Proceedings of the 2005 IEEE
Workshop on Information Assurance and Security
United States Military Academy, West Point, NY

A New On-line Certificate Validation Method using LDAP Component Matching Technology

Jong Hyuk Choi, Sang Seok Lim, and Kurt D. Zeilenga

Abstract— **This paper presents a new on-line certificate validation method which provides higher degree of security, scalability, and interoperability than do the pre-existing approaches. It combines two basic data structures for certificate revocation, Certificate Revocation List (CRL) and the authenticated dictionary such as Certificate Revocation Tree (CRT), into a single framework by utilizing the component matching enabled Lightweight Directory Access Protocol (LDAP) service. With the new method, end entities that want to check the validity of certificates can request an extended LDAP search operation with a component matching assertion against all revoked certificate components in a CRL and check whether a revoked certificate having the asserted serial number is found. In order to ensure strong security without requiring trusted directories, CRLs are represented as an authenticated dictionary when decoded from Distinguished Encoding Rules (DER) to an internal ASN.1 representation. The information required to construct the authenticated dictionary is conveyed from the Certificate Authority (CA) via a new CRL extension. The proposed method facilitates a number of advantages over the previous approaches like Online Certificate Status Protocol (OCSP): 1) it enables higher security because it does not require trusted entities other than the CA such as trusted LDAP servers and trusted OCSP responders; 2) it improves scalability and performance because it does not require responses to be signed as in OCSP; 3) it can interoperate well with the existing CRL framework; and 4) it does not need support for additional protocols for on-line certificate validation because it is built on LDAP which is the main access method to download CRLs. The proposed method can also be used as a CRL back-end of OCSP to offload CRL management and to enhance its trust model.**

I. Introduction

The scalability of Public Key Infrastructure (PKI) can be significantly limited by the certificate revocation mechanism it employs as evidenced by the large operational costs for certificate revocation described in the MITRE report [1] on the PKI for the Federal Government. This report focused on the analysis of revocation cost when Certificate Revocation List (CRL) is used to periodically disseminate revocation information to end entities via untrusted certificate repositories such as LDAP directories. Many remedies have been proposed to relieve the scalability requirements of the CRL mechanism. The CRL distribution points, delta-CRL, and freshest-CRL are examples of such remedies. Version 2 CRLs enable these schemes to be used by introducing the notion of extensions in CRL.

CRLs permit certificates to be validated while end entities are off-line since the end entities can perform searching for the certificate in question against the list of revoked certificates in the base and the delta-CRLs in local storage once downloading them from a certificate repository. While this can be considered adequate for the conventional end entities such as applications running in desktop computers and servers, it can be suboptimal in many cases.

First, constrained end entities such as mobile phones and PDAs tend to have very limited storage, computing power, and network bandwidth. It should be burdensome for such devices to store CRLs in its limited amount of storage and to perform checking for the presence of the certificate in question against the locally stored CRLs. Although delta-CRLs allow incremental download of CRLs, it often contains excess information beyond the necessary information to validate a certificate. This can be costly in a bandwidth constrained setup.

Second, delta-CRLs do not handle large-size revocations efficiently. Large size revocations may be unavoidable when a large scale organizational changes occur or when a Certificate Authority (CA) is compromised. The requirement of timely disseminating such a large amount of revocation information would degenerate the scalability of the delta-CRLs scheme close to that of the original CRL approach.

On-line certificate status checking protocols have been proposed to enable real-time certificate validation to provide information on a given certificate in a timely manner for those end entities which are expected to be always on-line and to provide certificate validation service for those end entities having constrained storage, computing power, and / or network bandwidth. However, the pre-existing protocols for on-line certificate validation are considered to be short of being complete in terms of security and scalability. Unlike CRLs, they require an additional trusted entity such as a protocol responder while CRL requires only one trusted entity, the CA. Moreover, while CRL is a pre-signed entity which does not require on-the-fly signing upon responding to end entities, the protocol responses of the pre-existing on-line certificate validation protocols needs be digitally signed every time they are returned in order to make the response unforgeable. This signing requirement can significantly degrade the scalability of such on-line certificate validation responders. To remedy this

