, the dependent entities as well as the certificate holders are usually not so knowledgeable, motivated and perceptive [Spruit98] regarding security controls. They may not always go through any required security steps that will ensure a secure communication between those two. Therefore, the certificate holders and the dependent entities are bound to make security mistakes or overlooks, consciously or unconsciously, leading to direct or indirect failing. Human failing is in many cases the reason for the malfunction of a security mechanism, the incorrect use of a security service or a security breach [Spruit96]

The security experts in this case are the people behind the Certification Authority that provides the certificates, CSI, timestamping or other services. It is the CAs that should offer CSI to the dependent entities and the certificate holders, in a *transparent* way, in order to avoid human failing.

Moreover, one should not assume that online dissemination of CSI is always feasible because the online capability of the majority of users is somehow restricted. However, probably the majority of users nowadays connect to a Wide Area Network (WAN) very frequently and for a small period of time. It could be the case of users that download their e-mail from their home computer once a day, it could be the users connecting with their computer at work, to the corporate intranet and thus to the Internet or other WAN their organisation's intranet is connected to. It could be students, using the university computing facilities. Therefore, users are not capable of having uninterruptible communication with CSI providers (which renders typical online CSI mechanisms rather useless) but they can have online capability frequently, at least for a short period each time. Therefore, the CSI mechanism has to have both offline and online characteristics. We claim that this can be done with the use of software agents.

Agents residing in the domain of dependent entities' will decide what CSI the dependent entities need or will need, roam the network in order to locate that CSI and retrieve it for the dependent entities once they are connected again to the network. The major difference of such a mechanism to the others would, or rather should, be *transparency*. The certificate holders or dependent entities, who are sometimes inexperienced computer users, would not have to understand exactly how revocation works and why they should frequently check for CSI, before accepting a digital signature or a certificate-based authentication as valid. However, they might have to answer some questions at certain points in order to facilitate the certificate path validation. Even in this case, the number of the questions has to be restricted and the way they are posed has to be studied carefully, otherwise the transparency will cease to exist and users or dependent entities may fall back to a position where they should strive to understand how the CSI mechanisms work and what should they do in order to help them operate without faults. This often leads computer users to direct or indirect failing [Spruit98] or 'authorisation fatigue'. If users are asked to answer a lot of security-related questions or perform a lot of security-related tasks they do not completely comprehend, then they might choose the default answers to these questions, perform the tasks in the wrong way or do not perform them at all, because they might think that they are not important or that the default answer to questions is a safe option.

Transparency is a solution to these problems. However, a transparent mechanism for the dissemination of CSI must not lack the security features provided by the other non-transparent mechanisms. An agent-based mechanism should meet certain criteria, such as those we have developed in section 3.1, in order to carry out successfully the task which it is charged with.

In the next section we present an introduction to Agents and in section 4.2 we present a prototype of a high-level specification for the operation of an agent-based mechanism for the dissemination of CSI. We call this mechanism ADOCSI (Alternative Dissemination of Certificate Status Information).

4.1 Agents

Agents have recently been (and still are) a major research topic, both for the academia and for the industry, for various reasons [Petrie97]. The researchers, trying to define accurately their view and specifications of Agents, came up with a variety of alternative names or adjectives. A short review of those follows [Nwana96], [Chess95]:

1. *Static*. Agents that do not transport themselves to execution environments other than the one they were originally. They remain at that execution environment and use other methods in order to communicate with other agents, such as RPC,

2. *Mobile*. These agents can transport themselves to other execution environments in order to communicate locally with other agents and at the same time they can preserve the external state they had in the previous location,

3. *Deliberative*. Agents that contain internal reasoning and collaborate with other agents in order to achieve their goals,

4. *Autonomous*. Agents that have states and goals and do not need user feedback or interaction (except perhaps an initial user feedback) in order to carry out their tasks and reach their goals,

5. *Co-operative*. Agents that co-operate with other agents in order to reach their goals,

6. *Interface* (also called User). Agents that sit between another agent and a user. The user communicates with the interface agent only, and the latter transforms the user's goals into formal statements and procedures and assigns those to the other agents; the interface agent launches agents on behalf of the user, co-ordinates them and provides the results of their work back to the user,

7. *Heterogeneous*. Agents that encompass several of the characteristics of the other agents and can talk to other agents through the use of a standard agent communication language, thus providing for interoperability between agents.

This wide spectrum of names result in semantic overlap and confusion. Furthermore, since there were no criteria for 'agenthood', some researchers came up with software constructs which they called Agents, even though they did not have any of the functionality that characterises Agents [Petrie97]. Genesereth [Gene94] argues that the criterion for 'agenthood' is a behavioural one; a software application is an agent if and only if it communicates with other software applications, using an Agent Communication Language (ACL). According to Genesereth, typical object-oriented applications cannot be characterised as agents since the messages exchanged between the objects may have different semantics when exchanged between different objects. On the contrary, the semantics of a language construct in an ACL is always the same, no matter which two agents is it that this message is being exchanged between. There are many criteria one can use in order to develop a taxonomy of Agents. However, the lack of a widely accepted standard practice which could act as a reference point, renders these taxonomies rather inaccurate or at least not in accordance with other ones.

We should clarify at this point that in this document we are not interested in any Artificial Intelligence properties that agents may have. We are only interested in the distributed functionality they offer, since we can use that in order to meet some of the CSI dissemination criteria (see section 3.1) that the other CSI dissemination mechanisms do not meet. The agents we describe in this document are mobile, deliberative, heterogeneous and (some of them) interface agents. However, since we are not interested in establishing a new name for the agents we discuss, we will be referring to them simply as Agents or CSI Agents. We would like to warn the reader that the term CSI Agents does not constitute a new flavour of Agents. We will use it in order to have a common name as a reference to the mechanics of our specification throughout this document. The agents we will describe must use a standard way of talking to each other. One of the examples of languages used by agents to communicate is the Knowledge Query and Manipulation Language (KQML) [Lamb97], [KQML], which describes programmatic content and the Knowledge Interchange Format (KIF) [Gene92b], [Chess95] which can be used to form knowledge representations, express tasks and goals. Another example is the more recent Agent Communication Language (ACL) of the Foundation for Intelligent Physical Agents (FIPA) [FIPA98b]. The advantage of using such a language for agent communication instead of messages with proprietary content format is that it will be possible for the agents involved in our scheme to be updated transparently by their owners or developers, without affecting the interoperability or agent inter-communication. CSI Agents could expire, and their owners or developers will have to (transparently) replace them with new ones at that time (see section 4.3). These actions are not going to affect the operation of our scheme, since the communication between agents will remain the same, being based on one of the aforementioned agent communication languages.

The agents we will present must meet the following requirements:

1. they must be able to suspend execution and resume it at another execution environment,

2. they must retain their state, when transporting themselves to other execution environments,

3. they must be able to create child agents and deploy them,

4. they must be able to select a network location, out of a list of locations, with the least network congestion,

5. they must be able to communicate the retrieved information back to their owner or to their owner's application that spawned the agent.

We should note here that we are not going to examine the mechanics of the implementations of Agent infrastructures. There are such implementations [Frost96], [Aglets], [Odyssey] that meet the requirements set in this section, as far as Agent design and operation is concerned.

4.2 ADOCSI

In this section we present the prototype of an alternative mechanism for the dissemination of CSI. We shall call this mechanism *ADOCSI* (Alternative Dissemination Of Certificate Status Information). The Agents described in this mechanism *exchange CSI, in one (or more) of the formats described in section 2*. This mechanism makes extensive use of the properties and functionality of the CSI mechanisms presented in section 2, and it provides at least one additional feature: the *transparency in the dissemination of CSI* (criterion M1, see section 3.1.1).

Let us consider the entities that take part in our mechanism (Figure 2):

1. CA: The three CAs in Figure 2 offer the standard services a CA should offer. In addition to that, they develop the CA-CSI Agents and the User-CSI Agents, which are used throughout ADOCSI for the dissemination of CSI

2. Agent Meeting Places (AMP) [Chess95]: They are also called Agent Platforms (AP) [FIPA98a]. These provide the physical infrastructure where agents can be transported to and communicate with other agents that are already there. In ADOCSI, AMPs must also possess a X.509v3 certificate which will be used in order to authenticate the AMPs against the Agents that visit them. If CAs decide to have their CA-CSI Agents (see CA-CSI Agents) installed in more than one AMP, the agents sent by the dependent entities in order to retrieve CSI (see User-CSI Agents) can select to visit a specific AMP if it serves the least number of CSI requests and if the communication lines to that AMP's direction are less congested to others. AMPs could be operated and controlled by the CAs themselves or they could be a separate Trusted Third Party.

3. Directory Facilitator Agent (DF): These agents are resident (static) within the AMPs. They keep a complete and up-to-date register [Gene92], [FIPA98a] with the services offered by the Agents that are in the AMP and the methods other Agents should use in order to communicate with the former. In our mechanism, we assume that the DF will also be able to perform verification of the digital signatures on Agents (see section 4.3). However, in an implementation it will most probably be another Agent inside the AMP that will do this job. We assume that the DF can undertake that task as well in order to reduce the complexity of our description

4. Dependent entity: An entity that wishes to verify the validity of the certificate of another entity. The latter may have tried to authenticate (authenticating entity) against the dependent entity in order to access the services offered by the dependent entity, or the latter may have received offline, signed communication (e.g. e-mail) from another entity (signer). In both cases, the dependent entity wishes to verify the validity of the certificate presented by the aforementioned entities (authenticating entity or signer).

