SCIS 2002 The 2002 Symposium on
Cryptography and Information Security
Shirahama,Japan, Jan.29-Feb.1,2002
The Institute of Electronics,
Information and Communication Engineers

A New Approach of X.509v3 Certificate for Full Path Validation

Jaegwan Park [*]
jgpark@icu.ac.kr

Kwangjo Kim [*]
kkj@icu.ac.kr

Abstract— The Public Key Infrastructure (PKI) and Wireless PKI (WPKI) are essential to many kinds of electronic businesses through the Internet. A certificate and related mechanism such as the Certificate Revocation List (CRL), and the certificate path validation are important components in PKI and WPKI. However, because a certificate does not contain its full path, a verifier must check the certificate revocation status and perform the certificate path validation, step by step. Even though a verifier finished to check the certificate revocation status and perform the certificate path validation, a verifier can only know the probabilistic answer about target certificate, because a CRL is published in every periodic time. In this paper, we will propose a new approach of X.509v3 certificate for full path validation. Using our proposed scheme, we can reduce the time complexity of the certificate path validation from $O(n)$ to $O(1)$, when n is the size of the certification full path. In addition, using our proposed scheme, we will show an application, the Online Certificate Verification Protocol (OCVP), which neither requires the CRL mechanism nor a new trusted server. With respect to the computational load, the loads in OCVP is $2n$ which is the same in the Simple Certificate Validation Protocol (SCVP). However, SCVP uses the CRL mechanism and a new trusted server, and gives us a probabilistic answer. But, OCVP uses all CAs who are located on the certification full path and gives us an exact answer.

Keywords: Certificate, Serial Number, Certificate Path Validation, Certificate Revocation List (CRL), Online Certificate Status Protocol (OCSP), Simple Certificate Validation Protocol (SCVP), Online Certificate Verification Protocol (OCVP)

1 Introduction

With the advance of the Internet, many businesses through the Internet such as M-Commerce, B2B, and B2E, are increasing. The Public Key Infrastructure (PKI) and Wireless PKI (WPKI) are essential to many kinds of electronic businesses through the Internet. PKI consists of a certificate, a certificate authority, an end-entity, a directory for the revoked certificates, cryptographic algorithms (i.e., public key cryptosystem, signature), and some computing mechanisms.

In particular, the certificate is a document for authentication that binds between the public key and the subject who holds the corresponding private key. Therefore, it is very important to verify the certificate in PKI and WPKI.

The certificate is available only during its time-period specified in the contents [4], [6]. However, sometimes the certificate is revoked because of loss of the private key, attacking by some viruses or hackers, changing of the subject name, and so on. In this case, the certificate authority (CA) must construct a list for revoked certificates, and publish it. This black-list is called the certification revocation list (CRL).

Therefore, when a user receives a certificate, for the first time, he has to check whether it is revoked or not. But, CRL is larger than other PKI components with respect to its size. And the management cost of CRL is more expensive than other PKI components [12]. Even though CRL has these problems, it is widely used. As noted in [11], many efforts to improve or modify CRL are proposed.

One of these efforts is the Online Certificate Status Protocol (OCSP) [7] and the other is the Simple Certificate Verification Protocol (SCVP) [8]. These two are request and response protocols. The former is used only for the certificate revocation checking. The latter is used for more general usage. But, both of them require a new trusted server.

After the certificate revocation checking, a process for certificate path validation is proceeded by the receiver or the server that he/she can trust. But, because the certificate does not contain the certification full path, the receiver should verify certificates step by step. Therefore, when the size of certification path is n and the verification time is t, the time complexity is $O(2nt) \approx O(n)$.

In this paper, we propose a new approach of X.509v3 certificate for full path validation. Using our proposed scheme, we can reduce the time complexity of the certificate path validation from $O(n)$ to $O(1)$. In addition, using our proposed scheme, we will show an application, the on-line certificate verification protocol (OCVP), without using the CRL mechanism and a new trust server. With respect to the computational load, the loads in OCVP is $2n$ which is same in SCVP. However, SCVP uses the CRL mechanism and a new trusted server, and gives us a probabilistic answer. But, OCVP uses all CAs who are located on the certification full path and gives us an exact answer.

