Good practices for long-term key management in a Public Key Infrastructure[*]

Marcelo Carlomagno Carlos Ricardo Felipe Custódio
Jeandré Monteiro Sutil
Universidade Federal de Santa Catarina
Laboratório de Segurança em Computação
Caixa Postal 476, 88040-900, Florianópolis, SC, Brasil
mcc@inf.ufsc.br, custodio@inf.ufsc.br, jeanms@inf.ufsc.br

Abstract

Optimal PKI life cycle management depends directly on the strategy to deal with the update and replacement of CA certificates and CA private keys. To reach optimal strategy, it is necessary to develop methods that the replacement is executed to match the specific needs of each PKI. Only one strategy is defined in RFC 4210, but real PKIs need a variety of different strategies. This paper classifies these strategies and presents the corresponding procedures to replace certificates and private keys.

Keywords : *PKI, certification authority, key pair update, certificate update, CA update*

1 Introduction

Wide scale use of public keys to improve security of communications and electronic transactions in computers is possible due in part to the use of cryptographic mechanisms and key management techniques that create a Public Key Infrastructure (PKI).

The scientific community, private companies and government agencies have made efforts to elaborate standards and recommendations [7, 5, 1] that regulate how a PKI should be implemented. This has facilitated the deployment of new applications using PKI services. However, after two decades of use, these recommendations had been revealed as extremely inflexible and their functionalities do not contain all the features necessary to maintain long-term PKI services. One of these problems, which this paper discusses, is the difficulty of replacing certification authority (CA) certificates and cryptographic key pairs.

Recently, PKI managers realized that there were many different types of certificate and private keys replacement which had not been foreseen in PKI deployment. It is proposed here that there are four types of replacement operations that could be carried out on a CA as shown in Figure 1: i) only the certificate; ii) the key pair without using the old private key; iii) the key pair using the old private key with copy restrictions; and iv) the key pair using the old private key without copy restrictions.

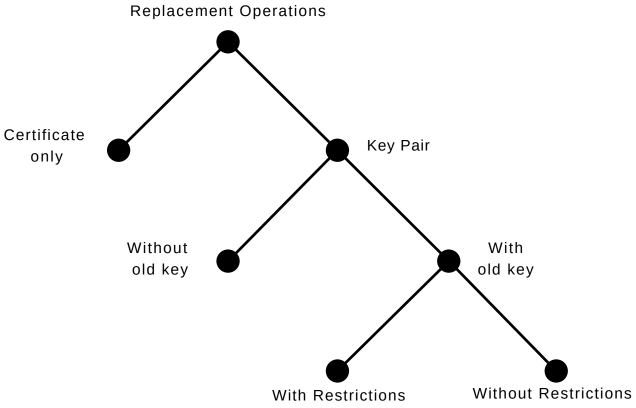

Figure 1. Types of replacements.

There are two solutions proposed in the literature for these replacements. One is defined in RFC 4210 [1] and the other one is the Certificate Trust List (CTL) [3, 8]. These solutions are adequate only for one particular case where the CA key pair needs to be replaced. They can be used only when the old key pair is available. This case is the last type above (iv) and involves replacing the key pair using the old CA key without restrictions.

Plainly, new procedures are necessary to manage the other situations shown in the figure. These situations occur when it is not possible to make a copy of the old private key either due to the hardware, key management protocols or policy restrictions, when access to the CA private key is not

[*] Supported by FINEP and ITI

available or when only CA certificate replacement is necessary. There are no references in the specialized literature to mechanisms that deal with these problems.

This paper proposes methods for the replacement of CA certificates and key pairs for each one of these situations that are not currently being taken into account.

Section 2 presents some definitions necessary to understand our proposal. Section 3 details the key replacement problem which currently has solutions in the literature. Section 4 presents the proposals for substitution of CA certificates and/or their related keys. Section 5 describes the validation procedure and testing of the new methods. Section 6 contains the final considerations of this paper.

2 Definitions

Before presenting methods of certificate update and key replacement, it is necessary define some terms that will be used in this paper as follows.

Keys Validity Period – the validity period of the keys is commonly the same as the certificate's. However, if they are not restricted by security or policy, the key validity period can be extended beyond the certificate validity to which it is related. Thus, after the certificate expiration, the key can be reused in a new certificate.

