A Best Practice for Root CA Key Update in PKI

InKyoung Jeun[1], Jongwook Park[1], TaeKyu Choi[1], SangWan Park[1], BaeHyo Park[1], ByungKwon Lee[1], and YongSup Shin[2]

[1] Korea Information Security Agency,
78, Garak-Dong Songpa-Gu, Seoul, 138-803 Korea
{ikjeun,khopri,tkchoi,shpark,parkbh,byungkle}@kisa.or.kr
[2] Ministry of Information and Communication,
100, Sejong-ro, Chongro-Gu, Seoul, 110-777 Korea
ysshin@mic.go.kr

Abstract. User authentication, data integrity and non-repudiation services using public-key infrastructure(PKI) are based on the assumption of the trust toward the root CA key in its domain. This root CA key which is commonly encoded as a self-signed certificate has a validity period and it must be updated before the expiration date of it. To do so, an appropriate root CA key update procedure must be proceeded. This paper explains the requirements and a concrete procedure for a root CA key update and the related security issues. Also we will provide an effective root CA key update mechanism considering a security and efficiency, which can be a best practice for handling the root CA certificate expiration.

1 Introduction

The basic role of PKI is the management of "trust" which is expressed in certificates and preserved by verifying the validity of the certificates along certificate paths. The beginning point of the stream of the "trust" in hierarchical PKI is the trust for a root CA certificate. Root CA Certificates are the parents of the subordinated CA's certificates which are parents of the end-user's certificates, in general. Hence if a root CA's certificate expires, in effect, so do all its children and henceforth all end user's certificates.

Thus, one of the most important things in PKI is to establish and manage the trust point for the root CA Certificates. X.509 based self-signed certificate is commonly used to establish the trust point of root CA[1,2]. Due to the fact that the trust in such self-signed certificates can only be implicit, the mechanisms to establish and manage the trust point of root CA Certificates must be well prepared and informed.

Very recently, there was widespread disruption of normal internet services, such as online banks in Singapore reportedly went offline, or at least refused to do any banking, etc., and it has been told that all such problems came from root CA's certificate business. Especially, inappropriate handling of the root CA

M. Jakobsson, M. Yung, J. Zhou (Eds.): ACNS 2004, LNCS 3089, pp. 278–291, 2004.

Certificate Expiration[3]. As we've seen from such cases, a best practices for handling the root CA Certificate expiration will be ever more emphasized.

Self-signed certificate contains the root CA public key and the corresponding private key must be stored in the root CA in a secure manner. The validation period of the root CA key is specified in the self-signed certificate. It cannot be no longer used after the expiration date. So the root CA key must be updated before the expiration date of it. Based on the updated root CA key, the root CA certificate will be updated and distributed to each subordinated CAs and subscribers, then it can be effective through the transition procedure of the trust point from the old root CA key to the new root CA key.

Up to our knowledge, the mechanisms which deal with the root CA key update was not published. There are some public documents only on the transition procedure of the trust point. RFC2510 of IETF and CTL(Certificate Trust List)method of Microsoft are some of the publications on the transition procedure of the trust point.

In this paper, we will investigate an effective root CA key update mechanism which can be a best practice for handling the root CA certificate expiration. Furthermore, We shall discuss the requirements, the desirable model and the detailed mechanisms for secure and efficient root CA key update on the hierarchical PKI.

The rest of this paper is organized as follows. We shall describe the preliminaries for the key update and the life cycle of the certificates in Section 2. In Section 3, we shall describe the process of root CA key update mechanism and compare the RFC2510 of IETF and the CTL method as transition procedure of the trust point. In Section 4, we shall specify the requirements for effective root CA key update and give the desirable solutions in terms of each requirement. And this paper will make a conclusion in section 5.

2 Preliminary

We have stressed a necessity of effective mechanism for the root CA key update in section 1. Before presenting the details about the mechanism, it is also useful for us to define a terminologies used in this paper as follows.

Validity period of root CA key: A root CA key which is used for the trust point in PKI has a limited validity period. Limiting the validity period reduce the possibility that an attacker can identify a root CA private key. The longer root CA uses a private key for certificate signing, the more information there will be that an attacker can use for cryptanalystic attack. Generally, the validity period of root CA key is the same as the validity period of self-signed certificate, because the root CA public key is encoded as self-signed certificate. Of course, the validity period of root CA private key is set differently with the validity period of the public key using the Private Key Usage Period extension in certificate[4,5]. In this paper, however, we assume that the validity period of root CA key pair is the same as the validity period of self-signed certificate.

