Limited Additional Mechanisms for PKIX and SMIME
Internet-Draft
Intended status: Standards Track
Expires: 24 October 2025

G. Wang, Ed.
Y. Yang
Huawei Int. Pte Ltd
J. Zhang
Huawei Tech. Ltd

22 April 2025

Root CA Certificate Rekeying in the Scenario of Post Quantum Migration
draft-wang-lamps-root-ca-cert-rekeying-02

Abstract

In the public key infrastructures (PKIs), root certifcation authority (CA) certificate rekeying is crucial to guarantee business continuity. Two approaches are given in [RFC4210] for entities which are belonging to different generations to verify each other's certificate chain. However, these approaches rely on the assumption that the old entities can be updated. In this draft, we propose a one-way link certificate based solution such that old entities are transparent to root CA certificate rekeying. Namely, during the overlapping lifetime of two root CA certificates, without any update in old entities, old and new entities can verify each other's certificate chain smoothly. Furthermore, the proposed solution works in both traditional PKIs, and post-quantum (PQ) PKIs, where the cerficate can be pure PQ ones or hybrid ones.

Editorial Note (To be removed by RFC Editor)

Discussion of this draft takes place on the rfc-interest mailing list (rfc-interest@rfc-editor.org), which has its home page at https://www.rfc-editor.org/mailman/listinfo/rfc-interest.

Status of This Memo

This Internet-Draft will expire on 24 October 2025.

Table of Contents

1. Notes of Change

1.1. Changes in v02

* Added an extension that allows root CA certificate can be swithced more smoothly in Section 7, by assuming that there is a maximum time difference (MTD) T bewtween any two entities when one sending a certificate chain and the other receiving.

1.2. Changes in v01

* Added the testing results using OpenSSL library in Section 6.

2. Introduction

In the public key infrastructures (PKIs), root certification authority (CA) certificate management is crucial to guarantee business continuity, as the CA certificate is the trust anchor for certificate chains, which establish the trust paths in PKIs from end users to the trusted authority, i.e., the root CA.

However, just like the certificates for end users, the root CA certificate has a limited time of liftime as well, which normally varies from a few years to several decades to satsify various applications. Further more, while new end user certificates can be issued when old ones are nealy expiring, but a new root CA certificate should be issued normally when the old root CA certificate has been just used for half of its lifetime. Here are the main reasonos behind this pratice.

A root CA ceritifate is generally used to issue one or more suborniate CA (sub-CA) certificates (for different departments in a given organization, for exmpale), and then each of scuh sub-CA certificate is used to issue lower level sub-CA certificates or end user certificates. The lifetime of ened user ceritificates are expected to be around a given period, say 5 years (for some specific products, for example). Accoridg to [RFC4210], to issue end user certificates with validity of 5 years, the remaining lifetime of the sub-CA certificate must be 5 years or more, such that the lifetime of each end user certificate MUST be covered by that of the sub-CA certificate. However, to avoid frequenly apply and mannage multiple or even many sub_CA certificates, the liftetime of each sub-CA certicate could be set as 10 years, which means for each 5 years sub-CA sub_CA certificates should be updated so that end ceritificates for new users or products can still be continously issued to guarratte business continuity. Similarly, for continously issuing sub_CA certificates with validity of 10 years without frequently changing the root CA certificate, it can be expected that each root CA certificate may has lifetime of 20 years, which also implies that each root CA certificate should be updated for each 10 years, not nearly the expiry date of the root CA certificate. Each such new root CA certificate can be called a new generation of root CA certificate, though all of them still belong to the same legal entity, a particular CA.

However, this also implies that different sub-CA certificates and end user certificates issued under different generations of root CA certificate co-exit during the overlapping period of those root CA certificastes. So, it may be not easy to verify each other's certificate chain between entities that hold different generations of certificates issued by the same CA in the aspect of legal meaning but under different generations of root CA certificates.

Say, for the above example, the sub-CA certifiates and end user certifiates issued under the first genarteation of root CA certificate may still be valid in the year of 12 when a given CA has been established, but the newly issued sub-CA certifiates and end user certifiates are actually under the second genarteation of root CA certificate. For simpliciy, we call the owner of a sub-CA certifiate or such an end user as an entity or a device. Moreover, a device or entity hold certificate chain issued by an old generation of root CA certificate are called as old device or old entity, while a device or entity hold certificate chain issued by a new generation of root CA certificate are called as new device or new entity. These terms are in a relative way.