Update Point – the time when the CA certificate must be updated. This point occurs when the CA cannot sign more digital certificates because its remaining validity period is less than the validity period of the certificate which it wants to sign.

Certificate update – in this paper, the term certificate update is related to the issue of a new certificate to replace an older one. In this case, the certificate can keep its key pair. Many reasons can cause the update of a certificate: revocation, modification of attributes, or expiry. In the case of revocation, an update can be done if the key was not compromised.

Key pair replacement – when the key pair replacement is needed, the certificate and the key pair should be changed together. Therefore, in this paper, the term key pair replacement means creation of a new key pair and the issue of a new certificate.

3 Current replacement methods of keys and certificates

Currently there are only two methods for dealing with the replacement of a certification authority key pair. This section describes in detail both of these methods.

3.1 Certificate Management Protocol (CMP)

To introduce a new digital certificate or new certificate revocation lists (CRL) signing key, a CA must issue rollover certificates for the old and new key pairs [1]. The rollover certificates are necessary to allow the subscribers of certificates belonging to a PKI to construct a valid certification path for certificates in the same PKI, but signed by a new private key (KR_{new}). The basic procedure consists of protecting the new public key (KU_{new}) using the old private key (KR_{old}) and vice versa, as is illustrated in Figure 2. In this figure, arrows represent certificates, circles represent CA entities, and squares represent end entities.

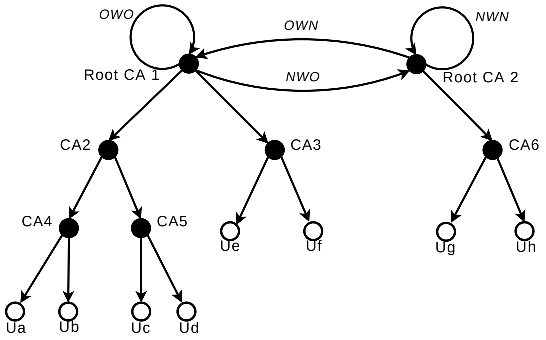

Figure 2. Hierarchic structure after the key pair replacement of the root CA.

For the accomplishment of this process as shown in Figure 2, 3 new certificates must be generated: the certificate NWN for the new root CA (Root CA2) and two rollover certificates. These are Old with New (OWN) and New with Old (NWO). The OWN contains the old public key (KU_{old}) and is signed using KR_{new}. The other, NWO, is signed by the old private key that contains the new public key.

At the end of the process, four certificates will exist: the original certificate – Old with Old (OWO); the self-signed certificate for the new root CA the – New with New (NWN); the OWN signed with the new private key; and the NWO signed with the old private key [1].

The OWN certificate contains the public key of the old certificate, and is signed by the new private key. Thus, it is possible for the subscribers of certificates signed by the new private key, to construct a valid certification path for the certificates signed with the old private key [4]. For this certificate, the validity period is initiated at the moment when the certificate is issued and finishes at the same time and date as the certificate that contains the old public key. The public key of this certificate is the same one as for the old certificate.

The NWO certificate contains the public key of the new certificate, and is signed by the old private key. Thus, the subscribers of certificates signed with old private key can construct a valid certification path for the certificates signed with the new private key [4]. The validity period is initiated at the moment when the certificate is issued and it has the validity period necessary to validate all entities of the old hierarchy. In the worse case, the expiry date is the same as for the old public key.

At the end of the process the old private key is no longer required, and the public key will remain in use (for use in certification chain validation) until all the final entities have securely acquired the new public key [1].

As described earlier in this section, the old private key will be used to sign a certificate that includes the new public key (NWO). Thus, an important issue in this method is that it requires the availability of the old private key.

3.2 Certificate Trust List (CTL)

Initially created as a mechanism for establishing trust in cross certification to prevent the necessity of directory use [3], the CTL also can be used as an integral part of a PKI trust point transition mechanism.