5. Authenticating Entity: An entity that attempts to authenticate against the dependent entity in order to gain access to the services offered by the latter.

6. Signer: An entity that has sent to the dependent entity a signed piece of information.

7. Certification Authority Certificate Status Information (CA-CSI) Agent: This Agent is developed by the CA and it contains the latest available CSI, in one (or more) of the formats discussed in section 2. Furthermore, CA-CSI Agents have a short validity period. Entities communicating with CA-CSI Agents can verify the validity period of such an Agent because the CA-CSI Agent contains a timestamp (time of CSI inclusion in the Agent) and a validity time period, and the Agent is signed with the CA private signing key; CAs issue new Agents if their validity period is close to the expiration date or if there is fresher CSI available at the CA, compared to the CSI stored in the Agent. Once a new Agent is issued, it is sent to the AMP. In case the Agents also serve CSI that is very frequently updated (e.g. Freshest CRLs, see section 2.2), then this CSI can be stored externally, in the AMP, in a location where Agents can have access to. User-CSI Agents communicate with CA-CSI Agents at the AMPs in order to locate, validate and retrieve the CSI they are interested in. If CA-CSI Agents can deliver CSI in more than one ways or formats, then they negotiate with the requesting agents in order to deliver the CSI in a manner that meets best the CSI requirements set forth by the requesting agents, like freshness of information, length of information. In

order to increase the availability of the system, CAs may decide to have their CA-CSI Agents installed in more than one AMP, thus allowing for load-balancing of the communication traffic due to CSI requests.

8. User Certificate Status Information (User-CSI) Agent: This Agent is also developed by the CAs and it is given to the dependent entities, that is any entity that wishes to use the agent-based scheme we present in order to retrieve CSI. This Agent is signed by the dependent entity every time the Agent is sent to locate and retrieve CSI. This signature is used throughout the Agent's course in order to verify its integrity and origin.

9. Interface Agent: The Interface Agent (IA) interacts with the User-CSI Agent and the applications that the dependent entity is using. IA generates and launches a new User-CSI Agent whenever the aforementioned applications need new CSI. Furthermore, it is the Interface Agent that installs the retrieved CSI at the local repository of the dependent entity, perform any certificate path validation needed and collaborate with the applications the dependent entity uses in order to inform the entity of any progress or changes concerning the requested or retrieved CSI. The IAs that interact with certain software applications and the User-CSI Agents should be developed by the software companies that developed the aforementioned software applications.

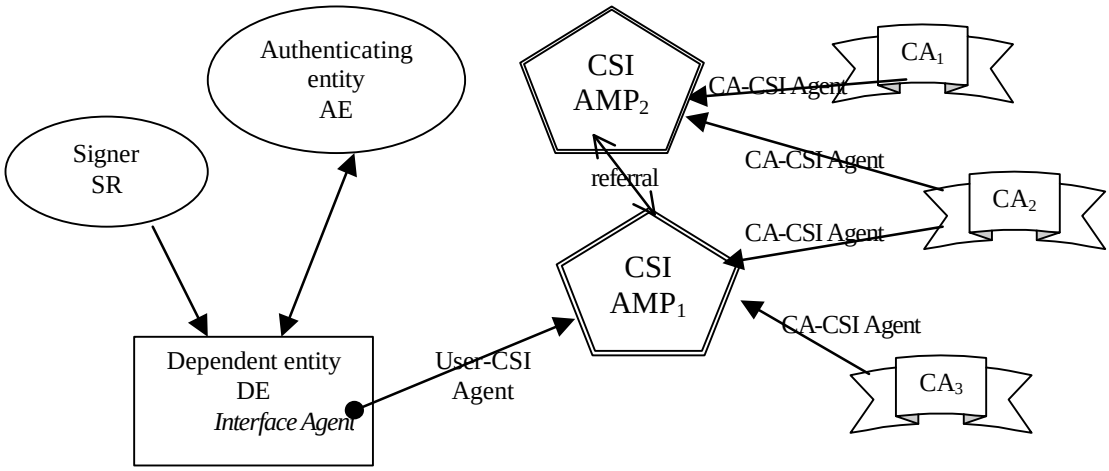

Figure 2: ADOCSI infrastructure

4.2.1 AMP Location Function

The AMP location function can be implemented in the following ways:

1. location information is stored at the CA, and the User-Agent will have to retrieve it before transporting itself to the AMP,

2. location information is stored in the CA-CSI Agent and it is given to the User-CSI Agent at the AMP,

3. location information is stored at the certificate of the authenticating entity (or signer) and the Interface Agent will have to retrieve it and deliver it to the User-CSI Agent,

4. location information is stored in the CA certificate and the Interface Agent will have to retrieve it and deliver it to the User-CSI Agent,

5. location information is stored in the User-CSI Agent.

The location of the AMP or AMPs that host the CSI-Agent of a CA can be stored in a repository maintained by that CA, such as the Directory. The location information should be signed by the CA in order to enable the dependent entity to verify its integrity, unless the communication protocol supported by the CA repository includes integrity checking. If this is the way the location function is implemented, the User-CSI Agent will have to lookup the location information at the CA repository and verify the CA signature on that, before transporting itself to the AMP. A possible enhancement could be caching the location information in the local user environment, in order to avoid having the User-CSI Agent lookup the location information every time, before transporting itself to the AMP.

However, a CA may decide to stop trusting a specific AMP for hosting the CA-CSI Agent, because malicious behaviour has been observed; the AMP may have tried to tamper with the CA-CSI Agent in order to make it deliver incorrect or incomplete CSI. Although there are mechanisms for protecting the Agents from malicious AMPs, they are either still under research or they are very expensive to implement. If the AMP is no longer trusted by the CA, the User-CSI Agent who is using the cached location information will not know about it. This is one of the cases where the short lifetime of the CA-CSI Agents could prove to be useful. If a CA decides to trust no longer an AMP for hosting the CA-CSI Agent, the CA will no longer update the CA-CSI Agent in that AMP, therefore even if a User-CSI Agent visits the specific AMP, it will not meet a CA-CSI Agent and therefore it will not retrieve incomplete or incorrect CSI. Certainly, there is a period (when the AMP starts tampering with the CA-CSI Agent and before the CA becomes aware of that and stops installing new CA-CSI Agents in that AMP) when the control of the CSI could be at the hands of the AMP.

There is a minor variation of the aforementioned location function. This can be done in case the CA decides that online communication between the dependent entities and the CA should be avoided or at least should not be frequent. The location of the AMP or AMPs that host the CA-CSI Agent can be stored inside the CA-CSI Agent himself. Certainly, the dependent entity would have to retrieve this information for the first time from the CA, manually. Once this is done and the User-CSI Agent knows which AMP to visit for the first time, it will cache at the local storage area of the dependent entity that information and update it whenever the CA-CSI Agent gives the User-CSI Agent new location information.

AMP location information can also be stored in the X.509v3 certificate delivered by the authenticating entity (or signer) to the dependent entity. The Interface Agent can retrieve that information from the certificate and deliver it to the User-CSI Agent, before sending the latter to the AMPs in order to retrieve CSI on the aforementioned certificate. In this case, the User-CSI Agent does not have to find the AMP location from the CA or the CA-CSI Agent, therefore the communication and processing costs related to the actions of User-CSI Agent are decreased. However, the costs associated with distributing the certificates themselves are increased in this case, since the certificates contain also the location information. However, the dependent entity is open to Denial of Service attacks, since the location information can only be changed (by the CA) whenever a new certificate is issued (or renewed). Therefore, if all AMPs mentioned in the certificate stop offering services, or if the CA stops trusting them and therefore stops sending updated CA-CSI Agents to those AMPs, the User-CSI Agent of the dependent entity will not be able to locate an AMP to retrieve CSI from.

A variation of the aforementioned location function is to store the location information in the CA certificate. In this case, the costs associated with the communication and processing done by the User-CSI Agent are also decreased, and costs associated with distributing the certificates are not that increased since it is only the CA certificate that is bigger, not all the

certificates issued by the CA. However, the dependent entity is more vulnerable to Denial o Service (DoS) attacks, because the validity period of a CA certificate is longer than that of any other certificate issued by that CA.

A way to deal with the possible DoS attacks related with the last two types of location functions would be to have the first type of location function operational as well. If the third or fourth type of location function fails (DoS), the first type of location function can provide the dependent entity with the AMP location that is needed.

The CAs could develop User-CSI Agents in such a way that the Agents themselves decide which location function to use, every time they are called to retrieve CSI. This decision of the Agents should be based on which location functions are available, which location function was used the last time the Agent was called to retrieve CSI, whether that function executed successfully or not, and on the user preferences regarding the location function to be used. Furthermore, the User-CSI Agent should (as all other Agents involved in this scheme) use a predetermined Agent Communication Language (ACL) [KQML], [Gene92b], [FIPA98b]. Therefore, the CAs will have the freedom to change the way they disseminate the AMP location information, if the need presents itself, without having to modify the underlying infrastructure and without the CSI services provided to the dependent entities being interrupted. If User-CSI Agents are implemented in the aforementioned way, any alteration in the rest of the infrastructure depicted in Figure 2 (e.g. change or upgrade of the CA-CSI Agent, change of the underlying applications used by the dependent entities etc) will not affect the operation of the User-CSI Agent and will not require that a change be made in the latter, as well.

Finally, the location information could be stored inside the User-CSI Agent. The Agent would not need to locate and retrieve the AMP location information. His operation will be simpler and the size of the User-CSI mobile code application will be smaller, therefore communications costs related to the transport of the User-CSI Agent will reduce.