This paper is organized as follows: Section 2 reviews related works such as a certificate, CRL, certificate path validation, OCSP, and SCVP. We propose a new approach of X.509v3 certificate for full path validation, and describe what is changed in the certificate path validation using our proposed scheme in Section 3. And, using our proposed scheme,we show an application, OCVP, in Section 4. Concluding remarks will follow in Section 5.

[*] International Research center for Information Security (IRIS), Information and Communications Univ. (ICU), 58-4 Hwaam-dong, Yuseoung-gu, Daejon, 305-732, Korea

2 Related Works

2.1 Certificate

A certificate is a document that binds the public key and its subject who holds the related private key. Only if a certificate is issued by a legal CA, it should have the validity. The goal of the certificate path validation is to check this validity which will be described more detail in Section 2.3.

X.509v1 has been further developed into X.509v2 and X.509v3 in order to overcome weaknesses in the earlier versions (now, the version 4 is drafted). And X.509v3 is the basis of the IETF PKIX working group which aims at developing a general purpose public key certification infrastructure for the Internet.

A certificate in [4] contains as the following fields:

Version
Serial Number
Signature Algorithm identifier
Issuer name
Period of validity
Subject name
Subject's public-key information
Issuer unique identifier
Subject unique identifier
Extension
Signature

Table 1: Data format of X.509v3

2.2 Certificate Revocation List

All certificates have a field, the time period for its validity. In this predetermined period, the certificate is valid. But, sometimes a subject could lose the control for his certificate, because of loss of the related private key, attacking by some viruses or hackers, changing subject name, and so on. When he can not control his certificate, he must revoke his certificate.

In this case, the CA who issued the revoked certificate should construct a list of revoked certificates and sign the list. This signed black list is CRL. CRL is published periodically by each CA.

But, according to [12], the maintaining cost of CRL is the highest among PKI components. Furthermore, CRL has some problems. The followings are examples:

- *Heavy communication load*: CRL is published by CAs and contains all revoked certificates information. A user who wants to know the certificate revocation status - revoked or not - must download CRL from the directory or a CA, and then find the information which he wants. But, the size of CRL is too heavy to download easily.
- *Inefficiency of finding information that a user wants*: This problem results from the size of CRL. When a user wants to find something from CRL, it is not efficient to find from CRL.
- *Periodic publication*: CRL is published periodically. How long and how often? One hour or one day. In the Internet, one day or one hour is enough time to cause trouble. This problem is also emerged in the Delta CRL [9] which has shorter periodic time than CRL.
- *Negative response*: From CRLs, we can only know the exact information when a certificates is revoked.

To solve these problems, as noted in [11], many efforts to improve or modify CRL are proposed such as CRL Distribution Point [9], Delta CRL [9], and Over-issued CRL [10].

2.3 Certificate Path Validation

In order for two users to verify the authenticity of each other's public key, it is sufficient that there exists a certification path between them. A certification path is an ordered sequence of certificates which can be processed to obtain the public key contained in the final certificate in the path, together with the public key of the initial certificate in the path[13].

Housley *et al.* [4] describe about the certificate path validation. The certificate path validation is a process of binding between the subject name and its public key, and/or verify whether the certificate is legally issued or not.

Regardless of a certificate revocation status - revoked or not-, the result of the certificate path validation is true if a legal CA issued a certificate. Therefore, we must confirm a certificate revocation status through some mechanisms such as CRL before processing the certificate path validation.

We will use the following notions to describe the certificate path validation.