It is a signed data structure using the PKCS#7 [11] format that contains information about policies, validity period, extensions and a list of the fingerprint certificates of the trusted CA certificates. The structure ASN.1 of a CTL is presented below:

```
CertificateTrustList ::= SEQUENCE {
  version                 Version DEFAULT v1,
  subjectUsage            Subject Usage,
  listIdentifier          ListIdentifier OPTIONAL,
  sequenceNumber          INTEGER OPTIONAL,
  thisUpdate              ChoiceOfTime,
  nextUpdate              ChoiceOfTime,
  subjectAlgorithm        AlgorithmIdentifier,
  trustedSubjects         TrustedSubjects,
  extensions              Extensions OPTIONAL }
```

To supply a new certificate and a new CA key pair, it is enough to add the new certificate fingerprint into the CTL of the old CA. Thus, the entities which trust the old CA key will automatically start to trust the key of the new CAs. This happens as soon as the entities acquire the CTL signed by the old CA. The process of presentation of the new certificate and key can be seen in Figure 3 and follows the steps [8]:

1. The new certificate of CA is created (CA_2);
2. The CA_1 gets the certificate of CA_2;
3. The CA_1 creates a CTL, signing it with its private key, including the CA_2 certificate fingerprint;
4. The CA_1 publishes the CTL in a repository (LDAP, HTTP, etc).

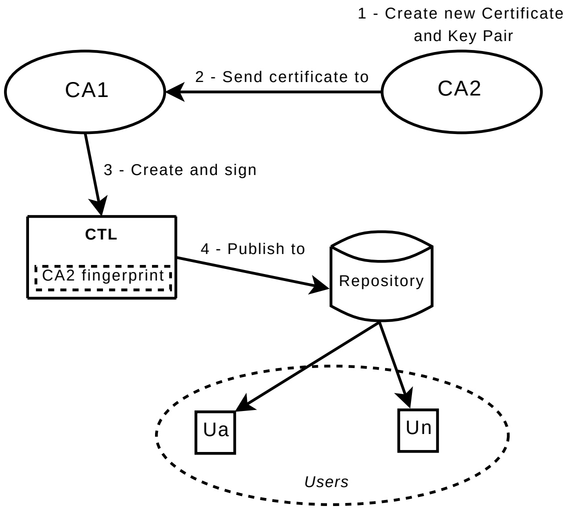

Figure 3. Process of establishing trust in a new certificate using CTL.

Thus, the subscribers of CA_1 start to trust the certificates issued by the CA_2 using the CTL. In contrast to the CMP protocol previously described, the new and old CAs do not have a direct relation to their topology; they need the CTL to establish trust among them.

Whereas the CMP, this method also requires the availability of the old private key. The related works of this paper just use known methods, but as described before, there are situations that only the known methods aren't enough.

4 The new approach

The methods described in Section 3 present the procedures to be carried out when a root CA key pair replacement is needed. Although these methods are sufficiently functional, they do not have the necessary coverage to fulfill all the requirements of a PKI during its life cycle. Its objective is the simple replacement of the root CA private key.

None of the methods presented can be used for simple certificate updating without a key replacement. Another relevant question is choosing whether to update the certificate or the key pair. Practice shows that different PKIs have different needs. Although the methods described above can be employed to solve any of the certificate and/or key replacements shown in the Figure 1, some of them are inadequate.

In this section, a new model will be presented in detail that addresses other replacement situations currently without adequate solutions. As already discussed, two types of operations exist that, if carried out in a CA, can have drastic effects on its topology. These operations are certificate updates and private key replacements. Each one of these modifications has some subdivisions that need to be dealt with in a particular way. In this section, each one of them will be described and respective solutions proposed.

4.1 Certificate replacement

Certificates have a fixed lifetime when they are issued. When its update point arrives, it is necessary to issue a new certificate [2] and, eventually, generate a new key pair.

CA certificate modification becomes necessary for various reasons. Beyond the matter of its validity period, there may be a need to change some of its attributes, or some policies. In all of these cases, the replacement of the private key is not necessary if it is available and there are no restrictions on its security or policy.

There are two types of this process: certificate modification, which is applied when some of the certificate's attributes need changing, and renewal, when it is only necessary to change the expiry date. In the first type, the interested party should issue a request with new attributes. This request is verified by a third party such as a registration authority. After verification, the request is sent to the issuing CA. In the second type, it is not necessary to check the attributes so the issuing CA can issue the certificate independently.

The modification process is started with the creation of a certificate request [10] keeping the same public key of the old certificate. However, some other fields must also be kept unchanged to ensure the path construction and validation [7, 9] for the new certificate. They are:

- **Subject:** this attribute is the basis for path construction. So, in order to maintain a valid path to the new certificate, the Subject must be kept unchanged;
- **Basic Constraints Extension:** if present, this extension is always checked in certification path validations. Any change must be evaluated to avoid policy violations of the PKI;
- **Others:** any other extensions whose changes involve policy restrictions and can compromise certification path construction.