4.2.2 CSI Location, Validation and Retrieval

Once the User-CSI Agent is in the AMP, it contacts the Directory Facilitator in order to locate the CA-CSI Agent that holds the CSI it is interested in. The User-CSI Agent has to form a query in a predetermined Agent Communication Language and send it to the Directory Facilitator, who will reply to the User-CSI Agent using the same ACL as well. If the CA-CSI Agent that contains the information the User-CSI Agent is interested in, does not reside in that AMP then the DF should refer the User-CSI Agent to another AMP.

DF maintains a database with all the CA-CSI Agents that reside in the AMP and the respective CSI they hold. Whenever a CA-CSI Agent is transported to the AMP by the respective CA, it has to contact the DF in order to inform the latter of the CSI it holds. The DF should also update the aforementioned database by polling the resident CA-CSI Agents. This ensures that even if the DF was unavailable at the time a new or updated CA-CSI Agent tried to contact the DF or if a CA-CSI Agent was removed abruptly (e.g. accidental cease of operation, perhaps due to a software bug) from the AMP, the DF will become aware of that and will have the database updated with the relevant information. A CA-CSI Agent may have to be removed from the AMP because the CSI it contains is no longer valid or because the Agent has expired (see section 4.3).

Having located the CA-CSI Agents they seek, User-CSI Agents communicate with them and perform a series of queries, using a predetermined ACL. Since CA-CSI Agents could be able to deliver CSI in a number of different formats (see section 2), User-CSI Agents will negotiate with the CA-CSI Agents in order to have the latter deliver the CSI in the way that meets best the requirements set by the dependent entity (see section 4.2.4). The negotiation parameters that can be used, include:

1. *Freshness of CSI*. The freshness of the CSI received by a User-CSI Agent may vary, depending on the mechanism the CA-CSI Agent used to deliver that information,

2. *Length of CSI*. The length of the CSI received by a User-CSI Agent may vary, depending on the mechanism the CA-CSI Agent used to deliver that information. There are mechanisms (e.g. CRL) where the communicated CSI contains always more information than necessary (e.g. CSI on certificates that the dependent entity is not interested in),

3. *Estimated time of CSI delivery to the dependent entity*. The time that is estimated to take for the CSI to be delivered to the dependent entity varies, depending on the length of the information and the availability and bandwidth of the network that connects the AMP and the dependent entity. The User-CSI Agent should be able to evaluate the bandwidth and availability of the aforementioned network and decide on the time it will take to carry the information back to the dependent entity. Furthermore, the User-CSI Agent should request from the CA-CSI Agent for CSI that meets certain length restrictions, in order to be able to transport it back to the dependent entity within the specified time period,

4. *Delegation of certificate path validation*. This validation can be performed either by the CA-CSI Agent or the Interface Agent (see sections 4.2.3 and 4.2.4), depending on the mechanism used for delivering the CSI to the User-CSI Agent,

5. *Bounded revocation*. The User-CSI Agent may choose to prefer one CSI dissemination mechanism to another because the former supports bounded revocation (see section 3.1.2).

Once the User-CSI Agent has at his disposal the CSI, it must validate that CSI. The Agent must verify that the CSI it has at his disposal, concerns the certificate it has been instructed to find CSI on. If it is so, the Agent must transport itself back to the dependent entity. If the latter is not connected to the network at that point of time, the User-CSI Agent should be allowed to remain at the AMP for a predetermined amount of time, waiting for the dependent entity to connect back to the network.

4.2.3 Certificate Path Validation

Validation of a certificate path can be performed by the Interface Agent or it can be delegated to the CA-CSI Agent. In any case, the dependent entity should be able to verify the authenticity of the received CSI. If the certificate path has been validated by the CA-CSI Agent, the dependent entity should be able to verify that the received, processed, CSI is authentic; if the certificate path is to be validated by the Interface Agent, the dependent entity should be able to verify the authenticity of the certificate path information received (see section 4.3).

4.2.4 Interface Agent

The role of the Interface Agent (IA) is to provide a link between the User-CSI Agent and the PKI-aware applications that the dependent entity uses. The IA has access to the certificate and CSI repository of the dependent entity (e.g. the computer filesystem), which it updates as soon as new CSI is available. Furthermore, the PKI-aware applications that reside at the site of the dependent entity communicate with the IA in order to request CSI on specific certificates. It is the IA that will then form a CSI query, using a predetermined ACL, describing the CSI that has to be retrieved and the retrieval negotiation parameters (see section 4.2.2). Furthermore, the IA may inform the User-CSI Agent on the location of the AMP to visit, if that information is stored in the certificate of the authenticating entity or signer, or in the respective CA certificate (see section 4.2.1). After that, the IA spawns a User-CSI Agent and assigns to it the task to perform the ACL query. Once the User-CSI Agent returns to the dependent entity location, it

communicates with the IA in order to deliver the CSI that has been retrieved. The IA must be able to identify the User-CSI Agent, once it comes back, verify his integrity, remember the query it was assigned with and examine whether the User-CSI Agent did complete his task, to retrieve the requested CSI. If certificate path validation has not been delegated to the CA-CSI Agent, the IA will perform the certificate path validation or leave this task for the application itself, if the application supports certificate path validation. The IA will install the new CSI in the certificate and CSI repository of the dependent entity and inform the application of the new CSI that is available.

In case the dependent entity wishes to locate CSI in order to verify a certificate that belongs to a signer, who has sent a signed document to the dependent entity, then there is no online communication between the certificate holder and the dependent entity. Therefore, there is no need for online location, retrieval and validation of CSI. If this is the case, the Interface Agent should send the User-CSI Agent to search for CSI on that certificate and notify the dependent entity (the human operator) that he may logoff the network if he wishes so. The User-CSI Agent must be checking frequently (polling) for the existence of the dependent entity in the network and return to it when the dependent entity connects back to the network. In this case, where the dependent entity may logoff the network, the User-CSI Agent may return to the dependent entity by other, offline communication means as well (e.g. e-mail). It is the duty of the IA then, to check frequently the mailbox of the dependent entity and locate any User-CSI Agents there and retrieve them.

4.3 ADOCSI Security

We have presented in the previous sections an alternative mechanism for the dissemination of certificate status information, which we call ADOCSI. In this section, we present the threats this mechanism faces, and the respective security services and mechanisms that have to be in place in order to deal with these threats. The threats ADOCSI has to deal with are the following [Gritz97]:

4.3.1 Unauthorised modification or replacement of Agents

An entity other than the CA could manage to modify a CA-CSI Agent before it is installed in the AMP, or install one that has been developed by that entity itself, in order to have that Agent give incorrect or false CSI to the User-CSI Agents. To deal with this threat, Agents should be distributed in a secure manner [Zhang97]; the integrity and origin of the CA-CSI Agents should be verifiable both by the AMPs and the User-CSI Agents. This can be done if the CA-CSI Agents are signed by the CA that sent them to the AMP. Both the AMP and the User-CSI Agents must have a copy of the CA certificate. The DF of that AMP and the User-CSI Agent must use the respective CA public key and verify the signature on the CA-CSI Agent. The AMP must contain the CA certificates of the CAs it receives CA-CSI Agents from, in a repository local to the AMP. The User-CSI Agent must also contain the CA certificates that will be needed to verify the digital signatures on the CA-CSI Agents the User-CSI Agent will communicate with.

An entity other than the dependent entity may attempt to modify or replace the User-CSI Agent while the Agent is distributed to the dependent entities by the CA that developed it, while the Agent is transported to an AMP or while the Agent is returning to the dependent entity. The CA that developed the User-CSI Agent should sign that Agent as well, in order to deal with the aforementioned threats. If the signature of that CA is on the User-CSI Agent, the dependent entity and the DF will be able to verify the integrity of the Agent once they receive it.

4.3.2 Unauthorised modification of information contained in the Agents

The CA-CSI Agents transport the CSI from the CAs to the AMPs. The integrity and origin of the CSI contained in the CA-CSI Agents can also be verified, by verifying the CA digital signature on the CA-CSI Agent. This signature might prove to be redundant for some cases of CSI (e.g. CRLs) where there is already a mechanism for verifying the integrity of the CSI (e.g. in the case of CRLs it is, again, the CA signature on the CRL).

The User-CSI Agents also carry information. When the User-CSI Agent leaves the dependent entity to be transported to an AMP, it carries the information that it will use in order to form the CSI queries it will direct to the CA-CSI Agent. A malicious entity might modify that information while the Agent is transported in order to prevent the dependent entity retrieve correct or complete CSI. This can be prevented if the dependent entity signs the Agent as well, before transporting it to the AMP. The AMP must verify both signatures (CA signature and signature of dependent entity) before allowing the execution of the User-CSI Agent.

If an AMP does not contain all the CA-CSI Agents a User-CSI Agent needs to contact, the AMP will refer the User-CSI Agent to another AMP in order to find the rest of the CSI it is looking for. The User-CSI Agent may be modified by an entity (without authorisation) while it is transported to that other AMP. In order to prevent that from happening, the AMP that refers the Agent must sign the Agent as well, in order to provide integrity protection for the CSI the Agent has retrieved from this AMP.