Jong Hyuk Choi: IBM T .J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, USA. jongchoi@us.ibm.com
Sang Seok Lim: IBM T .J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 USA. slim@us.ibm.com
Kurt D. Zeilenga: IBM Linux Technology Center, Carson City, NV USA. zeilenga@us.ibm.com

0-7803-9290-6/05/$20.00 ©2005 IEEE.

scalability limitation, it became a practice to cache pre-signed certificate validation responses or to return unsigned responses especially in large scale deployments, sacrificing some level of security over scalability.

In this paper, we propose a new on-line certificate validation method which utilizes the LDAP component matching technology we designed and implemented in our previous works [2] together with the authenticated data structures for attestation of the LDAP responses on the presence of the queried certificate in the CRL. The proposed on-line certificate validation method enables checking for the presence of a certificate in the sequence of the revoked certificates contained in a CRL by using the LDAP component matching mechanism. In component matching, Distinguished Encoding Rules (DER) encoded CRL attributes are decoded into the internal ASN.1 representation for matching against a Generic String Encoding Rules (GSER) encoded assertion value for the portion of the CRL located by the component reference in the component search filter. In order to provide a proof on the authenticity and integrity of the LDAP search result, the ASN.1 value of a CRL is represented as an authenticated dictionary such as the Certificate Revocation Tree (CRT).

The advantage of the proposed mechanism over the previous on-line certificate validation approaches is manifold.

- Much higher level of security can be maintained because the proposed method does not require a new type of trusted entities other than the CA. Fewer types of end entities that have to be trusted leads to improved level of security and to the reduced level of complexity in protocol design.
- Scalability of the responder will be significantly improved since the proposed method does not mandate the response to be signed to provide a proof of authenticity and integrity of the response. Instead, they are provided by a hierarchical authenticated data structure which can attest the presence of a certificate in a CRL by providing the value of the relevant CRL entry in the leaf level and the values of the internal nodes that together can reconstruct the signature at the root.
- Response time of the on-line certificate validation process will be improved by much because of the elimination of the signing step from the response generation path.
- Interoperability with the existing CRL certificate revocation framework can be guaranteed since the proposed method utilizes the existing infrastructure based on the extension mechanism of the X.509 Version 2 CRL and based on the extension mechanism of the LDAP Version 3 controls. As a result, the CRL framework can be used for both on-line and off-line certificate validation.
- The proposed method does not need support for additional protocols dedicated for on-line certificate validation, since it uses the same LDAP which is the main access method for certificates and CRLs.
- When used as a CRL back-end of the responders of the existing on-line certificate revocation methods, it is possible to eliminate or reduce the real time signing requirements of the responses when the authenticity and integrity proof is forwarded to the end entities instead of returning responses signed by the responders.

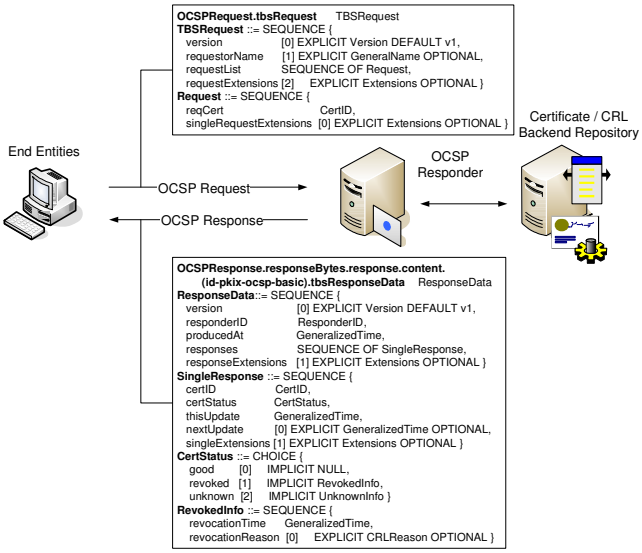

Fig. 1. Online Certificate Status Protocol.