Actually, to address the above issue, there are two solutions given in [RFC4210]:

* Old and new entities upload both old and new root CA certificates,

* Or upload two-way link certificates which introduce old CA to new CA and vice versa.

However, these two solutions do not work if old entities cannot upload the new root certificates or the necessary link certificate, or old devices are even out of maintenance. In fact, such worse cases are possible, due to either limited prediction fo product design such that old devices may do not support adding multiple root CA or link certificates (automatically), or still running old devices are not maintained well and automatically update is not supported.

Back a little bit, basically, there are actullay two approaches to update root CA certifiate, i.e., renewing and rekeying. The former means to extend the validity of an existing root CA certificate but the root CA key pair and the associated cryptographical algorithm is still the same. In this case, the validity of the same (old) key pair of the root CA will be extended for a longer period. So, attackers shall have longer time to cryptanalyze the same key pairs. Also, as time goes, the secuirty strenth of the associated cryptographical algorithm for the root CA certificate may become weaker and weaker. So, soon or later, it still needs to issue a brand new CA certificate.

For the latter, namely rekeying approach, a brand new root CA certificate will be issued to replace the old key pair. In this case,no just a new pair of keys, even different key length or new algorithm can be used for generating the new root CA certificate by considering the progress of cryptoanlysis and potentil security threads in the near future, like quantum computing. However, in this situation, a big challenge is to manage two or even multiple root CA certificates during the overlapping periods, which could be 20 years or more, as mentioned in the above. In particular, some old devices may be not able to install the new root CA or link certificates, such that the two solutions given in [RFC4210] do not work. Therefore, old devices may not be able to verify the new certificate chain of a new device, though a new device can be intalled all necessary certifiates and verify the old certificate chains of old devices.

Motivated by the above observation, this draft proposes a one-way link certificate based solution such that root CA certificate rekeying is transparent to old entities:

* During the overlapping period of two or multiple root CA certificates, without any update in old entities, old and new entities can verify each other’s certificate chain smoothly.

* The proposed solution works in the scenario of traditional PKIs, pure post-quantum PKIs, and also hybrid PKIs [I-D.D24], as the rationale of the solution does not rely on the type of undelying cryptographic algorithms.

* Esentially, the solution can be viewed as an extension of the approach for root CA certificate issuing and managnement specififed in [RFC4210] and [RFC5280].

3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. Design Goals of a New Solution for Root CA Rekeying

Basically speaking, to design the root CA rekeying solution based on one-way link certificate, the following principles are followed to maximize the potential employment of the solution in practice:

* Requirements on old devices are minimized such that the solution is applicable to as many as possible scenarios.

* New devices are supposed to do more as they are normally more powerful and designed with better prediction.

* When possbile, the usage of the new root CA keys will be maximized as normally they associated with cryptographically sronger algorithms .

* The existing standards [RFC4210] and [RFC5280] will be followed as much as possible, so that the deployment of the new solution for root CA rekeying can be implemented as easy as possile.

In Section 6.1 of [RFC5280] : the following is sepecified:

"A prospective certification path (a sequence of n certificates) satisfies the following conditions:

* (a) for all x in {1, ..., n-1}, the subject of certificate x is the issuer of certificate x+1;

* (b) certificate 1 is issued by the trust anchor;

* (c) certificate n is the certificate to be validated (i.e., the target certificate); and

* (d) for all x in {1, ..., n}, the certificate was valid at the time in question. "

While Link certificates was introduced in Section 4.4.1 for specifying CA operator actions to rekeying root CA certificate [RFC4210] as follows.

" To change the key of the CA, the CA operator does the following:

* 1. Generate a new key pair;

* 2. Create a certificate containing the old CA public key signed with the new private key (the "old with new" certificate);

* 3. Create a certificate containing the new CA public key signed with the old private key (the "new with old" certificate);

* 4. Create a certificate containing the new CA public key signed with the new private key (the "new with new" certificate);

* 5. Publish these new certificates via the repository and/or other means (perhaps using a CAKeyUpdAnn message);

* 6. Export the new CA public key so that end entities may acquire it using the "out-of-band" mechanism (if required).

The old CA private key is then no longer required. "

However, as we mentioned in Section 1, the above solution specified in [RFC4210] assumes that (new and old) end entities may acquire the new CA public key (using the "out-of-band" mechanism, if needed), as the above item 6 despicts.

5. The Proposed Solution Based on One-way Link Certificate

Here are the basic idea of the proposed root CA key rekeying based on One-way Link Certificate:

* Use the one-way link certificate, called newWithOld, which is the link certificate of the new public signed by the old private key of the root CA.