Root CA key update: As mentioned above, the root CA key has a validity period. Therefore, the new root CA key need to be generated in order to replace the ones that are discontinued. It refers to this situation as root CA key update. There is another reason for root CA key update. that's a security problem of root CA key pair. The better computer calculation ability and the hacking skills are continuously increased and advanced, the more threats to root CA key pair are increased. Therefore, the root CA key must be updated when its safety is impeded or judged to do so. If the root CA key is updated due to key expiration, we can use a same root CA key for updating. But the root CA key is updated due to key safety, we must change the root CA key for updating.

Certificate Update: When the root CA key is updated, a new self-signed certificate is needed to distribute the updated root CA key. To do this, the self-signed certificate of root CA must be updated. The subject name included in the updated self-signed certificate can be same or different with the subject name of old self-signed certificate. But the validity period of the updated certificate must be extended.

Certificate life cycle: Certificate has a life cycle such as issuing, updating, renewal and revocation which is defined in certificate policy(CP) or certificate policy statements (CPS) of each PKI domain. These terminologies about the certificate life cycle could be differently used in each countries according to their PKI policy. As mentioned the above, a certificate update is the process that a new certificate is issued in order to replace a certificate which will be expired. Certificate Renewal which is one of the certificate life cycle involves the generation of a new key pair and issuing a certificate of a new public key. It may arise in the case of a key compromise. The existing certificate is expired before the renewal. Finally, certificate modification becomes a modification of the contents of a certificate during the validity period of it. There may be a need to modify the contents of a certificate when the legal name is changed, and the certificate profile is changed as well as the other information. In case of the certificate modification, the validity period of certificate and the key pair cannot be changed [6,7].

3 Root CA Key Update Mechanism

Root CA key update brings the issues of how the self-signed certificate can be updated and how the new root CA public key can be distributed in an authenticated manner. For the self-signed certificate update, we must decide when the root CA key must be updated, whether the root CA key included in a self-signed certificate must be changed or not, and whether the subject name of self-signed certificate must be changed or not. The subject name and the public key in self-signed certificate can be same, and also changed. What kind of problems have occur if we change the subject name or the public key in the case

of root CA key update? These issues are very important but haven't defined yet in any specification regarding the root CA key update.

Once a root CA key is updated, it is necessary for all entities in PKI to receive the updated root CA public key in a secure manner in order to prevent an attacker substituting the wrong root CA public key instead the real public key. For a distribution of the update root CA key, we must consider a delivery method of the new self-signed certificate and transition procedure of the trust point from the old root CA key to the new root CA key in a secure and reliable way. We can use different techniques to delivery the self-signed certificate such as HTTP web service, LDAP repository and so on. For the transition procedure of the trust point, there are CMP(Certificate Management Protocol) method which is presented in RFC2510 of IETF and CTL(Certificate Trust List) method which is developed by Microsoft Corp[8,14].

3.1 Certificate Management Protocol(CMP)

As explained above, CMP is one of the standards related to the root CA key update and it states clearly the transition procedure of the trust point using the certificates issued by root CA. CMP is a protocol for issuing, revocation, renewal, and updating of certificate and it is used for certificate management in many PKI products. It is a fundamental concept that the old root CA key ensures a reliability of the new root CA key and new one ensures reliability of the old one. For this, a root CA issues a pair of link certificates simultaneously. The first link certificate contains the new root CA public key signed with the old root CA private key. The second link certificate contains the new root CA public key, and it is signed with the old root CA private key. In this way, subscribers who have a certificate signed with the old root CA private key, and subscribers who have a certificate signed with the new root CA private key, can validate each other's certificates[8].

To update the key of the root CA, certificates are issued as follows.

OldWithOld Certificate: This is containing the old root CA public key signed with the old root CA private key.

OldWithNew Certificate: This is containing the old root CA public key signed with the new root CA private key. This certificate allows the subscriber's certificate signed by the new root CA private key to construct a valid certification path to the certificate previously signed with the old root CA private key.