This paper will describe the operations of the proposed on-line certificate validation method in detail. The rest of the paper is organized as follows. Section 2 will describe the previous on-line certificate validation protocols and discuss their weaknesses. Section 3 will present how the component matching technology enables certificate validation by searching for CRL components. In Section 4, it will be discussed how to provide a strong proof of authenticity and integrity without needing trusted directories and without the need for real time signing of responses. Section 5 will summarize the paper.

II. On-Line Certificate Status Checking

As the first step towards a standardized on-line certificate validation protocol, the Online Certificate Status Protocol (OCSP) [3] defines a simple request-response protocol between an OCSP client and an OCSP responder as shown in Fig. 1. The OCSP responder is a trusted entity that responds with the information whether the certificates in a request are revoked, and if so, when and why they were revoked. As shown in the figure, an OCSP responder can contact various backends such as certificate / CRL repositories to retrieve revocation information.

Fig. 1 also shows the key components of the ASN.1 specification of the OCSP request and response as defined in [3]. An OCSP request is an optionally signed sequence mainly consisting of a sequence of certificate identifiers. A certificate identifier consists of a certificate's serial number and the proofs of the certificate issuer's identity. An OCSP response is a signed sequence mainly consisting of a sequence of *SingleResponse*s. A *SingleReponse* indicates whether the status of the queried certificate is *good*, *revoked*, or *unknown*. If it was previously *revoked*, the time and the reason of revocation are also provided to the client. The *thisUpdate*

```
(userCertificate:componentFilterMatch:=
  and:{
    item:{
      component "toBeSigned.subject",
      rule distinguishedNameMatch,
      value "cn=foo,dc=example,dc=com"
    }
    item:{
      component "toBeSigned.extension.*.
                          extnValue.(2.5.29.15)",
      rule bitStringMatch,
      value '100000000'B
    }
  }
)
```

Fig. 2. Certificate Component Matching.

and *nextUpdate* values provide the validity interval as an indication whether and how long the revocation information can be locally used by the client. Both the OCSP request and the OCSP response are equipped with optional extension fields for protocol extensibility.

The requirement for the OCSP responder to be a trusted entity is a serious fall back from the original trust model. PKI requires only the CAs to be trusted in the entire infrastructure. It significantly increases the chances of being compromised to have extra protocol entities that must be trusted in addition to the CA. Moreover, every OCSP response must be signed to make it unforgeable. The response signing significantly degrades the performance and scalability of certificate validation because of the increase in latency and decrease in throughput. The response pre-production scheme described in the OCSP proposed standard [3] cannot be considered as a highly secure solution unless the cached responses contain some binding information such as *nonce* in order to prevent replay attacks. If an OCSP responder cannot service a large number of certificate validation requests because of the security requirement inherent in the OCSP, it cannot be considered as a scalable protocol which is essential for the scalable implementation of PKI.

III. CRL Component Matching

A. Component Matching and Its Usage

Component matching is recently published in RFC 3687 [4] in an effort to provide a complete solution to the LDAP - PKI interoperability problem.

The certificate / CRL syntaxes of X.509 are defined in ASN.1 types. The type is structurally constructed from basic types to composite types. Every field of an ASN.1 type is a component. All attribute syntaxes of X.500 and LDAP are originally described in ASN.1 type specifications [5], [6]. ASN.1 types are encoded by using ASN.1 encodings such as BER, DER, and PER which keep the structural information of the specification. However, LDAP uses LDAP specific encodings which do not generally preserve the structural information in the original ASN.1 type, instead of relying on an ASN.1 encodings. With an ASN.1 encodings, it is easy to perform matching against arbitrary components of a certificate / CRL since ASN.1 provides a generic and flexible framework for data representation and matching in a heterogeneous environment. Good examples showing the power of ASN.1 are the versatile matching rules specified in X.509 [7] such as *certificateMatch* and *certificateListMatch*. With LDAP specific string-based encodings, the versatility and flexibility in data representation and matching can only be achieved by elaborate coding of LDAP syntax decoding and matching routines. This process has to be repeated for the definition of every new type such as in X.509 protocol extensions. Previous remedy to this mismatch in the PKI - LDAP interoperability was to extract the components of certificate / CRL and store them in separate searchable attributes. The component extraction process can be automated as described in [8]. Instead of performing searches on the certificate / CRL themselves, clients are directed to perform searches on the newly defined set of attributes which contains the values of the extracted components of the certificate / CRL.