Respecting the restrictions described above, a new certificate is then issued with the same public key. This procedure can be applied for both root and intermediate CAs and also for end entities.

The difference is that for the root CA, the new certificate is self-signed. In addition, for the intermediate CA or the end entity, the same CA that issued the old certificates issues the new certificate. Moreover, the issuing CA must give challenge to check possession of the private key.

Renewal is even simpler. With the old certificate data, a new certificate request is generated with the same information, and the same public key. At the moment when this process is applied to an intermediate CA or end entity, a challenge to guarantee proof of possession of the private key becomes necessary. With this certificate request, a new certificate is issued. The difference between the previous and the new certificate is in the date of issue validity period and in the serial number. All the other values are the same. Thus, there would be no problem in implementing a system which performs this process automatically, consequently eliminating the need for new certificate requests.

When a certificate is modified, it is recommended to revoke the old certificate, unless it already has expired. Such action is necessary to prevent the use of two certificates with the same public key, but with different attribute values. But, when the certificate is renewed, the old certificate can be kept available until it expires.

4.2 Private key replacement

When CA private key replacement is necessary, consequently the substitution of its certificate is also required. Two cases can occur: one when the private key is available; and the other when the opposing situation occurs, and the key is not available. Each one of these cases requires a specific treatment to provide a lower impact on the PKI topology. Next will be described methods indicating how to proceed in both of these cases.

4.2.1 With Unavailable Private Key

The question of availability is very important when dealing with access to a CA private key. When the private key is not available, either due to a failure in the cryptographic device which stores it, or because of policies or of physical restrictions on its use, the entire topological structure is harmed. Until the present, the only solution found for this problem is re-issuing all certificates issued by all the CAs of the PKI. Such a situation can be performed viably in small environments to solve problems such as the need for CRL emission [6]. However, in a CA with a great number of certificates issued, or in CAs that issue other CA certificates, this procedure can be difficult to manage and high cost.

None of the models presented in Section 3 can be applied in this case because the private key is not available, and consequently the old CA will not be able to sign rollover certificates or the CTL. Thus, a new method to issue a new certificate and a new key pair is necessary.

To be fully applicable, this new method must have the following requisites:

- it must allow the construction of a new certification path, compliant with actual methods;

- it must be independent of the old private key;
- it must be simple and easy to apply;
- it must minimize the transition costs over the PKI topology;
- it must use well known and consolidated data structures, preferably only certificates and certificate revocation lists.

Based on the requirements described above, a new procedure was developed.

In an already well established PKI, as that shown in the Figure 4, the first step is create a new CA called the Root CA2. This new CA will have a new key pair, however its certificate must contain the same data as the Root CA1 certificate, differing only in the issue and validity dates, and in the public key itself. The main difference between this certificate generation process and the renewal method described in Section 4.1 is that the key contained in the certificate request is the new public key, instead of the old one.

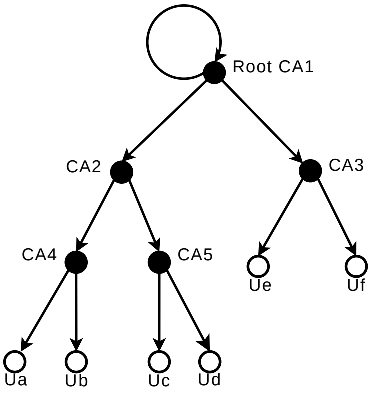

Figure 4. Hierarchical PKI.

With the Root CA2 working, the next step is to generate certificate requests for all the certificates issued by the Root CA1, forming a new hierarchy. The process is very similar to the update defined in Section 4.1. The only difference is that certificate requests will be submitted to the new CA (Root CA2). After this, Root CA2 issues the new certificates as shown in Figure 5.

None of the certificates issued on the lower levels of the hierarchy will suffer implications, because the private key used in the signature of the certificates will be the same. The new root CA certificate must be published in order to all subscribers which want to get this new certificate to do so. The new CAs generated in the procedure can execute the same role as the previous CA (issue and revoke certificates, create CRLs, etc). The new intermediate CA certificates (new_{CA2} and new_{CA3}, in the figure) issued during the process must also be published.