Finally, when the User-CSI Agent has gathered all the CSI it needs, it transports itself back to the dependent entity, either immediately or whenever the latter is connected again to the network. The Agent carries at that point some CSI for which the dependent entity cannot verify the integrity. This is the CSI the Agent retrieved from the AMP it is in at that moment, before transporting itself back to the dependent entity. The User-CSI Agent could generate at that time a symmetric key, encrypt that key with the public key of the dependent entity (the certificate of the dependent entity is contained in the User-CSI Agent) and store it along with the other information it carries. Before leaving the AMP, the User-CSI Agent should generate a Message Authentication Code (MAC) [Ford94], using that symmetric key; that MAC should be applied on the Agent itself and all the information it carries. Using that MAC, the dependent entity (the Interface Agent) can verify the integrity of the User-CSI Agent and of the information it retrieved for the dependent entity.

Verification of the origin of the CSI a dependent entity receives may be possible or not, depending on the CSI mechanism that the User-CSI and CA-CSI Agents decided to use in order to exchange the CSI that the former was looking for (see section 3.2, for the evaluation of those mechanisms concerning their capability for verification of the origin of CSI).

4.3.3 AMP masquerade

An entity may attempt to masquerade as an AMP and attract User-CSI Agents. If a User-Agent communicates with a (valid) CA-CSI Agent at an AMP that is controlled by an unauthorised, untrusted and possibly malicious entity, then that entity may manage to take control of the communication between the User-CSI Agent and the CA-CSI Agent and provide the former with false, inaccurate or partial CSI. Even if the two aforementioned Agents employ integrity mechanisms for the protection of their communication, while they are in the AMP, the AMP can always observe the Agents' actions, monitor the data they generate (e.g. symmetric keys that will be used for the protection of the integrity of the communication between these two Agents) and manipulate the communication between them. In order to prevent AMP masquerade, both the CAs and the dependent entities should encrypt their Agents with the public key of the AMP they intend to send their Agents to, before transporting them. Thus, the AMP will be able to decrypt and execute the Agents only if it is the AMP that the CAs or the dependent entities intended to send their Agents to.

4.3.4 Denial of Service

Denial of Service attacks could be launched against the AMPs. In order to deal with this threat, the CA-CSI Agents should be sent by the respective CAs to more than one AMPs. Therefore even if an AMP cannot deliver the CSI dissemination service with the expected quality (e.g. the communication line that leads to the AMP is congested due to the DoS attack, or the User-CSI Agents that reside at the AMP perform too many CSI requests to the CA-CSI Agents, thus consuming much of the processing power of the AMP), a User-CSI Agent may select another AMP to retrieve CSI from.

Another countermeasure that can be used in order to deal with DoS attacks would be to minimise the Central Processing Unit (CPU) time quota that is available to every User-CSI Agent that resides in the AMP, thus reducing the possibility that a malicious User-CSI Agent could use too many system resources. Research is being performed on limiting the CPU time that mobile code can consume [Gorri98].

4.3.5 Replay attacks

A malicious entity might attempt to capture a User-CSI Agent when it is transported back to the dependent entity and send another User-CSI Agent in its place, an Agent that was sent at an earlier date by the same dependent entity. If this is achieved, the dependent entity might be led into believing that the Agent it received is authentic, as well as the information it carries. However, that CSI will be old and possibly invalid. If that CSI has not been generated with a mechanism that supports bounded revocation, then the dependent entity will consider the invalid CSI it has received as valid. In order to avoid that, the dependent entities should include a nonce (Number used ONCE) [Schn96] in their User-CSI Agents in order to prevent such Agent replays.

A malicious entity could attempt to capture the CA-CSI Agents on their way to an AMP and replace them with valid, but old, CA-CSI Agents of the same CA. If this replay attack was successful, then the User-CSI Agents could retrieve old, and possibly invalid, CSI. However, the CA-CSI Agents are timestamped and they also contain a validity period, which is short. If an unauthorised entity attempts to replay an old CA-CSI Agent to an AMP then the AMP will dismiss that CA-CSI Agent as expired.

If an entity had taken control of an AMP it could attempt to store in that AMP only old CA-CSI Agents, with the purpose of disseminating old, and possibly invalid, CSI. The User-CSI Agents would not retrieve CSI from those expired CA-CSI Agents because the former would realise that the latter are expired (by verifying the timestamp and validity period they contain) Agents and therefore they should not trust the CSI they carry.

4.3.6 Malicious Agents

Signing the Agents with the private keys of the Agent developer or the entity that is sending the Agent does not guarantee that the Agent will not act in a malicious way either towards the AMP or towards other Agents. This measure provides only accountability for the actions that the Agent will perform, on behalf of the entity that sent the Agent. Protecting the AMP and Agents, from other Agents has been a topic of research both in academia [Necu98], [Necu96], [Wil98], [Wil99] and in the commercial world [Gong98], [Gong97].

Solutions to this problem include the use of tamper-resistant hardware [Wil98], [Wil99] and the use of language semantics in order to verify the safety of a specific application [Gong97], [Necu98] either dynamically, at run-time, or statically. Furthermore, access control mechanisms can be employed in order to restrict access to the resources or services offered by an Agent [Gong98]. Limiting the resources of an AMP that an Agent can use, in order to carry out certain tasks, is a subject that is also under research [Gorri98].

Besides these, confining the use and transportation of Agents inside network domains [Gritz98], and enforcing security policies for their operation could contribute to a controlled

execution environment for these Agents. Finally, the integration of application development [Jacl97] platforms that can be used for the development of Agents could lead into an integration of the respective security functionality. Such an integration could give to the developers more tools to address the security issues that arise from the operation of Agents [Gritz98].

4.4 Comparative evaluation

In this section we evaluate ADOCSI according to the criteria we have developed in section 3.1, and in comparison with the other CSI mechanisms presented in section 2.

4.4.1 Type of Mechanism

We first evaluate ADOCSI against the first set of criteria (type of mechanism) we have developed in section 3.1. These are the following:

M1. Transparency,

M2. Offline revocation,

M3. Delegation of revocation,

M4. Delegation of the CSI dissemination,

M5. Delegation of the certificate path validation,

M6. Referral capability,

M7. Revocation Reasons.

ADOCSI meets criterion M1 about transparency. Neither the certificate holders nor the dependent entities have to perform any tasks in order to locate and retrieve the CSI they need. Probably the only point at which human interaction is needed in this mechanism is when the Interface Agent (IA) is installed at the local repository of the dependent entity. At that time, IA should ask the dependent entity's human operator a series of appropriately formed questions in order to create a profile for the use of the ADOCSI mechanism. These questions will define the values of the negotiation parameters (see section 4.2.2) the User-CSI Agents will use when communicating with the CA-CSI Agents. However, special care has to be taken in the format of those questions; the dependent entity might not be an expert in revocation mechanisms, therefore the aforementioned questions should be put in such a way that the respective answers will provide the feedback that IA needs, and at the same time the human operator will not be confused about the questions asked and the answers he has to provide. Furthermore, the questions must be such that the human operator will be motivated to take the time to answer them. None of the CSI dissemination mechanisms we have examined at sections 2, 3.2 meet the transparency criterion (M1). All of those mechanisms require that either the dependent entities or the certificate holders (authenticating entities or signers), or even both, be motivated, knowledgeable and perceptive enough in order to carry out the necessary procedures to locate and retrieve the CSI they need. ADOCSI does not have such a requirement neither from the dependent entities nor from the certificate holders.

ADOCSI also meets criterion M2 about offline revocation. A CA may decide that some or all of the entities that were certified by that CA should be able to revoke their certificates themselves. In this case, the CA would have to incorporate the functionality of the Suicide Bureaus (see section 2.5) in the AMP or AMPs it owns. The CA-CSI Agents could then inform the User-CSI Agents on the status of the certificate or certificates the former were interested in,

based on the suicide notes that will have been received. The format of the CSI the User-CSI Agents will receive and the mechanism that will be used for the generation of that CSI does not have to be related to the ones described in section 2.5. The CA-CSI Agent uses the CSI that it contains and the CSI that could be stored locally at the AMP in order to decide on the validity of a certificate. The CA-CSI Agent can either forward that CSI directly to the User-CSI Agent or perform itself the certificate path validation function (criterion M5) and send responses to User-CSI Agents that could follow any of the formats and mechanisms described in section 2.

	Type of Mechanism						
CSI dissemination mechanism	M1	M2	M3	M4	M5	M6	M7
CRL				√		√	
FCRL				√		√	
RCRL				√		√	
Enhanced CRL Dist. Options				√		√	
Positive CSI		√	√	√	√	√	
CRS			√?	√			
OCSP			√	√	√	√	
Freshness constrained RevA			√	√	√		
ADOCSI	√	√	√	√	√	√	

Table 9: Evaluation of CSI mechanisms based on the type of mechanism

ADOCSI focuses on the dissemination of the CSI itself, rather than the revocation of the certificates. Therefore, if the mechanism used by ADOCSI to collect CSI (see section 2) supports delegation of revocation (criterion M3), then ADOCSI supports it as well. The actual revocation will occur according to the revocation mechanism and the dissemination of the respective CSI will occur through ADOCSI, therefore criterion M3 is met.

Although it is the CAs that disseminate the CSI, indirectly through the CA-CSI Agents, the dependent entities do not have to contact the CAs or RevAs; the User-CSI Agents will only contact CA-CSI Agents, wherever they may find them and return CSI to the dependent entities, therefore criterion M4 is met.

Moreover, ADOCSI meets criterion M6 about referrals. If a User-CSI Agent cannot find the appropriate CSI at a specific AMP, he is referred to another AMP (see sections 4.2.1, 4.2.2). This referral derives either from the information contained in the CSI itself (e.g. Redirect CRL, see section 2.3) or from the AMP. In the first case, the User-CSI Agent will ask from the CA-CSI Agent and probably the DF as well, where should he transport itself to locate the CSI in question. In the second case, the AMP transports automatically the User-CSI Agent to the AMP that contains the CSI that Agent is looking for.