The component matching can eliminate the need for the burdensome processes of extracting and synchronizing the searchable components of Certificate / CRL and of altering the Directory Information Tree (DIT) structure to host the extracted components in a separate entry which would have been required if those components were extracted a priori and therefore became mutable. With component matching, CRL is stored in the Distinguished Encoding Rules (DER) [9] and decoded on the fly into the internal Abstract Syntax Notation One (ASN.1) [10] representation for matching. Necessary decoders and component management routines are automatically created by the use of an ASN.1 compiler in OpenLDAP as described in our previous works [2]. Clients can make search requests by using a component filter in which the asserted component is specified by following the ASN.1 type structure and the asserted value is specified using a string-based ASN.1 encoding rules, Generic String Encoding Rules (GSER) [11].

Based on ASN.1 types, component matching [4] defines how to refer to a component within an attribute value and how to match the referred component against an assertion value. Matching rules are defined for the ASN.1 basic and composite types. It also defines a new assertion and filter tailored for a component, or each field of the ASN.1 type. These definitions are based on ASN.1 so that they can be applied to any complex syntax, as long as it is specified in ASN.1. The component matching defines a generic way of enabling matching user selected components of an attribute value by introducing a new notion of component assertion, component filter, and matching rules for components. With component matching, it becomes possible to perform matching of an assertion value against a specific component of a composite attribute value. For example, infrastructure is provided to perform matching against an arbitrary component of an X.509 certificate, such as *serialNumber*, *issuer*, *subject*, and *keyUsage*.

The search filter for component matching is a matching rule assertion [6] whose matching rule is *componentFilterMatch* and whose assertion value is a component filter. A component filter is a combination of component assertions each consisting of the following three parts:

- *Component Reference*: specifies which component of the

```
CertificateList ::= SIGNED { SEQUENCE {
   version             Version OPTIONAL,
   signature           AlgorithmIdentifier,
   issuer              Name,
   thisUpdate          Time,
   nextUpdate          Time OPTIONAL,
   revokedCertificates SEQUENCE OF SEQUENCE {
      serialNumber       CertificateSerialNumber,
      revocationDate     Time,
      crlEntryExtensions Extensions OPTIONAL } OPTIONAL,
   crlExtensions   [0] Extensions OPTIONAL }}
```

(a) Certificate Revocation List.

```
( certificateRevocationList:
    componentFilterMatch:=
      item:{ component "toBeSigned.
        revokedCertificates.*.serialNumber",
      rule integerMatch, value 12345 } )
```

(b) LDAP Filter for CRL Component Matching.

Fig. 3. CRL Component Matching.

attribute value will be matched against the assertion value.

- *Matching Rule*: specifies which matching rule will be used to perform matching on the value.
- *Value*: An assertion value in GSER.

Fig. 2 shows an example LDAP search filter for Certificate component matching. The component filter is the assertion value of the LDAP search filter. The component filter consists of two component assertions whose results are intersected. The first component assertion is an assertion on the value of the *subject* component of a certificate while the second component assertion is an assertion on the value of the *keyUsage* extension of a certificate. The component filter can be translated to "Match the certificates of *subject* value of *cn=foo,dc=example,dc=com* but only whose *keyUsage* is *digitalSignature*.

Compared to the attribute extraction approach, component matching has the following advantages:

1) It does not extract and store certificate / CRL components separate from the certificate / CRL themselves. Therefore, it does not increase storage requirements and does not open a potential to the compromised integrity between a certificate and its extracted attributes.
2) Matching is performed not on the extracted attributes' contents but directly on the certificate's content. It can return only the matched certificate out of multiple certificates in a user's entry if it is used in conjunction with the matched values control [12].
3) It becomes convenient to provide a complex matching flexibly because matching between attribute and assertion values is performed at the ASN.1 layer.