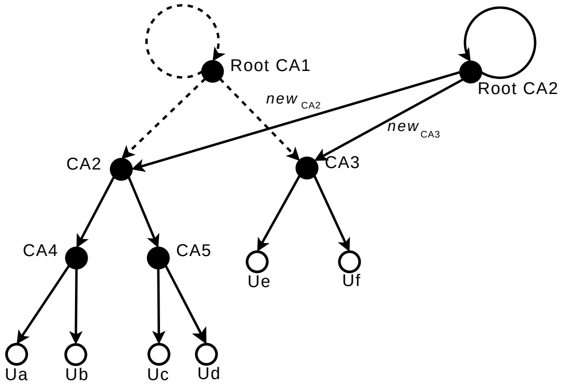

Figure 5. Root CA2 issues certificates.

The certificate of Root CA1 and all the old intermediate CA certificates involved in the process must be kept available until one of two events occur: all the entities which need to obtain them do so or until all the certificates issued by the CAs expire. There is no need to revoke the old certificates. The Root CA1 private key may be discarded, because its use will no longer be necessary. Finally, the keys of intermediate old CAs must be kept, since the new intermediate CAs use the same keys.

Figure 6 presents the new structure after the accomplishment of this process.

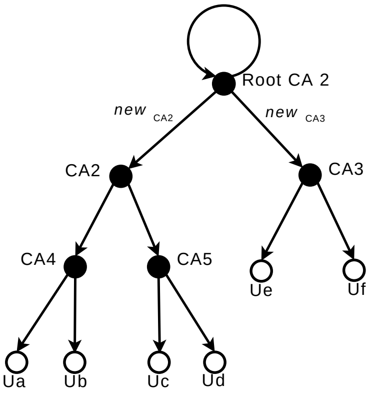

Figure 6. New hierarchical structure.

The validity periods of the new certificates issued in this procedure do not have restrictions beyond those that are applied to re-issue of any certificate, such as security verification of the key algorithm, its size, related policies, etc.

There are no technical restrictions to the use of this procedure in intermediate CAs instead of root CAs. However if this CA issues certificates only for end entities, the cost will be the same as the re-issue of all the already issued certificates.

4.2.2 With available private key

When the private key is available and a private key replacement is wished to be performed, two distinct cases can be identified. One when the private key can be copied and the other when this is not possible.

If the private key is available and there are no restrictions to create a backup, we can consider that the private key is really protected and there is no possibility of loss. Thus, both of the procedures described in Section 3 and the new method presented in this paper (described in Section 4.2.1) can be used without problems. According to the literature, these procedures can only deal with the root CA, but we have found no restrictions to use them to replace intermediate CAs as well.

When the backup of the private key we cannot be done, there is an important point to be taken into account. It is about the confidence in the entity that stores the key. If any problem occurs, the private key will be lost, but there may exist ways to restore it. Thus, two solutions can be performed: apply one of the procedures presented in the Section 3 to create a new CA in a trustworthy environment; or continue to use as is, but if any problem occurs, the procedure described in Section 4.2.1 needs to be applied.

5 Results

In order to verify the proposed methods, some hierarchical structures have been created, with distinct numbers of intermediate CA's, to simulate a real PKI. To increase the accuracy of the results, thousands of end entities certificates have been issued for each CA of the hierarchy.

Some of the issued certificates had the Authority Key Identifier (AKID) and Subject Key Identifier Extensions (SKID), while the others didn't. This was made to test the certificate path construction using the extensions or just with name chaining. Furthermore, two combinations of the AKID extension was tested. One containing only the keyIdentifier *attribute* that will be called basic form and another composed by the keyIdentifier, and also an authoritySerialNumber and authorityCertIssuer pair that will be called complete form.

The certification path validation was verified using the following tools:

- Internet Explorer Browser v6.0: is the most used web browser, with its cryptographic API integrated to Microsoft Windows;
- Mozilla Firefox Browser v2.0: uses your own cryptographic API (NSS) and is growing as an alternative tool for web navigation;
- OpenSSL Tool v0.98: is a tool largely used into open source community, with a vast cryptographic library.