NewWithOld Certificate: This is containing the new root CA public key signed with the new root CA private key. This certificate allows the subscriber's certificates signed by the old root CA private key to construct a valid certification path to the certificates signed with the new root CA private key.

NewWithNew Certificate: This is containing the new root CA public key signed with the new root CA private key.

The point to pay attention here is that these four kinds of certificates have to be published via repository or other means, like a CAKeyUpdAnn(CA Key Update Announcement) message. CAKeyUpdAnn message includes three types of certificates, OldWithNew certificate, NewWithOld certificate and NewWithNew certificate. When the root CA key is updated, CA may transfer this message to all entities to inform that the root CA key is updated. After all, CMP method uses CAKeyUpdAnn message or repository for delivery of the updated self-signed certificate and uses the link certificates like OldWithNew certificate and NewWithOld certificate for the transition of the trust point to the new root CA public key.

3.2 Certificate Trust List(CTL)

We will show the CTL as one of the transition method of the trust point. CTL is usually used as a mechanism in order to trust a CA certificate of other PKI domain for cross certification. It is a PKCS#7 signed data content which is signed with a trust CA key and is composed of a list which includes fingerprints of the trust certificates as below[14,15].

```
CertificateTrustList ::= SEQUENCE
  Version             Version  DEFAULT v1
  subjectUsage        SubjectUsage,
  listIdentifier      ListIdentifier OPTIONAL,
  sequenceNumber      INTEGER,
  thisUpdate          Time,
  nextUpdate          Time,
  subjectAlgorithm    AlgorithmIdentifier,
  trustedSubjects     TrustedSubjects,
  extensions          Extensions OPTIONAL
```

In the structure of CertificateTrustList, trustedSubjects field is a list of fingerprints of all trust CA certificates. After the self-signed certificate is updated, the fingerprint of it is added in the structure. It is the way that a user who trusts the old root CA key can trust the updated new root CA key by acquiring the CTL signed with the old root CA key. If a fingerprint of the updated self-signed certificate were included in the trustedSubjects list, a user can trust the new root CA public key. For the trust of new root CA key, a user only verify the signature of CTL using the old root CA key and there is no necessity for client to come in a root CA key update procedure. CTL dose not include any public key and just include a fingerprint of certificate. Therefore, there is another need for delivery mechanism of a self-signed certificate.

4 Consideration for Root CA Key Update

In this section, we represent a root CA key update procedure which is composed of self-signed certificate update and distribution of the updated root CA key. To

develop an appropriate root CA key update mechanism, we must consider the following requirements.

4.1 Selection Criteria for Root CA Key Update Mechanism

Self-signed certificate update. When the self-signed certificate is updated, there is no restriction for choosing the subject name of certificate and key pair. The root CA key pair can be updated using the same key or different key by considering its security. The subject name of self-signed certificate can be same or different by the policy. Also, the update point of the self-signed certificate must be calculated by considering a validity period of the subordinate CA certificate for offering a PKI service securely and continuously.

Delivery of self-signed certificate. Self-signed certificate can be delivered to PKI entities with various ways. The root CA key update mechanism must not restrict the delivery method of updated self-signed certificate, but ensure the security of the delivery procedure.

Transition of the trust point. After the root CA key is the updated and the new self-signed certificate is delivered, PKI entities must trust the updated root CA public key for certificate verification. This trust method should be performed in a reliable way. Also, this transition method of the trust point is practicable without regard to change of the key or other information of self-signed certificate.

4.2 Requirements in Terms of Self-Signed Certificate Update

First, we must consider whether the root CA key should be changed or not for self-signed certificate update as we mentioned above. The root CA key pair can be updated by the same key or different key considering the security of it. In the perspective of every root CA key update, we must examine an environmental factors such as a computing power, hacking technology, and then we must decide to change the algorithm or length of root CA key. There are two factors regarding to the root CA key security, one is a signature algorithm and key length, and the other is the validity period of key.