B. Certificate Revocation List (CRL) Access

We conceive that component matching enabled LDAP can also be used as an on-line certificate validation protocol. Fig. 3 (a) shows the ASN.1 specification of CRL as defined in the X.509 recommendation [7]. CRL is a sequence of pairs of a revoked certificate's serial number and revoked time [13]. In order to check status of the certificate, the client needs to make a component assertion against the serial number of the certificate under scrutiny. Then, the LDAP server will perform component matching on the CRL against the assertion to find the asserted serial number in the CRL. This is possible with the component matching, since the LDAP server understands the structure of the CRL and is able to match specific components of the CRL against the component assertion. In the attribute extraction approach, however, the serial numbers of all the elements of the revoked certificate list must be extracted as separate attributes which need to be stored in the individual subordinate entries. This not only increases the amount of storage and increases the complexity of managing directory significantly, but also makes the server vulnerable to malicious attacks.

With component matching, the whole CRL does not necessarily have to be downloaded to the clients and scanned by the client so as to relieve the requirements for the storage, network bandwidth and the client's computing power significantly. Especially for the clients which have limited amount of storage, low bandwidth network connection, and low speed processor, such as mobile devices, component matching will be a very effective solution to enable the clients to access PKI efficiently. Furthermore, an LDAP server already has been widely used for distributing CRLs and certificates. Hence, if the server can perform on-line validity checking over the CRL as well, it will be a very practical and more efficient alternative to OCSP which has limitations in security and scalability. Or, alternatively, the CRL component matching enabled LDAP servers can be used as CRL back-end to the OCSP responders in order to offload the CRL management and searching from the OCSP responders.

Fig. 3 (b) shows an example CRL component search filter which queries the presence of the revoked certificate entry having serial number 12345 in the CRL base entry where the component reference, the string following `component`, specifies matching against all (represented by "$*$") components of the *revokedCertificate* of CRL which is of `SEQUENCE OF` ASN.1 type [7]. If a revoked certificate having serial number 12345 is present in the CRL, the certificate was previously revoked by the Certificate Authority (CA). Upon matching, the client could be returned the value of the CRL attribute in the DER format for additional verification of the presence of the revoked certificate in the CRL. Alternatively, only yes / no information about the presence of the searched-for certificate in a CRL can be returned via a special LDAPv3 control in the LDAP search response.

One disadvantage of the simple certificate validation through LDAP component matching of CRLs proposed in