5.1 Expected Behavior

According to the literature [7, 9], in name chaining based certificate path construction, the only condition is that the issued certificate Subject matches the issuer certificate Subject. In case of more than one cadidate, the prefered path is that pointing to the newer certificate. With the AKID and SKID extensions, the behavior is quite similar, but the unique mandatory match is between the AKID keyIdentifier field and SKID. The other two fields in the AKID extension are used just to give preference for a candidate path over another. When the certificates has the complete form of AKID, the certificate path construction algorithm will initially choose the older certificate, but the path that includes the new certificate will be not rejected.

There is no rule to define how to choose candidate paths when all the fields used in construction are the same. So, when an entity is trying to build its certification path, it is suggested to always choose the path where the CA certificate is newer or the path that has the shortest lenght.

5.2 Certificate Modification

The implementation procedure for the method described in Section 4.1 of this paper was carried out in the following way:

1. a PKI was created to reproduce a common hierarchical CA structure;
2. thousands of certificates were issued by all CAs to create an environment similar to real CAs;
3. certification path was verified;
4. a new certificate request was created based on the root CA certificate (RCA_{old}) data, with some changes in its attributes;
5. a new root CA certificate (RCA_{new}) was issued using the certificate request;
6. the new certificate (RCA_{new}) was imported into the list of trusted certificates;
7. certification path was verified again.
8. the old root CA certificate (RCA_{old}) was removed from the list of trusted certificates;
9. certification path was once again verified.

A synthesis of the achieved results is shown in table 1:

As seen in table 1, Mozilla Firefox and OpenSSL didn't present the expected behavior with the complete AKID. This happened because both Firefox and OpenSSL handle

	Construction Method		
Tool	**Name**	**basic AKID**	**complete AKID**
Internet Explorer	✓	✓	✓
Mozilla Firefox	✓	✓	×
OpenSSL	✓	✓	×

Table 1. Results in Root CA certificate modification

the pair authorityCertIssuer and authoritySerialNumber as mandatory fields to construct the certification path, although this behavior is not in agreement with the literature. In Internet Explorer, the behavior was compliant with the expected.

The certification path building to certificates issued by the New CA or Old CA is very similar. In both cases, the certification path is easily built using the key identifiers [9]. When an entity (U_x) which trusts in the old CA (RCA_{old}) try to construct the certification path of an entity (U_y) that is issued by the new CA (RCA_{new}), the path building is made in the following way:

- U_x constructs the certification path of U_y until it find the self-signed certificate on top of U_y topology.
- When the path between U_y and RCA_{new} is verified, the U_x found a valid certification path to U_y and consequently verify that the RCA_{old} and RCA_{new} has the same key identifier.

The same occurs when the U_y tries to construct the certification path to U_x. Thus, the viability of this mechanism has been verified.

As the root CA key pair is the same, the impact of this technique on the remaining PKI can be considered null when certification path building and verification are taken into account.

5.3 Certificate Update

The steps to produce these tests was almost the same than in 5.2. The only difference is that now all the attributes are left unchanged. The serial number and validity are the unique differences between (RCA_{new}) and (RCA_{old}) certificates. The results are exactly the same seen above in table 1.

5.4 Private key replacement

As shown, the private key replacement can be made with and without the old private key. Tests are made for each one of these cases and was implemented and analyzed as follows.

5.4.1 Without an available private key

The implementation procedure of the process described in Section 4.2 was carried out in the following way:

- a PKI was created to reproduce a common hierarchical CA structure;
- thousands of certificates were issued by all CAs to create an environment similar to real CAs;
- certification path was verified;
- a new CA, with a new key pair and certificate, called root CA NWN was created;
- a new certificate request was created for each certificate on the second level of the hierarchy keeping the same data and key pair as used previously;
- the root CA NWN issued all the certificates using these requests;
- the new certificates were imported into the list of trusted certificates;
- certification path was verified again;
- the old root CA certificate (RCA_{old}) was removed from the list of trusted certificates;
- certification path was verified again .

After the procedures shown above, the following results were observed:

	Construction Method		
Tool	**Name**	**basic AKID**	**complete AKID**
Internet Explorer	✓	✓	✓
Mozilla Firefox	✓	✓	×
OpenSSL	✓	✓	×

Table 2. Results in Root CA key pair replacement

The above results presented the same problems seen in certificate update and modification. However, these problems would be avoided if the old sequence of certificate issues had been kept, in order to maintain the same certificate serial numbers found in the previous PKI. In this case, even Mozilla Firefox and OpenSSL would validate the new certificate path with complete AKID extension.