* The newWithOld certifies the new root CA key by the old one.

* So, during the overlapping period, old devices can verify a link certificate chain from a new device by using the old root CA certificate as the trust anchor.

* Other cases are simple …

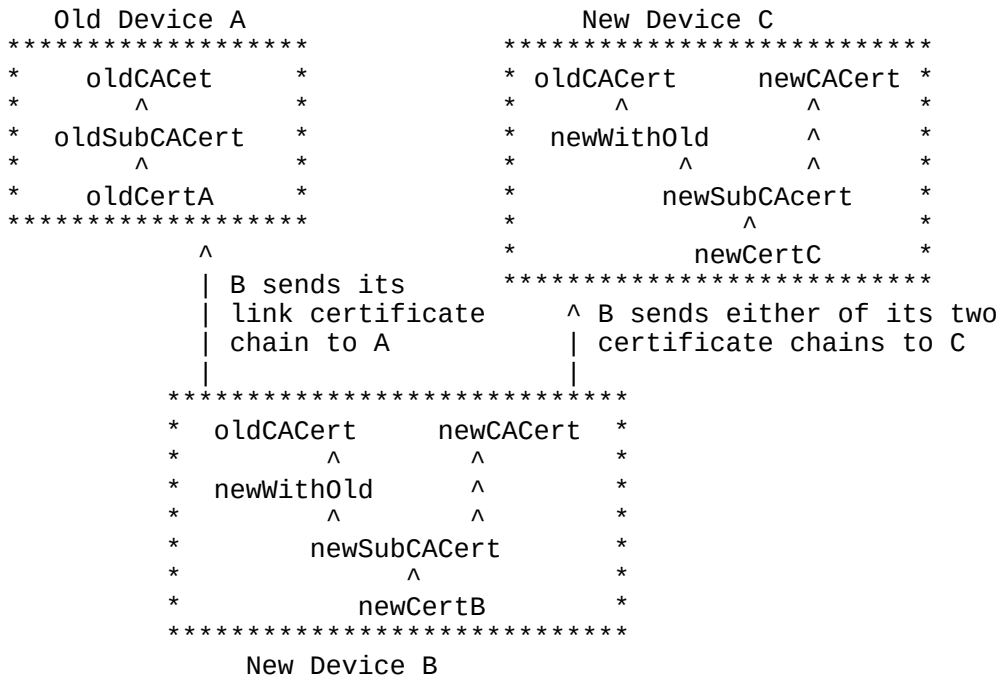

Figure 1. Illustraion to the Solution Based on One-way Link Certificate.

Figure 1 shows how a new device B can send out its link certificate chain, (oldCACert, newWithOld, nweSubCACert, newCertB), to an old device A such that A can verify B’s certificate chain by using the old root CA cerficate, which is A’s trust anchor. To repsond B, A just sends its old certificate chain, (oldCACert, oldSubCACert, oldCertA), to new device B, which can verify A’s certificate chain by using the old root CA cerificate, which is one of B’s two trust anchors, namely, the old or new root CA certificate.

To communicate with another new device C, new device B can send out either its link certificate chain, (oldCACert, newWithOld, nweSubCACert, newCertB), or the new certificate chain, (newCAcert, newSubCACert, newCertB), to new device C, which can verify either of them by using one of its two trust anchors, namely, the old or new root CA certificate. In this case, as both B and C are new devices, C can do as B does such that B can verify either of C’s two certificate chains similarly.

The case of one old device communicates with another old device is simple, as they just behaver as normaly by sending their own old certificate chain to the other.

More detailed description will be provided later.

6. Testing Results

The proposed solution has been tested using OpenSSL library. Here are the testing enviroments:

* 3 generations of PKIs with different root CA certificates.: G1 (2016-2025), G2 (2018-2027), and G3 (2020-2029).

* G1 and G2 implements RSA, with RSA 4096 for root CA certificate, RSA 3092 for subject CA certifiacte, and RSA 2048 for end entity certificate.

* G3 implements ECDSA, with ECDSA 384 for root CA certificate, and ECDSA 256 for both subject CA certifiacte and end entity certificate.

* Cross verifying tests are experimented among implmentations of running OpenSSL 1.0.1c, OpenSSL 1.0.2u、OpenSSL 1.1.1g and JDK 8u251. More specicically, G1 implements RSA with all these four softeare packages, while G2 and G3 implement RSA and ECDA respectively in the last three softeare packages, but not OpenSSL 1.0.1c. This is becuase OpenSSL 1.0.1c is a relativley old version of openSSL, so new system may not employ it.