According to the Data protection security survey of RSA Laboratories in 2003, RSA algorithm is recommended to use 1024 bits until 2010 year, minimum 2048 bits until 2030 year, and after then minimum 3172 bits[10]. Besides that, an announcement about the digital signature in German of RegTP, recommends that the RSA algorithm is safe to the end of year 2007 using minimum 1024 bits(recommend 2048 bits) and to end of year 2008 using minimum 1280 bits(recommend 2048 bits)[11]. Lenstra and Vercheul also recommend that we can use a RSA 1024 bits approximately by 2005 year and use a RSA 2048 bits

by 2025 year in Selection Cryptographic Key Size[12]. However, as for these factors, it has been consistently changed according to the environmental factors from time to time, so the safety of a root CA key should be considered by these factors. At the end, the decision of root CA key length, signature algorithm, and validity period becomes a political issue. A PKI policy manager must determine above factors within a possible range in order to make those be accepted.

Secondly, we must decide whether the subject name in updated self-signed certificate is changed or not. The subject name is significantly used for certificate path construction. According to the decision of root CA policy, the subject name can be changed or not. Considering the change of a key and the subject name in the certificate, a self-signed certificate update model can be divided into fore different types as shown Table 1. As we can see from this table, self-signed certificate update is free from the change of a key and subject name.

Table 1. Self-signed certificate update model

Type	Root CA Key Change	Subject name Change
All same	X	X
Only different the subject name	X	O
Only different the root CA key	O	X
All different	O	O

Finally, we must consider the update point of the key. It can be calculated by considering the validity period of subordinate CA certificate[1,5,10,12]. We assume that the validity period of root CA key is KP_{RCA} which is the ranged from the start date of root CA key validity(RK_{START}) to the expire date of root key validity(RK_{END}) and presented [RK_{START} - RK_{END}]. We can also assume that the validity term of subordinate CA certificate, which is issued at current time, is KT_{SCA}, that is a value of the expiration date of subordinate CA certificate validity(SK_{END}) minus the start date of subordinate CA certificate(SK_{START}). The expiration date of root CA key must be longer than expiration date of subordinate CA certificate. That is $SK_{END} < RK_{END}$. If $(RK_{END} - RK_{START}) < KT_{SCA}$, then $SK_{END} > RK_{END}$, and then verification of the subordinate CA certificate error has occurred since RK_{END}. Therefore, the root CA key must be updated before $(RK_{END} - KT_{SCA})$ point. Based on the above assumptions, we can propose some terminologies about the validity period.

Definition 1. *Update Point(UP)*
Update point is the time that the root CA key must be updated. As for this, the representation is possible as follows.

$UP = RK_{END} - KT_{SCA}$

The root key must be updated before UP for $SK_{END} < RK_{END}$.

Definition 2. *Valid Period(VP)*
Valid period is when the root CA key is effective, and the verification of a root CA key must be certainly possible during this period. VP is the same as the validity of self-signed certificate signed with the root CA key. That is,

$VP = KP_{RCA} = [RK_{START}$ - $RK_{END}]$

The root CA key must be always effective unless the root CA key isn't revoked or hold during VP.

Definition 3. *Active Period(AP)*
Active period is the period when a subordinate CA certificate or subordinate CA certificate revocation list issuing is possible with the root CA key. This is presented as follows.

$AP = [RK_{START}$ - $UP]$

Namely, AP is a period from RK_{START} to UP. The security of root CA key must be assured during this period.

Assume that the first root CA key is K1, and then sequentially updated key is K2, and so on. The UP, VP, AP of K1 are shown in Fig 1.

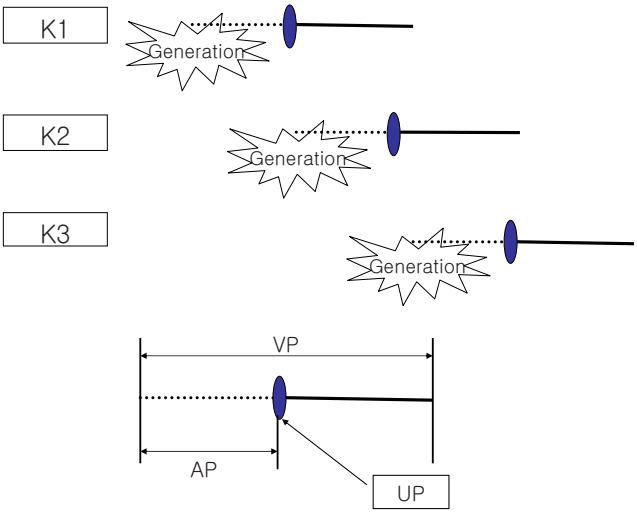

Fig. 1. Valid period of Root Key

4.3 Delivery Methods of Updated Self-Signed Certificate

After the self-signed certificate of root CA is updated, the certificate must be delivered to all subscribers for certificate validation. There are many methods

for delivery of updated self-signed certificate and PKI domain can choose the method as they want.