In analysis, if the impact of this procedure is compared to the strategy of re-issuing all certificates, it can be said that the workload is minimum, since the need to re-issue certificates is restricted to only one level of the hierarchy. For instance, in a root CA with 10 intermediate CA certificates, each one of these CAs having 1000 issued certificates, the application of this method would reduce the number to be re-issued from $10,010$ to only 10 certificates. Thus, the number of certificates to be re-issued using this method is equal to the number of certificates which the root CA has issued, ignoring any other certificates emitted by its intermediate CAs.

In the case of the use of this technique with an intermediate CA, the impact can be higher. It depends on whether the CA issues certificates for final entities or other CAs.

5.4.2 With an available private key

When there is permission to copy the private key, as described in Section 4.2.2, three different procedures can be performed, the CMP, the CTL or the new method.

After the accomplishment of tests using the previously described environment, it can be observed that both methods have been shown to be viable, however it was possible to identify some relevant differences between each method.

The CTL implementation was revealed to be simpler, since only 1 new CA certificate and 1 trusted certificate list need to be created, while the use of the CMP mechanism creates 3 new certificates. But when dealing with validation of certification path building, the CMP method is easier, since it uses only known structures of a PKI, while using the CTL the users should know how to manage this structure. This means installing new software into the user's machine.

Thus, when the private key is available, but for any reason there are restrictions on its use, a good choice would be replace the old key using the method described in the Section 4.2.1. It would be good to avoid situations (like hardware or software failure) that could make old key unavaliable.

When the private key is available without restrictions, any method can be used. The CA operator can choose the method that fits to the specific needs of his CA or PKI, and causes a lower impact over it.

6 Conclusions

This paper discusses and proposes new strategies for the replacement of certificates and private keys in long term PKI life cycle. Based on real PKIs, our paper classify the need of replacement in four cases. Based on this classification, this paper demonstrates that the methods proposed in the literature can only address two of these cases, and to the others there are no references.

Therefore, many possible procedures for keys and certificate replacement in different conditions have been described. For each one, an efficient mechanism have been proposed to realize the desired replacement.

For the case of certificate replacement, a method has been presented which can create a new certificate for a CA in a simple way without affecting the PKI topology. Moreover, the implementation of a system to automate this process has been revealed to be viable.

In the case of private key replacement, three categories have been discussed and a new procedure has been proposed based on private key availability. This has low impact at the PKI topology, and is efficient solution in situations not foreseen by recommendations and norms.

The proposals discussed have been verified in real situations using a prototype. The tests results presented proves that the method discussed in this paper is efficient and applicable in real world applications. Furthermore, the results shows that the softwares have different implementations when they deal with certification path construction and validation, and some of them don't address the definitions found at the standards.

References

[1] C. Adams, S. Farrell, T. Kause, and T. Mononen. RFC 4210 – internet x.509 public key infrastructure certificate management protocol (cmp). March 1999.

[2] C. Adams and S. Lloyd. Understanding PKI: Concepts, Standards, and Deployment Considerations. Addison Wesley, 2 edition, November 2002.

[3] T. Freeman. Certificate trust lists: What are they? why are they useful? presentation at the NIST PKI Working Group meeting, November 1998.

[4] R. Housley and T. Polk. Planning for PKI. Wiley, 1 edition, March 2001.

[5] R. Housley, W. Polk, W. Ford, and D. Solo. RFC 3280 – certificate and certificate revocation list (crl) profile. April 2002.

[6] Y.-K. Hsu and S. P. Seymour. An intranet security framework based on short-lived certificates. IEEE Internet Computing, 1998.

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

[8] I. Jeun, J. Park, T. Choi, S. Park, B. Park, B. Lee, and Y. Shin. A best practice for root ca key update in pki. In ACNS, pages 278–291, 2004.

[9] S. Lloyd. Understanding certification path construction. PKI Forum Technical Group, Setembro 2002.

[10] M. Myers, C. Adams, D. Solo, and D. Kemp. RFC 4211 – certificate request message format. September 2005.

[11] RSA. Pkcs#7: Cryptographic message syntax standard. November 1993.