Table 1 given below lists the life times of three generations of PKIs with different root CA certificates, namely, G1, G2 and G3. That is, the life time of the first genarteation PKI G1 is valid from 2016 to 2025. More specifically, the root CA certificate of G1 is valid from the first moment of the year of 2016 to the last moment of the year of 2025. Therefore, in 2015, G1 is invalid, namely, not issued yet. Similiarly, G1 is expired from the first moment of year 2026. The root CA certificate of the 2nd generation PKI G2 is valid from 2018 to 2027, while the root CA certificate of the 3rd generation PKI G3 is valid from 2020 to 2029.

	2015	2016	2018	2020	2025	2026	2028	2030
G1	Invalid	Valid	Valid	Valid	Valid	Expired	Expired	Expired
G2	Invalid	Invalid	Valid	Valid	Valid	Valid	Expired	Expired
G3	Invalid	Invalid	Invalid	Valid	Valid	Valid	Valid	Expired

Table 1: Life Times of 3 Generations of Root CA Certificate

The testing results for eight selected years from 2015 to 20230 are given in the following four tables, namely, from Table 2 to Table 5. The ruslts show positve answers for all the cases considered, as explanied below in more detail.

These tables show all representive cases needed to be considered. That is, for all three generateions of PKIs G1, G2 and G3, we have selected 8 years from 2015 to 2030 to include all possbile combinations of some of PKIs G1, G2, and G3 are not issued, valid and/or expired. The complete list of these seven combinations is given as following: (G1 invalid, G2 invalid, G3 invaid), (G1 valid, G2 invalid, G3 invaid), (G1 valid, G2 valid, G3 invaid), (G1 valid, G2 valid, G3 vaid), (G1 expired, G2 valid, G3 vaid), (G1 expired, G2 expired, G3 vaid), and (G1 expired, G2 expired, G3 expired). Note that other combinations in theory do not exist in practice, like (G1 valid, G2 expired, G3 vaid), as in the above test G1, G2, and G3 are issued and expired gradually in this order. This means that it is not possible for G1 valid and G2 expired. From Table 1, we actually can see that only for case (G1 valid, G2 valid, G3 vaid), twe years, namely 2020 and 2025, are selected. This is because 2020 is the year when G3 is issued, while 2025 is the year when G1 is in service for the last year. If fact, these eight years selected include all such critical years for these three generations of PKIs, namely the years when each G1, G2 or G3 is issued, and the years when each G1, G2 or G3 is in service for the last year.

	2015					2016					
	G1 Client		G2 Client		G3 Client		G1 Client		G2 Client		G3 Client	
	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.
G1 Server	-	-	-	-	-	-	OK	OK	-	-	-	-
G2 Server	-	-	-	-	-	-	-	-	-	-	-	-
G3 Server	-	-	-	-	-	-	-	-	-	-	-	-

Table 2: Testing Results for 2015 and 2016

Each table from Table 2 Table 5 gives the testing rsults what a client and a server output for verifying the certificate chain of the other, for each case where during each of the eight years selected, both the server and the client can be any of G1, G2 or G3 server and client. Namely, these sever and clients hold the certificate chains issued by the respective root CA, together with the one-way link

certifates NewWithOld which link the current root CA public key to the previous one or two old root CA public keys. The verification results from the client and the server are listed in the colums of "Cli." and "Ser.", respectivley. "-" means that such a case actually doest not exist, while "OK" means that such a verificaiton of the other peer's certificate chain is possitive, and "No" means that such a verificaiton of the other peer's certificate chain is negative as one or more certificates in such a certificate chain are expired already.

As the first example, in Table 2, the "-" shown in the six cells of the fifth row under "2016" mean that in the year of 2016, when a G2 server interacts with a G1, G2 or G3 client, they cannot verify and accept each other's certificate chains, as such a G2 or G3 server, G2 or G3 client should not exist. As the second example, in Table 4, the "NO" shown in the six cells of the fourth row under "2026" mean that in the year of 2026, when a G1 server interacts with a G1, G2 or G3 client, they cannot verify and accept each other's certificate chains, as the root CA certificate of PKI G1 is expired already. As the third example, in Table 3, the "OK" shown in the six cells from the fourth row to the sixth row under "2020" mean that in the year of 2020, when any G1, G2 or G3 server interacts with any G1, G2 or G3 client, both the server and the client can verify and accept each other's certificate chains, which includes the one-way link certificates specified in this draft.