First method for a delivery of self-signed certificate is that root CA may simply post a self-signed certificate on a web site or in repository with or without a secure measures. Subscribers can acquire the self-signed certificate through accessing the web or repository. The other method for delivery a self-signed certificate is to use of subscribers software. Root CA can make arrangements with manufactures of subscribers software to have the root CA public key implemented on the software. When this mechanism is used, the root CA public key have already delivered in a reliable manner to subscribers without the need for special method as repository access. Finally, a root CA or subordinate CA or RA which is trusted by CA may directly provide the updated root CA public key to the relying party during a face-to-face meeting at the time of initial registration[13]. In this case, the self-signed certificate could be contained in a storage device as floppy disk or USB token.

4.4 Transition Procedures of the Trust Point

After all entities like CAs, users, and PKI application servers are received the updated self-signed certificate, there is a necessary for transition procedure of the trust point from the old root CA public key to the new root CA public key contained in the acquired self-signed certificate. As for the transition procedure of the trust point toward the new root CA key, we mentioned CMP and CTL method in the above. In this section, we will compare these two methods in details.

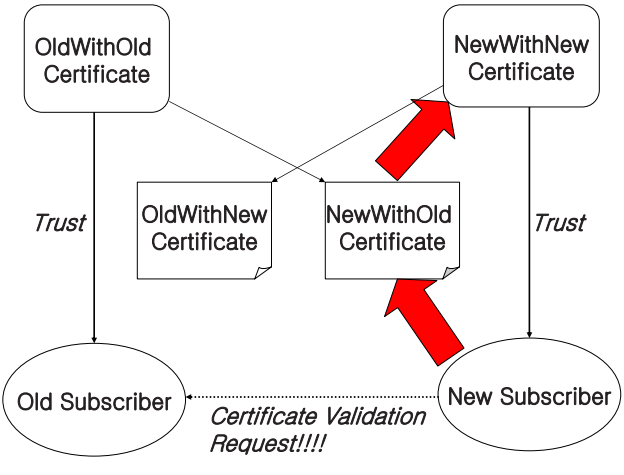

Fig. 2. Transition of the trust point using CMP

Fig.2 is shown that a user who trust the old root CA key can trust a new root CA key using CMP. An old subscriber who have trusted the old root CA key must trust the new root CA key for validation of the new subscriber's certificate which is issued by the new root CA key. To do this, the old subscriber uses the NewWithOld link certificate which is signed with the old root CA key trusted by itself. That is the old root CA key ensures a reliability of the new root CA key using singing. The new subscriber who trust the new root CA key uses the OldWithNew link certificate to trust the old root CA key, vice versa.

If we use the CMP method for trust of updated root CA key, then we will meet some restrictions. The first restriction is the CMP is practicable when the root CA key is updated with the changed key. And in the link certificates which are issued by the root CA, the issuer and subject names are identical. That is, the root CA key must use a different key and a same subject name for updating[2,8]. If the safety of root CA key is ensured enough after updating, and the root CA policy dose not need to change the root CA key, then this method can cause the additional cost along with the root CA key update. It is obvious if we update the root CA key by using a different key, its safety would be relatively improving. However, this is must be set up within a policy range. The root CA key update method using a same key can be needed when the its safety is ensured during the validity period after updating. We already dealt with this issue in section 4.2.

There is another restriction of a CMP mechanism. It is a complexity of procedures for verifying the new root CA key. The users who trust the old root CA key must acquire the NewWithOld link certificate to verify the new root CA key and vice versa. This means the increment of a certificate chain by the link certificates in the existing certificate path chain. This verification method is unsuitable for an application which request very short time for verification of certificate and it causes a implementation complexity of the end entity software. Also, CMP is defined as many certificate management protocols like certificate issuing, update, and revocation. For using the CMP method in the root CA update mechanism, the above full protocol must be implemented in software of CAs and subscribers. This is another restriction of the CMP method.