- $< x, y >$: x issued y's certificate
  When x issued his own certificate, namely, x's certificate is self-signed, the notion is $< x, x >$.
- $Cert_x$: a certificate of x
- CA_x: a CA, x
- U_i: a user, i
- V_j: a verifier, j
- $\{< x_1, x_1 >, < x_1, x_2 >, ..., < x_{n-1}, x_n >\}$: full path for certificate chain, and x_n is a user in the last $< x_{n-1}, x_n >$. The first $< x_1, x_1 >$ is self-signed relation.

And we will explain under the following assumptions.

- U_B sends $Cert_{U_B}$ to V_A.
- The size of certification full path is n.
- And the certification full path of $Cert_{U_B}$ is $\{< CA_1, CA_1 >, < CA_1, CA_2 >, ..., < CA_{n-1}, U_B >\}$.

After receiving $Cert_{U_B}$, V_A can know who issues $Cert_{U_B}$. Fig. 1 is a diagram for the certificate path validation and Fig. 2 is an algorithm of the certificate path validation processed by V_A.

Assuming that the computation time of *verify* $< x, y >$ and *CRL check* is t, and the downloading time for CRL and every certificate can be ignored in Fig. 1, the time complexity is $O(2nt) \approx O(n)$. Because **for** statement in Fig. 2 must iterate n times at the worst case, the verification operates twice every time.

2.4 Online Certificate Status Protocol

OCSP [7] is a protocol that an OCSP Server responds to a user's request with a certificate revocation status, namely, whether a certificate is revoked or not.

A user sends a request that consists of protocol version, service request, target certificate identifier, and optional extensions to an OCSP server.

When an OCSP server receives a request from a user, the server checks the followings order:

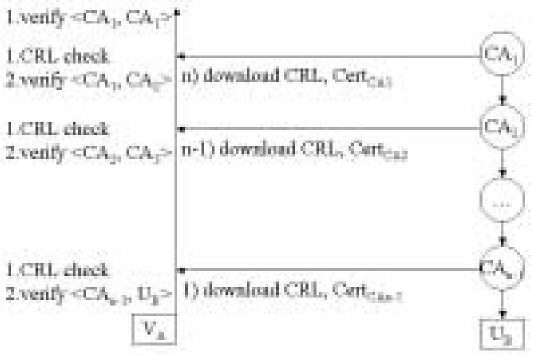

Fig. 1: Certificate path validation in RFC 2459

- The request is well formed.
- The server is configured to provided the request by himself.
- The request contains the information needed by himself.

If above three conditions are not met, the server responds an error message. Otherwise, he produces a response message and sends it to a requester. In a response message, the important information is a certificate status value. A certificate status value is one of the three status, - *good*, *revoked*, and *unknown*. First, *good* means that the target certificate is not revoked and second, *revoked* means that it is revoked. Finally, third, *unknown* means that the server doesn't know about the target certificate.

2.5 Simple Certificate Validation Protocol

SCVP [8] is the more general request/response protocol than OCSP. It deals with the full certificate verification process rather than only the certificate revocation checking as OCSP.

In SCVP, a user can construct more various requests message such as

- Build a certification path to a trusted root.
- Build a validated certification path to a trusted root.
- Do revocation status checks on the certification path.

OCSP only performs the last request, the revocation status checking.

3 Our Scheme

3.1 Problems and Assumptions

When a user wants to verify a received certificate, he/she must download CRLs, check the revocation status of the received certificate, and perform the process of the certificate path validation. These all processes are described in Section 2. As mentioned in Section 2.3, the time complexity of the certificate path validation is $O(n)$, when the size of certification path is n. This computational load is too heavy to be performed by a user. For this problem, some mechanisms such as OCSP, and SCVP, are proposed.