	2018					2020					
	G1 Client		G2 Client		G3 Client		G1 Client		G2 Client		G3 Client	
	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.
G1 Server	OK	OK	OK	OK	-	-	OK	OK	OK	OK	OK	OK
G2 Server	OK	OK	OK	OK	-	-	OK	OK	OK	OK	OK	OK
G3 Server	-	-	-	-	-	-	OK	OK	OK	OK	OK	OK

Table 3: Testing Results for 2018 and 2020

	2025			2026			
	G1 Client		G2 Client		G3 Client		
	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.
G1 Server	OK	OK	OK	OK	OK	OK	NO	NO	NO	NO	NO	NO
G2 Server	OK	OK	OK	OK	OK	OK	NO	NO	OK	OK	OK	OK
G3 Server	OK	OK	OK	OK	OK	OK	NO	NO	OK	OK	OK	OK

Table 4: Testing Results for 2025 and 2026

	2028			2030			
	G1 Client		G2 Client		G3 Client		
	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.	Cli.	Ser.
G1 Server	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
G2 Server	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
G3 Server	NO	NO	NO	NO	OK	OK	NO	NO	NO	NO	NO	NO

Table 5: Testing Results for 2028 and 2030

7. An Extension for Switching Root CA Certificate More Smoothly

There will be a time difference between when a device A sends its certificate chain and when another device B recives the certificate chain. This time difference is due to a number of reasons, including communnicaiton delay, operation time (especially for PQ cryptographic operations) needed at the receiving device B, the clock drift between devices A and B. In practice, the solution introduced in Section 5 requires that a new device SHOULD switch its link certificate chain to its new certificate chain around the expiry time of the old root CA certificate, as explained below.

* Case 1: A new decive A switches its certificate chain a little late. In this case, for a new receiving device B, it may be not able to verify this certificate chain as the old root certificate is expired when it recieved the certificate chain, though such a

connection is expected to be successful. In fact, if B could try to use the root CA cerificate verifying the certificate chain, the result SHALL be positive. However, such disturbance has been caused to new device B.

* Case 2: A new decive A switches its certificate chain a little early. In this case, for an old device B, it may receive a new ccertificate chain from A beore the expiry time of the old root CA certificate. Then, naturally, B will be not able to verify this certificate chain when B received the certificate chain, though such a connection is supposed to successful as old device B does not retire yet.

To address the above issue for switching the link certificate chain early or late, this section desribe an exension to the solution proposed in Section 5. The basic idea is to introduce a maximum time difference (MTD) T bewtween any two entities in a given domain when one entity sending a certificate chain and the other receiving it. Then, the extension works as follows.

* For any certificate issued by the old root CA certificate, its validity time MUST be at least 2T less than T3, the expected expiry of the old root CA certificate and the newWithOld link certificate.

* Each new device is supposed to complete its certificate chain switching on T3-T.

* When a new device A sends out its link certificate chain before T3-T, it will be recieved by a device B by T in B’s local time. At this moment, does not matter if B is a new or old old device, the old root CA certificate is still valid according to B’s local time. Therefore, B can verify A’s link certificate chain and the exension works.

* When a new device A sends out its new certificate chain after T3-T, it will be recieved by a device B after T3-2T in B’s local time. At this moment, if B is an old device, it retired already. If B is a new device, the new root CA certificate is valid alredy, so B can verify A’s new certificate chain. So, for either cases, the exension works.

8. Security Considerations

Security analysis will be given later.

9. Acknowledgments

To be added later.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC4210] Adams, C., Farrell, S., Kause, T., and T. Mononen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)", RFC 4210, DOI 10.17487/RFC4210, September 2005, <https://www.rfc-editor.org/info/rfc4210>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

11. Informative References

[I-D.D24] F. Driscoll, F., "Terminology for Post-Quantum Traditional Hybrid Schemes", Work in Progress, Internet-Draft,, February 2024, <https://datatracker.ietf.org/doc/draft-ietf-pquip-pqt-hybrid-terminology/>.

Authors' Addresses

Guilin Wang (editor)
Huawei Int. Pte Ltd
9 North Buona Vista Drive, #13-01
The Metropolis Tower 1
SINGAPORE 138588
Singapore
Email: wang.guilin@huawei.com

Yanjiang Yang
Huawei Int. Pte Ltd
Email: yang.yanjiang@huawei.com

Jie Zhang
Huawei Tech. Ltd
Email: zhangjie184@huawei.com