Finally, ADOCSI does not meet criterion M7. ADOCSI does not take into consideration the reasons for the revocation of a certificate, when performing the certificate path validation function. This could be a subject for future work on ADOCSI (see section 5).

In conclusion, ADOCSI is a CSI dissemination mechanism that is rich in features. However, that does not impose any restrictions on the way the CAs, the dependent entities or the certificate holders work. Dependent entities and certificate holders do not need to have a profound understanding of the mechanics of ADOCSI and do not have to disrupt their normal operation, whatever that may be, in order to locate, retrieve and use CSI.

4.4.2 Efficiency

We now evaluate ADOCSI against the second set of criteria (efficiency of mechanism) we have developed in section 3.1. These are the following:

E1. Timeliness of CSI,

E2. Freshness of CSI,

E3. Bounded revocation,

E4. Emergency CSI capability,

E5. Economy,

E6. Scalability,

E7. Adjustability.

The Online Certificate Status Protocol seems to be the most efficient mechanism (see Table 10), out of those presented in section 2. However, it does not meet the criteria of economy and adjustability. As far as economy is concerned, the use of OCSP requires that the dependent entities locate and retrieve CSI for every certificate they wish to verify, separately. Furthermore, in OCSP, CAs are not required to have a common location for their CSI, a location that is easily found if the location of the CA itself is known. Finally, the dependent entities must know which access protocol to use in order to retrieve CSI from the OCSP repository. CRLs are also not economic (see evaluation of CRLs at section 3.2.1). However, CRLs and other proprietary mechanisms that make use of the same functionality offered by OCSP are the most commonly used CSI dissemination mechanisms in practice.

	Efficiency of Mechanism						
CSI dissemination mechanism	E1	E2	E3	E4	E5	E6	E7
CRL			√				
FCRL	√	√	√	√			√
RCRL			√			√	
Enhanced CRL Dist. Options			√			√	
Positive CSI	√?	√		√	√		
CRS		√	√			√	
OCSP	√?	√	√	√		√	
Freshness constrained RevA		√	√	√			√
ADOCSI	√	√	√?	√	√	√	√

Table 10: Evaluation of CSI mechanisms based on the efficiency of the mechanism

The cost for implementing and deploying the base infrastructure for the ADOCSI mechanism is probably higher than for any other mechanism. However, once ADOCSI is deployed, it can prove to be a very economic mechanism, especially as far as the dependent entities and the certificate holders are concerned. Neither of them has to have a profound understanding of ADOCSI in order to use it. Furthermore, neither of them has to disrupt their

normal working practice in order to locate and retrieve the necessary CSI. CSI is delivered to the dependent entities transparently.

The only overhead in the normal operation of CAs is that they may have to produce CSI in more than one formats or with more than one mechanisms (see mechanisms in section 2). Part of the efficiency of ADOCSI lies in the fact that it can take advantage of more than one CSI formats, offering thus adjustable (criterion E7) CSI to the dependent entities. Once this is done, the CAs are relieved from the responsibility of the CSI dissemination, and at the same time they can ensure that the correct CSI will reach the dependent entities, thanks to the security scheme that protects the operation of the CA-CSI Agents (see section 4.3). Therefore, economy (criterion E5) is one of the strong points of ADOCSI.

ADOCSI also meets the freshness and timeliness criteria. The dependent entities can state their preferences, either in their custom applications or to the Interface Agents, regarding the freshness of the CSI they wish to receive and the way they wish the CSI delivery to be performed (see sections 4.2.2 and 4.2.1). Moreover, the dependent entities can specify different levels of CSI freshness, depending on the importance of the certificates they are verifying, on the resources they wish to avail to the CSI retrieval for a specific group of certificates and on the consequences of not having timely CSI on a specific group of certificates (see sections 4.2.2 and 4.2.1). Therefore, ADOCSI also meets the adjustability criterion. The only other mechanism, out of those reviewed in section 2, which supports adjustable levels of freshness and timely delivery is the FCRL (see Table 10). However, that mechanism does not support fine granularity in the aforementioned levels; CSI can either be fresh or not. On the contrary, the dependent entities can specify with precision how fresh do they want the CSI to be. Certainly, the User-CSI Agents will retrieve CSI that meets these specifications to the best possible extent. It may not be possible to meet perfectly those CSI specifications always, due to the restrictions imposed on CAs by the revocation mechanisms they use internally. However, if the CAs use a variety of such mechanisms, that offer respectively a variety of freshness for the issued CSI, then ADOCSI does support effectively a fine granularity for the adjustability levels.

ADOCSI also meets the emergency CSI criterion. If emergency CSI is issued, it can be sent either directly to the CA-CSI Agents that already reside at the AMPs (e.g. using the CSI format described in section 2.2 or the one described in section 2.8), or new CA-CSI Agents can be issued that contain themselves the emergency CSI. In any case, the emergency CSI will be available for retrieval at the AMPs, on time. The dependent entities (or the Interface Agents that control the use the User-CSI Agents of the dependent entities) can choose specific certificates or group of certificates for which they wish to have emergency CSI. These certificates might be high in the hierarchy of certificates that the dependent entities use, therefore revocation of those certificates could result in revocation of many certificates the dependent entities use. If the preferences of dependent entities (which are stored at the local repositories of the dependent entities) indicate that emergency CSI on certain certificates should be made available to them, then the Interface Agents will instruct the User-CSI Agents to look for such information when they roam the network and transport themselves to AMPs. In conclusion, ADOCSI meets the emergency CSI criterion, but at the same time it minimises the respective operational costs this might entail, since the User-CSI Agents do not look for and retrieve emergency CSI on all certificates but only on those the dependent entities specify.

ADOCSI could support bounded revocation, if all the mechanisms used by the CAs in order to generate CSI (see section 2) support bounded revocation. Since the only mechanism that does not support bounded revocation is “Positive CSI” (see section 3.2.5 and Table 10), then if CAs do not use that mechanism, ADOCSI can also support bounded revocation. However, if this mechanism is used for the generation of CSI, ADOCSI cannot support bounded revocation, therefore ADOCSI supports the bounded revocation criterion (E3) partially.

The scalability (criterion E6) of ADOCSI depends on a number of factors, such as the number of the available AMPs, the number of Agents each AMP can host and the number of AMPs to which the CAs send their CA-CSI Agents. Assuming that there are sufficient AMPs, therefore we have sufficient storage space and processing capability, and assuming that the CAs do replicate their CA-CSI Agents to more than one AMP, then ADOCSI meets the scalability criterion.

One of the major advantages of ADOCSI over the other CSI dissemination mechanisms is that ADOCSI has a combination of online and offline features. The User-CSI Agents can be transported from the dependent entity to an AMP that is currently available. If that Agent needs to visit another AMP in order to retrieve CSI and that other AMP is not available at that time, the Agent can stay at the first AMP until the other AMP is available again. Certainly, the amount of time an Agent can stay at AMP, in wait for another AMP to become available, should have an upper limit in order to avert Denial of Service attacks. If entities send a large number of Agents to an AMP, looking for CSI that is known to be available at another AMP that currently is not available for some reason, then these Agents could take up all the available storage at the first AMP, thus provoking a Denial of Service for other Agents that would like to visit the aforementioned AMP.

The fact that the User-CSI Agent can stay at an AMP for a predefined amount of time before transporting itself to the next destination can prove to be beneficial for the dependent entities as well. A dependent entity may disconnect itself from the network or network connectivity between the dependent entity and an AMP may fail temporarily. If a User-CSI Agent has gathered all the CSI it has been sent to retrieve and attempts to return to the dependent entity at a time that the latter is not available for some reason, the User-CSI Agent can remain at the AMP where it is currently, until the dependent entity connects back to the network or until the network connectivity is restored. Therefore, ADOCSI meets the scalability criterion, if the number of the available AMPs is sufficient.

In conclusion, ADOCSI is a more efficient CSI dissemination mechanism compared to the ones we have presented in section 2. Although the set of features it supports is larger than the respective sets of other mechanisms (see section 4.4.1), it is still an economic mechanism. Furthermore, it provides for timely and adjustable dissemination of fresh CSI. Most of the other mechanisms that offer fresh CSI in a timely manner are either not economic or not scalable. Finally, ADOCSI is offering fine grained adjustability levels. The only other mechanism offering this is the Freshness constrained Revocation (see section 2.8), but at the cost of loosing economy and scalability.

4.4.3 Security

We now evaluate ADOCSI against the third set of criteria (security of mechanism) we have developed in section 3.1. These are the following:

S1. CSI disseminator authentication,

S2. CSI integrity,

S3. CA compromise,

S4. RevA compromise,

S5. Contained functionality,

S6. Availability.

CSI retrieval occurs within the execution boundaries of the AMP. User-CSI Agents contact with CA-CSI Agents in order to locate the CSI they look for and retrieve it. Authentication of

the CA-CSI Agents is achieved by verifying the digital signature CAs have put on these Agents. User-CSI Agents must carry with them the respective CA certificates in order to verify the aforementioned digital signatures (see section 4.3.1). Furthermore, authentication of the AMPs where User-CSI Agents reside and execute can be performed, if the dependent entities encrypt those Agents with the public key of the AMP they are going to send them to (see section 4.3.3). Therefore, criterion S1 is met.

Verifying the integrity of the CSI is being performed by the User-CSI Agents and by the Interface Agent, once the former arrive at the dependent entity, by verifying the CA signature on the CSI Therefore, criterion S2 is also met.

