论文题目：

Some improvement of certificate revocation mechanism

作者：　　龚俭　　　东南大学计算机系，

　　　　　刘建航　　中兴通信股份有限公司。

通信地址：　　南京　东南大学计算机系　邮编 210096

联系电话：　　025－3794341

传真：　　　　025－3614842

e-mail：　　　jgong@njnet.edu.cn

Some improvement of certificate revocation mechanism

GONG Jian LIU Jianhang*
Computer Department, Southeast University
Nanjing 210096, China

【Abstract】The certificate revocation mechanism is the key to the reliability of PKI. Aiming at two important elements influencing the validity of CRL, this article put forward two methods: CRL Caching Service to reduce the global traffic caused by CRL verification; and CRL Grouping to cut down the size of CRL. Examples are given to explain how the improved CRL mechanism works.

【Keyword】PKI, X.509, Certificate, Certificate Revocation, CRL, CRL grouping.

1. Introduction

In 1978, Kohnfelder in MIT put forward the concept of certificate which is a signed data block containing the name of the user and his(her) public key. Protected by a signature, the certificate can be transported and stored in insecure network. Now, X.509 certificate has been applied extensively, and is becoming an important method to manage public key in open network environment.

Besides the digital certificate, many other entities are needed to complete the key management work, such as certificate authority(CA), register center(RA) , repository, etc. Every entity should maintain its own security parameter, ex., its key pair, the public key of CA and its security policy. These parameters consist of the end entity's personal security environment---PSE. Meanwhile, certificate revocation mechanism and reasonable certificate issuing policy are also needed. All the elements mentioned above consist of the public key infrastructure----PKI.

X.509 certificate is offline, the application needn't contact the issuer when using it to convey public key. However, in many cases a certificate needs to be revoked. For example, the name of the key pair owner has been changed, or the private key is compromised, etc. So that Certificate revocation list (CRL) mechanism is needed to revoke those certificates which are issued before but are no longer valid now. This function is the key to the reliability of PKI. Aiming at two important elements influencing the validity of CRL, this article put forward two methods: CRL Caching Service and CRL Grouping, so as to enhance the feasibility of CRL in large-scale

*The author is now with ZhongXing Telecom LTD.

network. Some examples are given to explain how the improved CRL mechanism works.

2. Certificate revocation mechanism in current PKI

According to the X.509 documents, CRL is issued periodically, which includes all the serial numbers of the certificates which are still in validity period but have been revoked. The verifier of a certificate can determine the validity of this certificate by checking whether it is included in the CRL, and this can be achieved in three steps. The first step for verifier is to get the CRL corresponding to the certificate. In this step, the verifier needs to know where the CRL is, and this can be realized by appending "CRLDistributionPoint" field to the certificate. The second step is to verify the validity of the CRL. Here the fact that one CRL is valid means not only that the digital signature on the CRL can be verified, but also that the CRL is related to the certificate being verified. If the certificate has been revoked, its serial number must be included in the CRL; and if there is no such serial number on the CRL, the certificate can be considered still valid. In general, CRL is related to issuer, that is, the revoked certificates which have same issuer will appear in the same CRL. The last step is that the verifier checks whether the certificate has been revoked.

PKI is complicated in large-scale network. Usually, it is not a single certificate but a certificate chain that is used to identify a key. To verify the certificates in the chain, the verifier needs to acquire all the CRLs for these certificates and check them one by one. If many CAs are hierarchically related, it will be burdensome to download and maintain CRL.

There is also an uncertain period from certificate revocation request to a new CRL being issued, because CRL is issued periodically, but the revocation requests arrive randomly. During this uncertain period, the status of the revoked certificates is inconsistent: all the end entities except the holder consider that the certificate is valid according to the latest CRL, but the holder think it's invalid because he has declared the disable of that certificate by requesting revocation to CA. If the end entities revoke their certificates frequently, the inconsistency brought by the delay between revocation request and the next CRL issuing will influence the quality of X.509 certificate service afforded by PKI. Tradeoff must be considered between the frequency of CRL announcement and the cost of the announcement.

Delay and scalability of CRL is the sticking point of validity of certificate revocation mechanism, and also influences the reliability of PKI. Some solutions have been proposed in PKI. For example, the CA in higher layer can issue CRLs for the CA in lower layer in order to reduce the number of CRLs. That is to say, the issuer of a certificate perhaps is not the same issuer of the CRL of this certificate. By this way, the number of CRLs in PKI can be decreased, but the length of each CRL will be increased, which may make the download or action of CRL be more difficult.

3. CRL Caching Service

CRL is issuer related. As the number of user increased, the workload of the issuer will also increased, which demands the issuer more processing capability and the network more bandwidth, and will bring about more and more global traffic in the Internet. Therefore, it may be better to set up CRL caching server(s) and provide CRL caching service to the users within the security domain(usually also a management domain). The main purpose of this service is to maintain a CRL cache repository, return the CRL corresponding to user's request, decide which CRL should be maintained according to the access frequency, and update the repository when needed. Generally speaking, CRL Caching Service maintains CRLs for local end entities, so that the directly access to the original issuer is reduced.