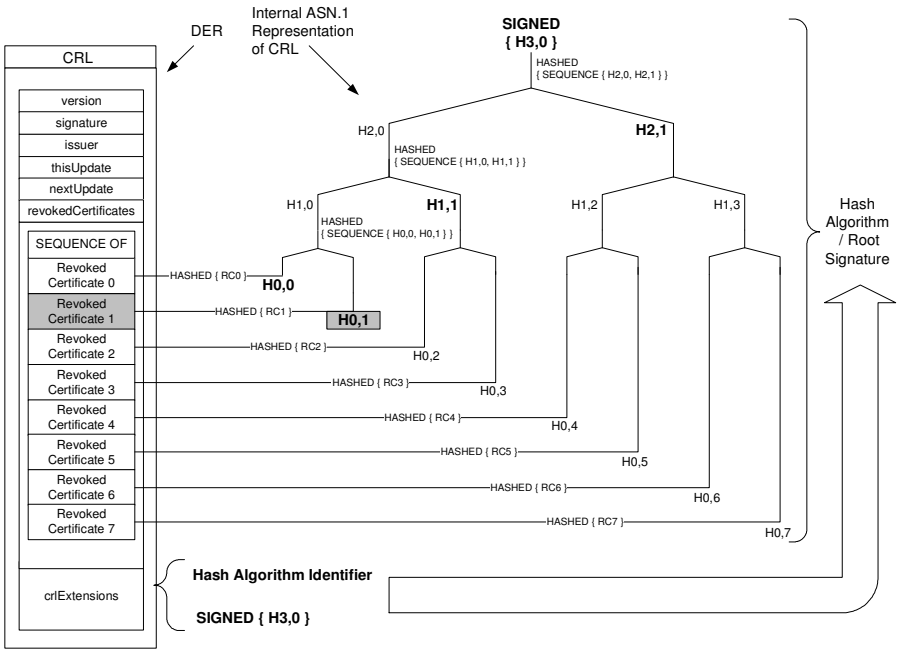

Fig. 4. Authenticated Dictionary for CRL.

this section is its weaker trust model than the original CRL download scheme. It is not possible to detect whether the LDAP directory server is compromised. If a compromised LDAP server answers that a certificate is not revoked, end entities will be exposed to security threats.

The next section will present a new mechanism proposed in this paper to make the CRL component matching secure by structuring the internal representation of CRL as an authenticated data structure. Together with the component matching, it makes certificate validation result from an LDAP server unforgeable while not requiring to have the LDAP server as a trusted entity nor to sign every LDAP response on the fly as in OCSP. If such CRL component matching enabled LDAP directories are used as CRL back-end of the OCSP responders, it is possible to relieve the requirements of having the OCSP responders as trusted entities and those of signing OCSP responses on-the-fly.

IV. Authenticated CRL Component Access

The component matching enabled LDAP CRL repository must be a trusted entity unless some form of proof is provided to attest authenticity and integrity of it responses. In this paper, we propose to use an authenticated dictionary data structure such as the Kocher's Certificate Revocation Trees (CRT) [14] and the Naor's Authenticated Search Data Structures [15] in the internal ASN.1 representation of CRL in the component matching enabled LDAP CRL repository in order to vouch LDAP results with the proof of authenticity and integrity. Each CRL (including delta-CRL) will be represented as the tree structured authenticated data structure when it is decoded from the DER encodings to its ASN.1 internal representation.

Fig. 4 illustrates a CRL containing eight CRL entries in it. When it is decoded from DER, it is represented as a binary

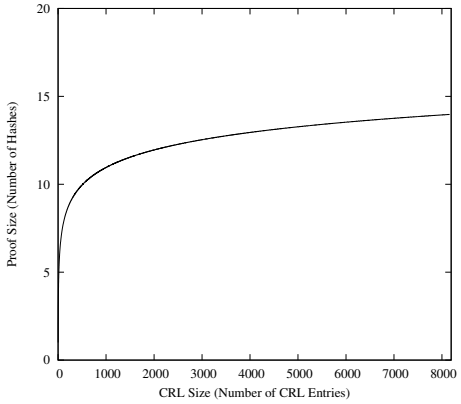

Fig. 5. Size of Proof bw. CRL Repository and End Entity.

hash tree whose leaf nodes correspond to the hash value of the identity of the CRL entries in the CRL. The value of a superior node in a tree is determined by hashing the combined values of its subordinate nodes. When the identity of the queried certificate matches that of the second CRL entry in the CRL as shown in Fig. 4, the minimum set of hash values required to calculate the hash value at the root are used along with the signature of the root hash value signed by the CA are used as the proof attesting that the queried certificate is revoked and no other part of the CRL was altered. In the example in Fig. 4, the proof consists of $H_{0,0}$, $H_{0,1}$, $H_{1,1}$, and $H_{2,1}$, together with the signature of the root hash value, $H_{3,0}$. If any part of the CRL internally stored in the LDAP CRL repository is compromised, end entities can easily detect it through signature verification using the public key of the CA.

In order to make the proposed mechanism operate within

the contexts of the X.509 PKI standard [7] and of LDAP, we rely on the extension mechanisms of Version 2 CRLs and of LDAP Version 3. The signature of the root hash value and the algorithm used to generate hash values can be conveyed from the CA to the LDAP CRL repository by using a new per-CRL extension of the Version 2 CRL. The signature and the hash values of the required internal nodes of the tree can be sent to the client as an operational attributes or by using the control mechanism of LDAPv3 [16].