The effects of having a CA key compromised depend directly on the method used by the CA to generate the CSI, and therefore of the CSI format they CA-CSI Agents carry. Taking into consideration that CA-CSI Agents would probably carry CSI in more than one formats, in order to accommodate for the varying needs and requirements of the dependent entities, we assert that ADOCSI does not meet criterion S3 about CA compromise. If a CA private key is compromised, then the entity that holds the compromised key will be able to issue new certificates and revoke old ones, if the dependent entities are not aware of the CA key compromise.

All of the mechanisms we have evaluated in section 3.2 meet criterion S4 at least partially. Since ADOCSI is a mechanism that does not perform revocation in itself, but uses the CSI produced by other mechanisms and disseminates it, then ADOCSI also meets partially criterion S4. Compromising an authority that actually performs certificate revocation and produces the respective CSI, would have the same effect if ADOCSI was also used for CSI dissemination or not.

	Security of Mechanism					
CSI dissemination mechanism	S1	S2	S3	S4	S5	S6
CRL	√	√		√?	√?	√
FCRL	√	√		√?	√?	√
RCRL	√	√		√?	√?	√
Enhanced CRL Dist. Options	√	√		√?	√?	√
Positive CSI	√?	√?	√?	√?	√?	√
CRS		√	√		√	√
OCSP	√	√		√	√	√?
Freshness constrained RevA	√	√	√	√?		
ADOCSI	√	√		√?	√?	√

Table 11: Evaluation of CSI mechanisms based on the security of the mechanism

CAs should produce CSI in more than one formats (see section 2) and install that CSI in their CA-CSI Agents. If this is done, then those Agents could verify the validity of CSI they have in one format (e.g. CRL) based on the respective CSI in another format (e.g. CRS). An entity should compromise all the RevAs that ADOCSI is using in order to generate CSI (e.g. CRL and CRS), in order to be able to issue new certificates as well. Therefore, ADOCSI meets criterion S5, at least partially, as well.

The availability of ADOCSI depends directly on the number of operating AMPs and the number of AMPs the CAs send their CA-CSI Agents to. If there are enough AMPs and the CAs send their Agents to a sufficient number of AMPs, then the availability of ADOCSI is

ensured. The cost for this availability is the frequent communication of the AMPs with the CAs.

In conclusion, ADOCSI also meets, at least partially, most of the security criteria we have developed in section 3.1.3. We should mention, though, that ADOCSI faces more threats than the other mechanisms, since it is the only mechanism that involves the transport and remote execution of applications (see section 4.3). Research on Agent Security [FIPA98c], [Necu98] and its relation with PKI [He98] [Thiru95] is currently being performed. It is possible that some of the security issues [Wil98], [Wil99] that relate to Agents and have not been solved effectively until now, will be solved in the near future.

5. FUTURE WORK

Protecting Agents from AMPs is a still a subject of research [Wil98], [Wil99]. Perhaps the security countermeasures that have been used in order to protect the AMPs from the Agents' actions can be used here as well. If the CAs could verify that the DF Agent code, as well as the code of the rest of the static Agents of the AMP has specific safety properties[Necu96], [Necu98], or if the CAs themselves could check for these properties [Gong97], then the CAs could sign the aforementioned Agents, if they meet certain criteria. Therefore, CAs could protect Agents from malicious AMPs, since the actions of an AMP would be predictable.

A dependent entity Agent might not want to let a CA-CSI Agent or an AMP know for which CSI the User-CSI Agent is looking for, unless the CA-CSI Agent does contain that CSI. A dependent entity Agent might want that, in order to protect his privacy (at least from the AMPs or CA-CSI Agents that do not have the CSI that the dependent entity is looking for). If that is achieved, then the risk of having a compromised AMP or CA-CSI Agent giving false CSI to User-CSI Agents could be reduced as well, because the AMP or CA-CSI Agent will not know for which CSI the User-Agent is looking for, unless the AMP contains such CSI. One way to achieve that [Riord98] would be to form the CSI query in the following way:

Query: H(N, CertID, CaCertID),

where H() is a hash function, and CertID is an identifier for the certificate the dependent entity is looking CSI for. It could be the public key of that certificate, the hash of that public key, the hash of the serial number of the certificate and the respective Distinguished Name [X.500] contained in the certificate, or any other construct that can identify uniquely a certificate issued by a specific CA. Furthermore, the query should include another certificate identifier, for the CA that issued the certificate that the dependent entity is looking CSI for. Finally, in order to avoid dictionary attacks from an AMP that would want to find out for which certificate is the dependent entity looking CSI for, a nonce (N) should be included as well. This nonce could be the nonce included in the Agent in order to protect the entities that receive the Agent from replay attacks (see section 4.3.5).

The query is formed from the hash of the aforementioned information. The User-CSI Agent will be retrieving the respective certificate identifier values from the CA-CSI Agent and will be calculating the respective query values until he finds one that matches. If this is the case, the User-CSI Agent may retrieve that CSI from the CA-CSI Agent at that time. If the CA-CSI Agent or all the CA-CSI Agents that reside in the AMP, do not contain the CSI that the User-CSI is looking for, then the latter will communicate with another CA-CSI Agent, or will transport itself to another AMP. However, neither the CA-CSI Agents that the User-CSI Agent communicated with, nor the AMP the User-CSI Agent was residing in, will be able to know for which certificate was the Agent looking for.

Neither the CSI mechanisms we have reviewed in section 2, nor ADOCSI, take into consideration the reasons for the revocation of a certificate when performing certificate path validation. Furthermore, during a certificate path validation, none of the aforementioned mechanisms takes into consideration certificates that contain the same public key but belong to other certification paths, and they have been revoked. Obviously, the aforementioned certificates should belong to the same entity, but it could also be the case of two or more entities that had -on purpose- the same key pair, certified by different CAs.

Assume that certificates C_k and C'_k belong to the respective certificate chains C and C', depicted in Table 12, and contain the same subject public key. Should C_k be revoked, that does not mean that C'_k will be revoked as well, as it belongs to another certificate chain. The fact that C_k has been revoked should affect C'_k in a different way, depending on the reason for the revocation of C_k [Fox98]. Currently, PKI does not support the revocation, or any other action

upon a subject's certificate, if another certificate containing the same public key is revoked. Let us examine the various reasons for the revocation of a certificate. These reasons can be included, optionally, as an extension to the CRL [ISO9594], [Hous99]. This extension is called CRLReason (see Table 13).

$C_1, C_2, \ldots C_{k-1}, C_k, \ldots C_n,$
$C'_1, C'_2, \ldots C'_{k-1}, C'_k, \ldots C'_n$

Table 12: Certificate chains

C_k could be revoked because of the following reasons [ISO11770], [Fox98]:

1. The public key contained in C_k should not be trusted anymore because it has been compromised. The CRL entry should contain the reason 'keyCompromise'. In this case, C'_k should be revoked as well, because it contains the same, compromised public key.

2. The binding between the public key contained in C_k and the subject name contained in C_k should no longer be trusted, because the affiliation of the certificate holder has changed, there is a new certificate that supersedes C_k or because the certificate holder will not be using anymore C_k. The respective reason codes that should be used in the CRL entry are: 'affiliationChanged', 'superseded' and 'CessationOfOperation'. If C'_k - contains information that bind the person denoted by the subject name in C'_k to the same affiliation as is the case with C_k then C'_k should be revoked as well. If C_k was revoked because it was superseded then C'_k might not need to be revoked. Finally, if C_k was revoked because the certificate holder will not be using that certificate anymore, then C'_k should be revoked or should not be revoked depending on the reason why the certificate holder will not be using C'_k anymore.

3. C_{k-1} (certificate holder k-1 is the issuer of certificate C_k) is no longer a position to vouch for the validity of C_k. This would be due to a compromise of the key contained in C_{k-1}. The reason code that should be used in the CRL entry is: 'cACompromise'. In this case, C'_k should not be revoked, because C'_{k-1} (and every certificate higher in the hierarchy of C') is not affected because of the compromise of C_{k-1}.

We have demonstrated that there are cases when the revocation of a certificate belonging to one certificate chain may provoke the revocation of a certificate that belongs to another certificate chain but bears the same public key. However, the current, supported certificate path validation mechanisms and CSI location mechanisms do not support such actions. Even if such actions were supported buy the aforementioned mechanisms, there would still be questions to be answered regarding the revocation of one certificate on another certificate that contains the same public key. One of the issues that would have to be tackled with, is what would happen in case a certificate needs to be revoked for a reason that is not included in the list of reasons proposed by [ISO9594], [Hous99]. It could be the case that a CA (the holder of C_{k-1}) has realised that an entity (the holder of C_k) has been misidentified, either because of a procedural registration error or because the entity has provided false identification [ISO11770] to the CA during the registration phase. In this case, none of the existing, proposed reasons to be included in the CRL entry do not concern the real reason for the revocation. Another issue that would have to be tackled with, is the actions to be taken with respect to C'_k in case C_k appears in a CRL with the value certificateHold in the entry extension reasonCode. A certificate could be suspended (certificateHold) for many, different reasons and it would not be clear in this case what actions should be taken either by the entities that own the certificates in the certification

chain $C^{'}$ or by the dependent entities that wish to use certificates (e.g. $C^{'}_{k}$, C_k) that belong to these chains.