Another problem in the general certificate verification process is that we can not know the exact status at that time. Because CRL is published in periodically and the answer of the certification path validation is positive regardless of the revocation status. Therefore, in the general method, we do verify the certificate with the probability that the certificate is not revoked during the last periodic term.

```
Start-sub = $U_B$ ;
Start-cert = $Cert_{U_B}$ ;
Start-CA = Extract issuer from Start-cert ;

**for**(; Start-sub $\neq$ Start-CA ;){
   Start-CRL = Download CRL issued by Start-CA ;
   CA-cert = Download $Cert_{Start-CA}$ ;
   **if**( Start-cert in Start-CRL = Yes and
        verify the sign in Start-CRL = No){
     return fail ;
   }
   **if**( verify <Start-CA, Start-sub> = No){
     return fail ;
   }
   **if**( trust Start-CA = Yes) {
     return success ;
   }
   Start-sub := Start-CA
   Start-cert := CA-cert
   Start-CA := Extract issuer from Start-cert
}

**if**( trust Start-CA = Yes) {
   return success ;
}
return fail ;
```

Fig. 2: Certificate path validation algorithm in RFC 2459

Let's consider the followings:

- A CRL is published every predetermined periodic δ time by every CA.
- A general certificate is revoked under the probability ϵ in δ.
- CA's certificate is revoked under the probability ϕ in δ.
- The size of certification full path is n.
- A time to check the signature in a CRL is $t_{CRL} \approx t$.
- A time to verify one relation between a CA and a user ($< CA, user >$) or two CAs ($< CA, CA >$) is $t_{verify} \approx t$.
- The searching time in CRL and the downloading time can be ignored.

In [12], the revocation rate is defined 'the probability that any single entity's key will be revoked'. However, in this paper, we divide the rate into two categories according to the holder. A user (or server) who wants to verify a certificate can know the result after the following efforts.

- Checking n times of CRL, total time is $n \times t_{CRL}$.
- Verification n times of relations, total time is $n \times t_{verify}$.
- the probability that the target certificate is not revoked is $1 - \epsilon$

- the probability that CA's certificate is not revoked is $1-\phi$, the number of CA is $n-1$. Therefore, the probability that all CAs on the certification full path are not revoked is $(1-\phi)^{n-1}$.

It takes $n \times (t_{CRL} + t_{verify}) = n \times (2t) = 2nt$ time under the probability of $(1-\phi)^{n-1} \times (1-\epsilon)$ for verifying a certificate.

In this Section, we propose a new approach of X.509v3 certificate for full path validation. And, we describe what is changed in the certificate path validation using our proposed scheme.

Our proposed scheme has the following assumptions:

(A1) CA can have at maximum 255 sub CA's.

(A2) The certificate hierarchy has a strict structure.

(A3) At least, a user can believe on every root CA.

(A4) The number of certificates issued by a CA is less than 16, at the same time .

(A1) comes from the limitation of each CA's ID field. Both (A2) and (A3) are the most basic concept for current PKI. If CA's capability is over (A4) and/or a CA can have more sub CAs than (A1), we can modify the field of CA's ID and/or time.

3.2 Format of serial number

The format of our proposed scheme consists of three part. The first part is the *flag part* which has the information for length. The second is the *time field* which is recorded the issued time by an issuer. The last part is the *CA's ID* that records the unique name of all CAs who are located on the certification full path.

- *flag*: The first bit represents who is the holder of the certificate, a CA or a general user. If the first bit is 1, the holder is CA. Otherwise, the holder is a general user. The four bits from 2nd to 5th are the number of related CAs.

1	4	3
holder	number of CA's ID part	000

- *time field*: The length for time is 40 bits. The first 4 bits are the order of certificate issued at the same second.

4	6	6	5	5	4	10
	second	minute	hour	date	month	year

- *CA's ID*: The size of this part is $8 \times depth\ of\ hierarchy$ bits. The first 8 bits represent the leaf CA's ID, and the last 8 bits represent the root CA's ID.

8	...	8
the leaf CA's ID	...	root CA's ID

The total size of our proposed scheme is
$6\ bytes + 1\ byte \times depth\ of\ the\ certification\ hierarchy$.

3.3 How to make the serial number of a certificate

We consider the following two cases with respect to the types of a holder of a certificate. The first case is that a certificate is only used by a CA. The second case is that a certificate is used generally by a user.