Another trust method for the updated root CA key is a CTL.The trust procedure of the new root CA key using CTL is shown in Fig.3 and the detailed steps are as follows.

1. CA1 acquires the certificate(PK_{CA2}) containing the updated public key of CA2.
2. CA1 creates a CTL signed with the CA1's old key, CTL include a fingerprint of certificate of CA2(PK_{CA2}).
3. CA1 announces the CTL to a repository using LDAP or HTTP.
4. A user who want to verify the new user certificate signed with the new root CA key acquires from a repository.
5. A user can trust the new root CA key of CA2 from the verification of CTL CA1 public key and confirming that the fingerprint of CA2 self-signed certificate is included in that CTL.

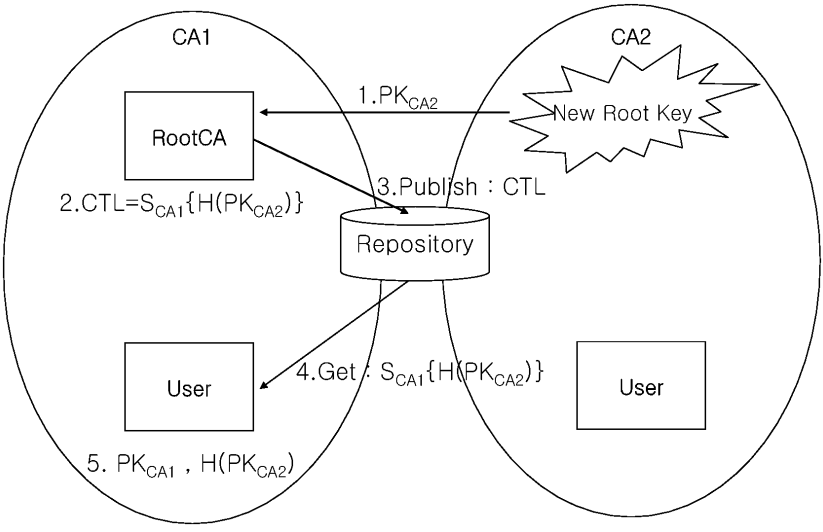

Fig. 3. Transition procedure of the trust point using CTL

The CTL method is similar with the CMP in relation that the old public key ensure the reliability of the new public key. But this mechanism is different with a CMP as the following reasons.

- In the CMP, the NewWithNew certificate or link certificates contain a new root CA public key. But in CTL, only fingerprint of certificate is included. That is, CTL does not contain any public key. So the new public key must be acquired using any other mechanism mentioned in 4.3.
- The number of contained public key in CMP certificate is the only one. Whereas a CTL may contain many fingerprints of the public key certificates. Only one CTL is needed whether the number of root CA public key is one or many.
- In the CMP mechanism, a root CA commonly withdraws the new root CA key by issuing an authority certificate revocation list(ARL). Whereas CTL withdraws the new root CA key by excluding the fingerprint of the new certificate in the trust list.
- The CMP mechanism is standardized by IETF whereas the CTL has not been standardized. But the CTL has been used in many applications like MS explorer.
- The root CA update protocol of CMP is only for the root CA key update, so it needs another method for cross certificate like issuing of cross certificate or certificate trust list[9]. The CTL can be used for the cross certificate too. The hash value of other PKI domain certificate is included in certificate trust list. In this case, the certificate for the cross certificate must be distinguished from the self-signed certificate for the root CA key update.

The most characteristic when we use CTL for trust of a new root CA key is an independency for changing of a root CA key and subject name, so it is available to any root CA key update model mentioned section 4.2. The validity period and the serial number of certificate are changed in all of the self-signed certificate update models, that's why its fingerprint is changed.

4.5 Our Choice for Transition Procedure of the Trust Point

Table 2.is shown the comparison of the CTL transition procedure of the trust point and the CMP transition procedure. Although these two methods are based on the assumption of the trust to the old root CA key, we may find there are some differences. We mentioned our requirements for the root CA key update mechanism in above. In this section, we will select the best procedure for the transition of the trust point to be satisfied with our requirements.