ADOCSI could possibly provide the means to deal with the aforementioned problems, since the Agents involved in ADOCSI do have access to a common pool of CSI resources (the set of all AMPs and CA-CSI Agents). User-CSI Agents could seek the necessary information in this pool in order to verify the validity of a certificate, even if the same public key is contained in more than one certification paths. In addition to that, the dependent entity could customise the User-CSI Agents (through a graphical user interface provided by the Interface Agent) in order to let them know the preferences of the dependent entity regarding the revocation of certificates, depending on the reasons other -relevant- certificates have been revoked. Clearly, this subject has to be studied further.

6. CONCLUSIONS

We reviewed and evaluated the CSI dissemination mechanisms that had been proposed to the time of writing of this document. Clearly, each mechanism presents its own advantages and disadvantages and therefore a different CSI dissemination mechanism could be selected by the CAs, depending on the application area of the PKI.

We also developed an alternative mechanism for the dissemination of certificate status information (ADOCSI), which presents some enhancements, compared to the other mechanisms that have been proposed. One of the most important enhancements is that this mechanism meets the transparency criterion (see section 3.1.1). The entities involved with the use of the security schemes that PKI offers are not always knowledgeable, motivated and perceptive towards security procedures (see section 4). Therefore, such a mechanism could be of use, since it relieves these entities from the burden of carrying out security-related tasks in order to locate and retrieve Certificate Status Information. In addition to that, new features can be implemented in the operation of this CSI dissemination mechanism in a transparent way. New features could be developed and integrated in the mechanism (e.g. in the CA-CSI Agents and the User-CSI Agents), that possibly serve the needs of the dependent entities in a better way. However, the dependent entities themselves need not know about the mechanics of these new features.

ADOCSI presents other advantages as well. The network traffic that results from CSI queries can be reduced, since the queries take place on a local level, inside the AMPs. Furthermore, ADOCSI is resilient to unreliable networks, because it has both online and offline features.

Most of the existing CSI dissemination mechanisms restrict the Certification Authorities in the way they must operate. CAs cannot re-partition the CSI space, or transport their CSI repositories elsewhere, because the dependent entities will not be able to track those changes easily. ADOCSI lifts that restriction. CAs can re-partition their CSI space, transport the CSI repositories and also maintain CSI in a number of different formats. Such actions would result in confusion of the dependent entities and would create problems in the certification process, if a CSI mechanism out of those presented in section 2 was used. On the contrary, if ADOCSI is used as a CSI dissemination mechanism, such action on behalf of the CAs would result in even more efficient CSI dissemination.

ADOCSI provides for adjustable CSI. The dependent entities can retrieve the CSI that meets their own requirements, and at the same time without understanding the mechanics of the CSI dissemination process and without the dependent entities having to act themselves or contribute in any other way to the aforementioned process. The only other mechanism offering fine-grained adjustability in CSI location and retrieval is the Freshness constrained Revocation (see section 2.8), but at the cost of loosing economy and scalability.

ADOCSI is a more efficient CSI dissemination mechanism compared to the ones we have presented in section 2. Although the set of features it supports is larger than the respective sets of other mechanisms (see section 4.4.1), it is still an economic and scalable mechanism. Furthermore, it provides for timely dissemination of fresh CSI. Most of the other mechanisms that offer fresh CSI in a timely manner are either not economic or not scalable (see Table 10).

One of the disadvantages of ADOCSI is that it assumes that there is sufficient number of collaborating AMPs and that each AMP contains a sufficient number of CA-CSI Agents. Furthermore, ADOCSI assumes that all Agents involved in the scheme are using the same Agent Communication Language. Although these assumptions do not pose any problem at all, on a technical level, there might be other obstacles in implementing such a CSI scheme. There might be commercial CAs that do will refuse to have their CA-CSI Agents in the same AMP as another CA, or in the AMP developed by an organisation that is somehow related with another CA. Furthermore, if a CA has a large user base (entities that have been issued certificates from

that CA) that CA may refuse to serve CSI requests from User-CSI Agents developed by other CAs. In general, real-life implementation of ADOCSI requires that CAs will collaborate, in order to deliver CSI. This might not be always feasible, at least not in the commercial sector.

Another disadvantage of ADOCSI is that it requires that the Trusted Computing Base (TCB) [Goll99] of the dependent entities and CAs grows in size. The User-CSI Agent and the Interface Agent are part of the TCB of the dependent entity. These Agents are locating, retrieving and possibly validating CSI on behalf of the dependent entity. CA-CSI Agents also form part of the new TCB of CAs. CA-CSI Agents are the ones that disseminate the available CSI, not the CAs, anymore. Taking into consideration that security issues related with the use of Agents still exist (see section 4.3), such a growth of the Trusted Computing Bases requires that these problems have to be solved, in an economic way, before using Agents in the CSI process.

Generation and dissemination of Certificate Status Information has to be studied further. There are problems related with all the CSI mechanisms (see section 5), which still have not been solved. These problems could require the sharing of CSI among a wide group of CAs and the processing of that CSI before communicating it to the dependent entities. We believe that Agent-based CSI mechanisms could possibly provide the means to deal with the aforementioned problems since the Agents involved in such mechanisms have access to a common pool of CSI resources, the set of all collaborating AMPs and the respective, resident CA-CSI Agents.

In this dissertation, we have evaluated the proposed mechanisms for generating and disseminating Certificate Status Information. Also, we have identified the disadvantages that the use of each of the aforementioned mechanisms presents. Finally, we have developed a prototype of an Agent-based mechanism for disseminating CSI, which we believe offers certain improvements in CSI dissemination, compared to the other mechanisms.

References

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

[Aglets] IBM Aglet Workbench, available at http://www.trl.ibm.co.jp/aglets

[Berne94] Berners-Lee T., Universal Resource Identifiers in WWW, IETF Network Working Group, Request for Comments 1630, Category: Informational, June 1994, available at http://www.ietf.org/rfc/rfc1630.txt

[Chess95] Chess D., Grosof B., Harison C., Levine D., Parris C., , IBM Research Division, T.J. Watson Research Center, New York, & Tsudik G., IBM Zurich Research Laboratory, Itinerant Agents for Mobile Computing, Research Report 20010, 1995

[Corige95] ISO/IEC JTC 1/SC 21, Technical Corrigendum 2 to ISO/IEC 9594-8: 1990 & 1993 (1995:E), July 1995

[Denni82] Denning D. E., Cryptography and Data Security, Addison-Wesley, 1982

[Dramn96] ISO/IEC JTC 1/SC 21, Draft Amendments DAM 4 to ISO/IEC 9594-2, DAM 2 to ISO/IEC 9594-6, DAM 1 to ISO/IEC 9594-7, and DAM 1 to ISO/IEC 9594-8 on Certificate Extensions, June 30, 1996

[Euro98a] European Commission, Communication from the Commission to the European Parliament, the Council, the Economic and Social Committee and the Committee of the Regions, Proposal for a European Parliament and Council Directive on a common framework for electronic signatures, COM(1998)297final, 13 May 1998

[Euro98b] European Commission, Proposal for a European Parliament and Council Directive on a common framework for electronic signatures, Official Journal of the European Communities, COM(1998) 297 final - 98/0191(COD), 21 October 1998

[FIPA98a] Agent Management, Part 1, Foundation for Intelligent Physical Agents Specification, Geneva, October 1998

[FIPA98b] Agent Communication Language Version 2.0, Part 2, Foundation for Intelligent Physical Agents Specification, Geneva, October 1998

[FIPA98c] Agent Security Management Version 1.0, Part 10, Foundation for Intelligent Physical Agents Specification, Geneva, October 1998

[Ford94] Ford W.,Computer Communications Security, Prentice-Hall, 1994

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

[Frost96] Frost H. R., Cutkosky M. R., Design for Manufacturability via Agent Interaction, In Proceedings of the ASME Computers in Engineering Conference, Irvine, CA, August 1996

[Gene92] Genesereth M. R., An Agent-Based Approach to Software Interoperability, In Proceedings of the DARPA Software Technology Conference, 1992

[Gene92b] Genesereth M., Fikes R., et al., Knowledge Interchange Format, version 3.0, reference manual, Technical Report, Computer Science Department, Stanford University, 1992

[Gene94] Genesereth M. R., Ketchpel S. P., Software Agents, In Communications of the ACM 37 (7), 1994

[Goll99] Gollman D., Computer Security, John Wiley & Sons, 1999

[Gong97] Gong L., Mueller M., Prafullchandra H., Schemers R., Going Beyond the Sandbox: An Overview of the New Security Architecture in the Java Development

Kit 1.2, In Proceedings of the USENIX Symposium on Internet Technologies, 1997

[Gong98] Gong L., Signing, Sealing and Guarding Java Objects, Lecture Notes in Computer Science (LNCS), Vol.1419, Springer-Verlag, June 1998

[Gorri98] Gorrieri R., Marchetti R., Applet Watch-Dog: A Monitor Controlling the Execution of Java Applets, Proceedings of the IFIP SEC'98, published by Austrian Computer Society, 1998

[Gritz97] Gritzalis, S., Spinellis, D. Addressing Threats and Security Issues in World Wide Web Technology, In Proceedings of the 3rd IFIP International Conference on Communications and Multimedia Security, Chapman & Hall, 1997

[Gritz98] Gritzalis S., Iliadis J., Addressing security issues in programming languages for mobile code, In Proceedings of the DEXA '98 9th Workshop of Database and Expert Systems Applications, IEEE Computer Society Press, August 1998

[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

[He98] He Q., Sycara P., Finin T. W., Personal Security Agent: KQML-Based PKI, In Proceedings of the ACM Conference on Autonomous Agents (Agents'98), May 1998

[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

[ISO11770] ISO/IEC JTC 1/SC 27/WG 1 11770-1.2, Information Technology - Security techniques - Key management - Part 1: Framework, 1996