1. *When the holder is a CA.*
   Suppose that CA_A is issuing a certificate of CA_B. CA_A separates each part from his own certificate. In the flag part, the first bit is set with 1, and add 1 to the number of CA's ID part of his own certificate. In the time field, put the issued time. In the CA's ID part, copy the third part of CA_A's certificate, generate a new ID for CA_B according to CA_B's policy, and concatenate the new CA ID to old one of CA_A. Finally, let's combine these three parts.

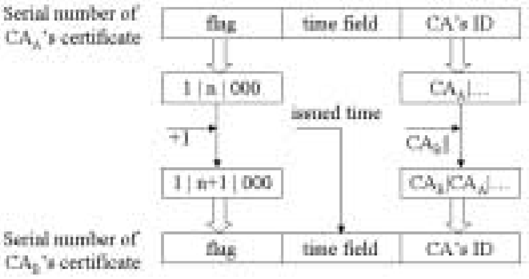

Fig. 3: Serial number format of a certificate, when the holder is a CA.

2. *When the holder is a general user.*
   Suppose that CA_A is issuing a certificate of U_B. CA_A separates the each part from his own certificate. Copy the first and third parts, namely, the flag part and the CA's ID part, with his own certificate. And, change the first bit of the flag part with 0. Put the issued time into the second part. Finally, let's combine these all.

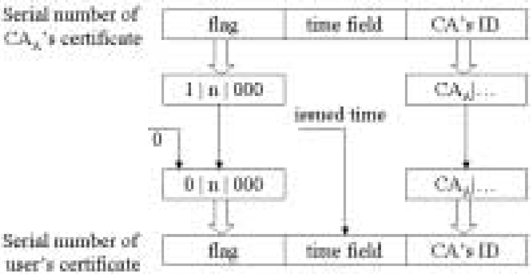

Fig. 4: Serial number format of a certificate, when the holder is a general user.

3.4 Certificate Path Validation

When V_A receives $Cert_{U_B}$, he compare $Cert_{V_A}$ with $Cert_{U_B}$. Because a certificate contains all CA's IDs who related to issue a certificate, V_A can know all CAs that he can not trust and the last subordinate CA that he can trust.

V_A downloads every mistrust CA's certificate, and related CRL at a time. And then, he can check all CRLs, and verify all download certificates and $Cert_{U_B}$ simultaneously. Fig. 5 and Fig. 6 are a certificate path validation diagram and algorithm, respectively, when the certificate is used our proposed scheme.

In section 2.3, the time complexity using general method is $O(n)$ and the process can not be adapted to a parallel mechanism. Because a verifier can not know each issuer until receiving a related certificate. Therefore, all processes are accomplished step by step.

But, using the certificate made by our proposed scheme, a verifier can know all CAs who relate to issue a certificate, and find the last subordinate CA which he can trust. He can download all information at a time due to this reason.

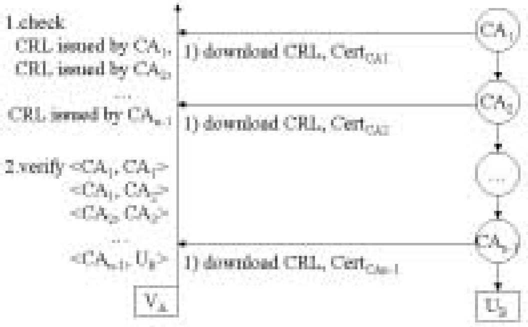

Fig. 5: Certificate path validation using our proposed scheme

```
My-sub = $U_A$ ;
My-cert = $Cert_{U_A}$ ;
Start-sub = $U_B$ ;
Start-cert = $Cert_{U_B}$ ;
CA[ ] = compare My-cert with Start-cert ;
CA-cert[ ] = download all certificate of CA[ ] ;
CA-CRL[ ] = download all CRL issued by CA[ ] ;

if( Start-cert and CA-cert[ ] in CA-CRL[ ] = Yes
      and verify the sign in CA-CRL[ ] = No){
   return fail ;
}
if( verify <CA[ ], Start-sub> and
      all <CA[ ], CA[ ]> = No){
   return fail ;
}
return success ;
```

Fig. 6: Certificate path validation algorithm using our proposed scheme

And then, all processes can be done in parallel. Therefore, in Fig. 6, if we have same assumptions in Section 2.3, the time complexity is a constant, $O(2t) \approx O(1)$.