Table 2. Comparison CTL method with CMP method

Issue	CTL method	CMP method
Include the public key	Not included	Included
Basic Assumption	Trust of old root CA key	Trust of old root CA key
Change of root CA key	Independent	Must be changed
Changed of subject name	Independent	Must be identical
Implementation	easy	complex
Scalability to the cross certificate	possible	Impossible
Efficiency	Good	Not good

Our first requirement for the self-signed certificate update is that a root CA key and subject name can be changed or not. A root CA key must be changed in CMP method for root CA key update, but the CTL method satisfies this requirement. That is, the CTL method doesn't require that we must changed the root CA key. Also, a subject name in certificate must not be changed in CMP method, but the CTL method has no restriction for choosing the subject name. That is, the CTL method satisfy our first requirement.

The second requirement is that the updated self-signed certificate could be delivered in many ways. In the CMP method, the self-signed certificate must be posted into repository or transferred by only the CAKeyUpdAnn message. But the CTL method has no restriction for delivering it. The updated certificate can be posted in repository or web, implemented in subscribers software, and directly provided by the root CA as well as the subordinated CA.

The third requirement about the transition of the trust point is satisfied by these two. Besides that, the CTL method is useful for implementation and scalability to the cross certificate. As compared with the CMP, the CTL method holds these merits for efficiency. Because of these reasons, we select CTL method for transition procedure of the trust point in this paper.

5 Conclusion

It is necessary to update the root CA key which is the beginning point of the trust in PKI, due to the validity period expiration of the key and the concern about the safety. If a root CA key is updated by that reason, all of the end entities in relevant PKI domain must receive the updated root CA key in a reliable way and need a transition procedure of the trust point from the old root CA key to the new root CA key. In this paper, we have described the process of root CA key update mechanism which is composed of a self-signed certificate update, distribution of the updated root CA key, and transition procedure of the trust point.

If a root CA key is updated according to the validity period expiration, we must estimate an appropriate update point, considering a validity period of the subordinate CA certificate. Also in the perspective of every root CA key update, the signature algorithm, key length and validity period of the root CA key must be reconsidered in order to ensure its safety. The root CA certificate must be updated after the root CA key is updated and distributed to each subscribers. Then, it can be effective through the transition procedure of the trust point as the CTL method.

For the update of a root CA key, it requires an appropriate change to all entities software of relevant PKI domain, and it causes a consequence for additional cost spending. Through the practical use for this best practice for a root CA key update, each PKI domain is able to establish a root CA key update procedure effectively and safely.

Acknowledgement. We would like to thank Dr. Moti Yung for his valuable suggestion and useful comments.

References

1. Tim Moses, *PKI Trust Models*, available at http://www.itu.dk/courses/DSK/E2003/DOCS/PKI-Trust-models.pdf, 2000
2. Russ Housley, Tim Polk, *Planning for PKI*, Wiley Computer Publishing, pp.103-105, 2001
3. Matt Londy, ZDNet(UK), *Last week's mini-Y2K:What went wrong?*, available at http://zdnet.com.com/2100-1107-2-5140009.html, January 13, 2004
4. R. Housley et al., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC3280, IETF, April, 2002
5. ITU-T Recommendation X.509 (1997) | ISO/IEC 9594-8:1998, *Information technology - Open Systems Interconnection - The Directory : Authentication Framework*, 1998
6. Information Security Committee, American Bar Association, *PKI Assessment Guidelines*, June 2001
7. S. Chokhani el al., *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, RFC2527, IETF
8. C. Adams et al., *Internet X.509 Public Key Infrastructure Certificate Management Protocols*, RFC2510, IETF, 1999

9. Interoperability Working Group, *Asia PKI Interoperability Guideline v1.0*, Asia PKI Forum, March 2003
10. Burt Kaliski, RSA Laboratories *TWIRL and RSA Key Size*, available at http://www.rsasecurity.com/rsalabs/technotes/twirl.html, Revised May 6, 2003
11. RegTP, *Notification in accordance with the Electronic Signature Act and the Electronic Signature Ordinance*, Federal Gazette No 49, pp 4202-4203 of 11 Match 2003
12. Arjen K.Lenstra, Eric R.Verheul, Selecting Cryptographic Key Sizes, 2001
13. Radis Perlman, Sun Microsystems, *An Overview of PKI Trust Models*, IEEE Network, 1999
14. Trevor Freeman, *Certificate Trust List*, Microsoft Corporation
15. A RSA Laboratories, *PKCS7 : Cryptographic Message Syntax Standard*, Revised November 1, 1993