[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]

[Jacl97] Jacl and Tcl Blend, (1997) available at http://sunscript.sun.com/java/

[KQML] ARPA Knowledge Sharing Initiative, External Interfaces Working Group, Specification of the KQML agent-communication language, working paper, July 1993

[Lamb97] Lambrou Y., Finin T., A Proposal for a new KQML Specification, Technical Report CS-97-03, University of Maryland, Baltimore, US, 1997

[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

[Necu96] Necula G.C., Lee P., Safe Kernel extensions without run-time checking, Proceedings of the Second Symposium on Operating Systems Design and Implementation, 1996

[Necu98] Necula G. C., Compiling with Proofs, PhD Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, September 1998

[Nwana96] Nwana H. S., Software Agents: An Overview, In Knowledge Engineering Review, Cambridge University Press, 1996

[Odyssey] General Magic Odyssey, available at http://www.genmagic.com/agents

[Petrie97] Petrie C. J., What's an agent… and what's so intelligent about it?, IEEE Internet Computing Vol. 1, Number 4, July-August 1997

[Riord98] Riordan J, Schneier B., Environmental Key Generation towards Clueless Agents, In Proceedings of Mobile Agents and Security, LNCS, Springer-Verlag, 1998

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

[Schn96] Schneier B., Applied Cryptography, John Wiley & Sons, 1996

[Spruit96] Spruit M. E. M., Looijen M., IT security in Dutch practice, Computers & Security, 15-2, 157, 1996

[Spruit98] Spruit M. E. M., Competing against human failing, In Proceedings of the XV IFIP World Computer Congress: 14th IFIPSEC '98 International Information Security Conference, September 1998

[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

[Thiru95] Thirunavukkarasu C., Finin T., Mayfield J., Secret Agents - A Security Architecture for the KQML Agent Communication Language, In Proceedings of CIKM'95 Intelligent Information Agents Workshop, December 1995

[Wil98] Wilhelm U. G., Staamann S., On the Problem of Trust in Mobile Agent Systems, In Symposium on Network and Distributed System Security, Internet Society, March 1998

[Wil99] A Technical Approach to Privacy Based on Mobile Agents Protected by Tamper-resistant Hardware, PhD Thesis, Wilhelm U. G., Department of Informatics, EPFL, Lausanne, 1999

[X.500] CCITT Recommendations X.500-X.521, Data Communication Networks Directory, CCITT, November 1988

[X.509] ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1997

[Zhang97] X.N.Zhang, Secure Code Distribution, IEEE Computer, June 1997

Appendix

Table 13: Certificate and CRL Extensions

X.509v3 Certificate Extension	cRLDistributionPoints ::= {
	CRLDistPointsSyntax }
	
	CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint
	
	DistributionPoint ::= SEQUENCE {
	distributionPoint [0] DistributionPointName OPTIONAL,
	reasons [1] ReasonFlags OPTIONAL,
	cRLIssuer [2] GeneralNames OPTIONAL }
	
	DistributionPointName ::= CHOICE {
	fullName [0] GeneralNames,
	nameRelativeToCRLIssuer [1] RelativeDistinguishedName }
	
	ReasonFlags ::= BIT STRING {
	unused (0),
	keyCompromise (1),
	cACompromise (2),
	affiliationChanged (3),
	superseded (4),
	cessationOfOperation (5),
	certificateHold (6) }

X.509v2 CRL Extension	issuingDistributionPoint ::= SEQUENCE { distributionPoint [0] DistributionPointName OPTIONAL, onlyContainsUserCerts [1] BOOLEAN DEFAULT FALSE, onlyContainsCACerts [2] BOOLEAN DEFAULT FALSE, onlySomeReasons [3] ReasonFlags OPTIONAL, indirectCRL [4] BOOLEAN DEFAULT FALSE }

Table 13: Certificate and CRL Extensions (cont.)

X.509v2 CRL	CertificateList ::= SEQUENCE {
	tbsCertList TBSCertList,
	signatureAlgorithm AlgorithmIdentifier,
	signatureValue BIT STRING }
	
	TBSCertList ::= SEQUENCE {
	version Version OPTIONAL,
	-- if present, shall be v2
	signature AlgorithmIdentifier,
	issuer Name,
	thisUpdate Time,
	nextUpdate Time OPTIONAL,
	revokedCertificates SEQUENCE OF SEQUENCE {
	userCertificate CertificateSerialNumber,
	revocationDate Time,
	crlEntryExtensions Extensions OPTIONAL
	-- if present, shall be v2
	} OPTIONAL,
	crlExtensions [0] EXPLICIT Extensions OPTIONAL
	-- if present, shall be v2
	}

X.509v2 CRL Extension	`DeltaCRLIndicator ::= BaseCRLNumber`
	
	`BaseCRLNumber ::= CRLNumber`

X.509v2 CRL Entry Extension	`CRLReason ::= ENUMERATED {`
	`unspecified (0),`
	`keyCompromise (1),`
	`cACompromise (2),`
	`affiliationChanged (3),`
	`superseded (4),`
	`cessationOfOperation (5),`
	`certificateHold (6),`
	`removeFromCRL (8) }`

Table 14: Enhanced CRL Distribution options

StatusReferrals CRL extension	statusReferrals EXTENSION ::= {
	SYNTAX StatusReferrals
	IDENTIFIED BY <oid tbd> }
	StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral
	
	StatusReferral ::= CHOICE {
	cRLReferral [0] CRLReferral,
	DeltaReferral [1] DeltaReferral,
	oCSPReferral [2] OCSPReferral}
	
	CRLReferral ::= SEQUENCE {
	issuer GeneralName OPTIONAL,
	location GeneralName OPTIONAL,
	DeltaLocation GeneralName OPTIONAL,
	cRLScope CRLScopeSyntax,
	lastUpdate GeneralizedTime OPTIONAL,
	lastDelta GeneralizedTime OPTIONAL}
	
	DeltaReferral ::= GeneralName
	
	OCSPReferral ::= SEQUENCE {
	issuer GeneralName OPTIONAL,
	location GeneralName OPTIONAL }

cRLScope CRL extension	cRLScope EXTENSION ::= {
	SYNTAX CRLScopeSyntax
	IDENTIFIED BY { <oid tbd> } }
	
	CRLScopeSyntax ::= SEQUENCE SIZE (1..MAX) OF PerCAScope
	
	PerCAScope ::= SEQUENCE {
	cAName [0] GeneralName OPTIONAL,
	distributionPoint [1] DistributionPointName OPTIONAL,
	onlyContainsUserCerts [2] BOOLEAN DEFAULT FALSE,
	onlyContainsCACerts [3] BOOLEAN DEFAULT FALSE,
	onlySomeReasons [4] ReasonFlags OPTIONAL,
	serialNumberRange [5] NumberRange OPTIONAL,
	subjectKeyIdRange [6] NumberRange OPTIONAL,
	nameSubtrees [7] GeneralNames OPTIONAL
	}
	
	NumberRange ::= SEQUENCE {
	startingNumber INTEGER,
	endingNumber INTEGER,
	modulus INTEGER OPTIONAL }

Table 15: OCSP referral to CRL

Extension in OCSP responses, referring to a relevant CRL	CrlID ::= SEQUENCE { crlUrl [0] EXPLICIT IA5String OPTIONAL, crlNum [1] EXPLICIT INTEGER OPTIONAL, crlTime [2] EXPLICIT GeneralizedTime OPTIONAL } crlUrl is the pointer to the CRL, crlNum is the Number extension of the specific CRL and crlTime is the time when the specific CRL was issued (also exists in the CRL as an extension)

Table 16: Certificate identification in OCSP

Part of the OCSP request message. These values are used to identify the certificate for which a dependent entity asks for CSI from the entity that offers the OCSP service.	CertID ::= SEQUENCE { hashAlgorithm AlgorithmIdentifier, issuerNameHash OCTET STRING, -- Hash of Issuer's DN issuerKeyHash OCTET STRING, -- Hash of Issuers public key serialNumber CertificateSerialNumber }

Table 17: Key Usage

The use of a public key contained in a certificate (and the respective private) is defined in this X.509v3 certificate extension	keyUsage EXTENSION ::= {
	SYNTAX KeyUsage
	IDENTIFIED BY id-ce-keyUsage }
	
	KeyUsage ::= BIT STRING {
	digitalSignature (0),
	nonRepudiation (1),
	keyEncipherment (2),
	dataEncipherment (3),
	keyAgreement (4),
	keyCertSign (5),
	cRLSign (6) }

Glossary

ACL	Agent Communication Language
ADOCSI	Alternative Dissemination of Certificate Status Information
AMP	Agent Meeting Point
AP	
AP	Agent Platform
CA	Certification Authority
CPU	Central Processing Unit
CRL	Certificate Revocation List
CRS	Certificate Revocation Status
CSI	Certificate Status Information
DF	Directory Facilitator
DoS	Denial of Service
FCRL	Freshest CRL
FIPA	Foundation for Intelligent Physical Agents
FRIP	Freshest Revocation Information Pointer
IA	Interface Agent
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
MAC	Message Authentication Code
OCSP	Online Certificate Status Protocol
PKI	Public Key Infrastructure
RA	Registration Authority
RCRL	Redirect CRL
RevA	Revocation Authority
RPC	Remote Procedure Call
SB	Suicide Bureau
SN	Suicide Note
TCB	Trusted Computing Base
TTP	Trusted Third Parties
URI	Universal Resource Identifier
URL	Uniform Resource Locator
WAN	Wide Area Network