4 Applications

As mentioned in Section 3.1, we get the probabilistic answer in the general verification method. This probabilistic answer is not suitable for a big contract. Therefore, we propose a protocol that is suitable for a big contract without a new trust server and the CRL processing.

4.1 Online Certificate Verification Protocol

OCVP is a request and response protocol. But OCVP uses all CA who are related to issue a certificate that a user want to verify, instead of a new trusted server.

For description, the following assumptions and notions are used.

- Pr_x: a private key of x
- $Sig_{Pr_x}(M)$: sign message M with a private key Pr_x
- $h(M)$: hash message M
- the CA's ID part in the serial number of $Cert_{V_A}$ is CA_z, CA_y, CA_x, CA_w.
- the CA's ID part in the serial number of $Cert_{U_B}$ is $CA_a, CA_b, ..., CA_c, CA_x, CA_w$.

When V_A receives $Cert_{U_B}$, he can know all mistrust CAs $(CA_a, CA_b, ..., CA_c)$ and the last subordinate CA (CA_x) among CAs $(CA_a, CA_b, ..., CA_c, CA_x, CA_w)$ who relate to issue $Cert_{U_B}$. Then, he makes a request message as the following format, and sends it to CA_a who issued a received certificate.

Message 1 :
$Sn_{Cert_{V_A}}||Sn_{Cert_{U_B}}||Nonce_{V_A}$
$||Sig_{Pr_{V_A}}(h(Sn_{Cert_{V_A}}||Sn_{Cert_{U_B}}||Nonce_{V_A}))$

In message 1, $Sn_{Cert_{V_A}}$ is the serial number of $Cert_{V_A}$, $Sn_{Cert_{U_B}}$ is the serial number of $Cert_{U_B}$, and $Nonce_{V_A}$ is the nonce generated by V_A.

When CA_a receives a request message from V_A, check that he is one of common CAs which is trusted by both V_A and U_B or not. If, he is not a common CA, he constructs a new request message as the following format, and sends it to prior CA(CA_b).

Message 2 :
$Message1||CA's\ ID||Answer$
$||Sig_{Pr_{CA}}(h(Message1||Ansewer))$

In message 2, the answer is the result of checking $Sn_{Cert_{U_B}}$ or the previous CA's status, 'Yes' or 'No'. When $Sn_{Cert_{U_B}}$ or the previous CA's status is not revoked, the answer is 'Yes'. However, if the answer is 'No' in previous request message, the answer must be 'No'.

When he is a common CA, he makes a response, message 2, and sends it to V_A. Fig. 7 is a flow for the OCVP.

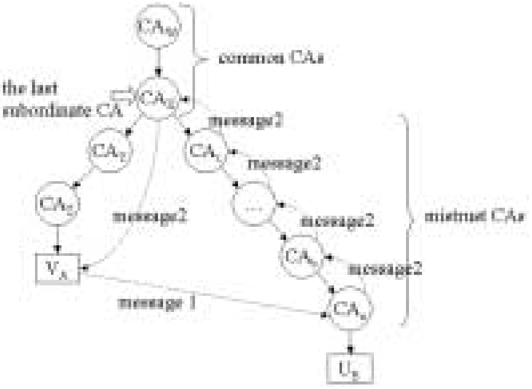

Fig. 7: Flow of the online certificate verification protocol

4.2 Comparison OCVP with CRL, OCSP, and SCVP

OCVP does not need to a new trusted server. And it does not use the CRL mechanism. Because it only uses all CAs who are located on the certificate full path. Due to this reason, the result of OCVP gives us the exact certificate status, unlike the general method.