The communication cost required to transmit the proof information from the CA to the LDAP CRL repository is fixed and does not increase as the number of CRL entries in a CRL increases. Only the signature of the root hash value (*SIGNED* $\{$ *Root Hash Value* $\}$) and the hashing algorithm identifier need be transmitted from the CA to the LDAP CRL repository. The communication cost required to transmit the proof from the LDAP CRL repository to the end entities when an on-line certificate validation response is sent to the end entities is logarithmic to the number of CRL entries in a CRL. When the cardinality of the hash tree is C, $(C-1)\log_C N+1$ hash values and one signed root hash value need be transmitted as the proof of authenticity and integrity of the LDAP component matching response. Fig. 5 shows that the size of proof between the LDAP CRL repository and end entities are small. The size of the proof is not large even for very large CRLs. With a binary hash tree, only 21 hash values are required for a CRL having a million CRL entries, which is far less than the size of the CRL itself.

V. Summary

As PKI becomes indispensable in ensuring security of increasing number of applications in e-Business and in emerging applications like Web Services, it becomes imperative to provide an effective means of disseminating certificate revocation information in a scalable and timely manner. The on-line certificate validation method proposed in this paper utilizes the LDAP component matching and the authenticated dictionary data structure to facilitate secure and scalable on-line certificate validation on top of the existing base of the certificate revocation via CRL. As the next step of this study, we are currently evaluating the performance of the proposed on-line certificate validation system in comparison with the existing on-line certificate validation schemes such as OCSP and we are defining the required extensions to the X.509 Version 2 CRL, the LDAP Version 3, and the OCSP toward standardization.

References

[1] MITRE Corporation, "Public key infrastructure final report." `http://csrc.nist.gov/pki/documents/mitre.ps`, 1994.

[2] S. S. Lim, J. H. Choi, and K. D. Zeilenga, "Design and implementation of LDAP component matching for flexible and secure certificate access in PKI," in *Proc. of the 4th Annual PKI R&D Workshop*, pp. 37–47, April 2005.

[3] M. Myers, R. Ankney, A. Malpani, and C. Adams, "Internet X.509 public key infrastructure online certificate status protocol - OCSP." RFC 2560, June 1999.

[4] S. Legg, "X.500 and LDAP component matching rules." RFC 3687, February 2004.

[5] ITU-T Rec., "X.500, The directory: Overview of concepts, models and service," February 2001.

[6] J. Hodges, R. Morgan, and M. Wahl, "Lightweight directory access protocol (v3): Technical specification." RFC 3377, September 2002.

[7] ITU-T Rec., "X.509, The Directory: Public-key and attribute certificate frameworks," March 2000.

[8] D. W. Chadwick and M. V. Sahalayev, "Internet X.509 public key infrastructure LDAP schema for X.509 CRLs." <draft-ietf-pkix-ldap-crl-schema-03.txt>, October 2004.

[9] ITU-T Rec., "X.690, ASN.1 encoding rules: Specification of basic encoding rules (BER), canonical encoding rules (CER), and distinguished encoding rules (DER)," 1994.

[10] ITU-T Rec., "X.680, Abstract syntax notation one (ASN.1): Specification of basic notation," December 1997.

[11] S. Legg, "Generic string encoding rules." RFC 3641, October 2003.

[12] D. W. Chadwick and S. Mullan, "Returning matched values with LDAPv3." RFC 3876, September 2004.

[13] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 public key infrastructure certificate and CRL profile." RFC 2459, January 1999.

[14] P. C. Kocher, "On certificate revocation and validation," in *Proc. of the 2nd Int'l Conference on Financial Cryptography (Lecture Notes in Computer Science, Vol. 1465)*, pp. 172–177, 1998.

[15] M. Naor and K. Nissim, "Certificate revocation and certificate update," in *Proc. of the 7th USENIX Security Symposium*, pp. 217–228, January 1998.

[16] M. Wahl, T. Howes, and S. Kille, "Lightweight directory access protocol (v3)." RFC 2251, December 1997.