We will compare OCVP with CRL, OCSP, and SCVP. Using CRL, the verifying process is performed by a user.

And, through CRL, a user can only know a certificate revocation status. Using OCSP, a user can only know a certificate revocation status, too. However, the CRL checking process is performed by a OCSP server.

Using SCVP, a user can know not only the certificate revocation status but also the certificate path validation under the probability. However, the process is performed by a SCVP server, like in OCSP.

	CRL	OCSP	SCVP	OCVP
need of new trusted server	X	O	O	X
usage of CRL	O	O	O	X
number of checking CRL	n	n	n	X
check of validation of certificate	X	X	O	O
number of signature	X	X	n	$2n$
total number of signature	n	n	$2n$	$2n$

Table 2: Comparison of OCVP with CRL, OCSP, and SCVP

Table 2 shows the comparison of OCVP with CRL, OCSP, and SCVP. In the table, the number of checking CRL is the number of verifying the signature when a user checks CRL.

The total number of signature in OCVP is same as that of SCVP. However, OCVP, neither uses the CRL mechanism nor require a new trusted server. Furthermore, as OCVP uses all CAs who are located on the certification full path, the result of this process is not probable.

5 Conclusion and Further Work

In this paper, we proposed a new approach of X.509v3 certificate for full path validation. Using our proposed scheme, we can reduce the time complexity of the certificate path validation. Because we can know all CAs who are located on the certification full path from the received certificate's serial number, we can download related CRLs and certificates at the same time and perform the checking of the certificate revocation status and the certificate path validation, in parallel.

In addition, we showed typical application, OCVP, which neither requires the CRL mechanism nor a new trusted server. With respect to the computational load, the loads in OCVP is $2n$ which is the same in SCVP. However, SCVP uses using the CRL mechanism and a new trusted server. Therefore, the SCVP gives us a probabilistic answer. However, OCVP uses all CAs who are located on the certification full path and gives us an exact answer.

However, the communication loads among CAs is too big. Therefore, the reduction of the communication loads among CAs needs for further study.

References

[1] W. Diffie, and M. Hellman, "New Directions in Cryptography", IEEE Transaction on Information Theory, IT-22, 1976, pp. 644-654.

[2] Loren M. KohnFelder, "Towards a Practical Public-key Cryptosystem", B.S. Thesis, supervised by L. Adleman, MIT, May 1978.

[3] Carl Ellison and Bruce Schneier. "Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure", Computer Security Journal, Vol. 16, No. 1, 2000, pp. 1-7.

[4] R. Housley, W. Ford, W. Polk, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", January 1999, IETF RFC 2459.

[5] M. St Johns, "The PKIX UserGroupName GeneralName Type", July 2001, IETF draft-ietf-pkix-usergroup-00.

[6] ASPeCT, "Report on final trial and demonstration", AC095/PFN/W12/DS/P/19/1, April 1998.

[7] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", June 1999, IETF RFC 2560.

[8] Ambarish Malpani, Paul Hoffman, Russ Housley, and Trevor Freeman, "Simple Certificate Validatin Protocol (SCVP)", July, 2001, IETF draft-ietf-pkix-scvp-06.txt.

[9] Warwich Ford and Michael S. Baum, "Secure Electronic Commerce", Prentice Hall PTR, 1997.

[10] David A. Cooper, "A model of certificate revocation", In Proceedings of the Fifteenth Annual Computer Security Applicatons Conference, December 1999.

[11] A. Arnes, "Public Key Certificate Revocation Schemes", Ph.D Thesis, Queen's University, February 2000.

[12] Shimshon Berkovits, Santosh Chokhani, Judith A. Furlong, Jisoo A. Geiter, and Jonathan C. Guild, "Public Key Infrastructure Study: Final Report", MITRE Corporation, April 1994.

[13] A. Menezes, P. Van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", CRC Press.