The provider of CRL Caching Service is called CRL Cache Server, which can be an independent entity or integrated into local CA/RA. The basic mechanism of this server can be illustrated by the following figure:

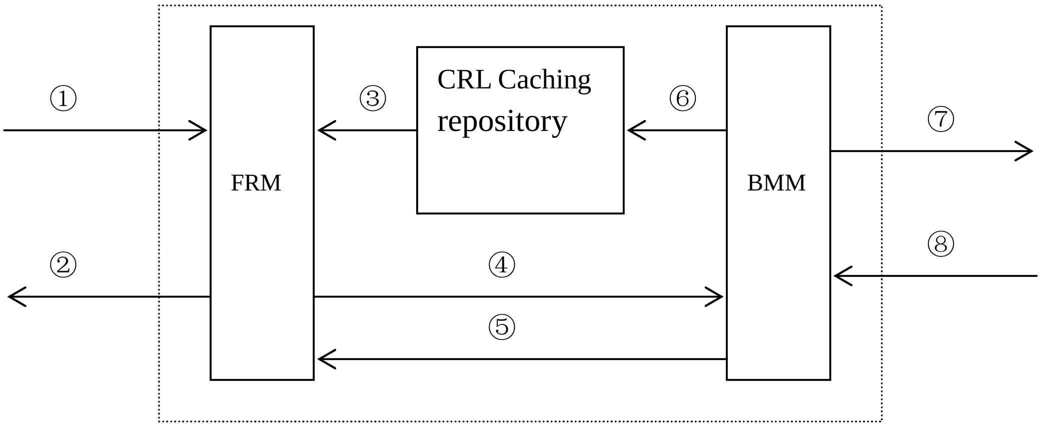

Fig. 1　CRL Caching Service Model

CRL Cache Server is composed by a foreground responding module(FRM) , a background-maintaining module(BMM) and a CRL cache repository. When FRM receives a request from user (①) , it searches the corresponding CRL in CRL cache repository. If a match is found, it returns the result to the user (②) and reset “RecentNonAccessPeriod” field in this item; otherwise it submits the request to BMM (④)and returns the result given by BMM (⑤). CRL caching repository includes the following fields:

IssuerName: X.500 name of the CA who issues this CRL;

DistributionPoint: URI which is used to identify the source of this CRL;

RecentNonAccessPeriod: record the elapsed time from the last request for this CRL;

BeValidTime: at which time this CRL becomes valid, equals to "ThisUpdate" in CRL;

InValidTime: at which time this CRL become invalid, equals to "NextUpdate" in CRL;

CRLContent: whole content of this CRL.

When BMM receives the user's request from FRM, it will download CRL from corresponding CA or directories which are specified by the "IssuerName" or "DistributionPoint" field in the request(⑦⑧), then returns the CRL to FRM and updates the CRL caching repository.

Besides responding to the CRL request submitted by FRM, the main work of BMM is to update the content of the CRL caching repository continuously. It scans every item circularly, then do the following works:

(1) Delete the item if its "RecentNonAccessedPeriod" exceeds the specified value;

(2) If the "InValidTime" in this item expired already, download the CRL again according to the "IssuerName" or "DistributionPoint" field in the item, then update three fields in that item: "CRLContent", "BeValidTime" and "InValidTime" ;

(3) Increase the "RecentNonAccessedPeriod" in every item.

The idea behind the CRL Caching Service is very simple: the server(s) maintains a local CRL cache. For the relevance of requests from one administration domain, CRL Caching Service will reduce network workload by decrease the CRL download times. By clearing the not frequently used items, CRL caching server needn't maintain all the CRLs. The server could monitor the status of every CRL in the repository on one's own initiative, and determine the time at which it will update the item. Most requests can get response immediately, because the CRL caching server has updated the CRL in the cache repository in advance.

4. CRL Grouping

The corresponded CRL is found by the verifier through CRLDistributionPoin of a certificate, but whether its content is still correct can not be known until CRL is completely downloaded to local again. In order to assure the correctness of the content, the verifier may have to download repeatedly the same content of a CRL. Therefore, a special mechanism is needed to tell the verifier if the CRL has been updated, and to decide if the completely download of the CRL is required, so as to improve the system efficiency. The mechanism suggested in this paper is called CRL grouping. The certificates can be grouped according to various principles, for example, the name space the user belong to or the purpose of the certificates, to create a new type CRL, called group CRL. Each certificate will be classified into one group, so that when this certificate is revoked, it will go to the corresponding CRL.

To group certificates, we suggest a extended field in X.509v3 certificate, called GroupCRL. The new field is defined as following:

```
#define struct {
        int                             GroupNo,
        distributionPointName           ThisGroupCRLDistributionPoint,
        distributionPointName           AllGroupCRLDistributionPoint,
        generalNames                         GroupCRLIssuer
} GroupCRL;
```

The CRL including the certificate extended with the "GroupCRL" field will be issued by "GroupCRLIssuer". Revoked certificates are grouped according to some predefined rules and put into various group CRLs. A group number will be assigned to each certificate issued. The certificate verifier can download the corresponded group CRL from the position "ThisGroupCRLDistributionPoint".

Grouping CRL can only cut down its size. To decrease the redundancy in downloading CRLs, the "AllGroupCRLDistributionPoint" in the "GoupCRL" field points to a group-brief-CRL. The "RevocationCertificates" field in this CRL is always empty, which means that this special CRL doesn't include any detailed information about revoked certificates. The main content in this CRL is an extended "GroupCRLInfo" field which includes the information about the entire group CRLs issued by the "GroupCRLIssuer". The extended field includes the following content:

```
#define struct   {
        struct {
                int                     GroupNo,
                time                    ThisUpdate,
                distributionPointName   ThisGroupCRLDistributionPoint
        } GoupCRL[0..MAX];
} GroupCRLInfo;
```

When the verifier of the certificate gets the group-brief-CRL, he can check whether he has the latest CRL version by means of comparing the "BeValidTime" in the group- brief-CRL and the group CRL. Because the number of groups is quite limited, the size of group-brief-CRL is much smaller than group CRLs' and is often invariable. The cost of downloading the group-brief-CRL is much lower than that of the group CRL. On the other hand, this group-brief-CRL can be issued frequently for its small size, so the end entity can be more conscious of the update of group CRL. As a result, the delay between the certification revocation request and CRL issuing can be shortened.

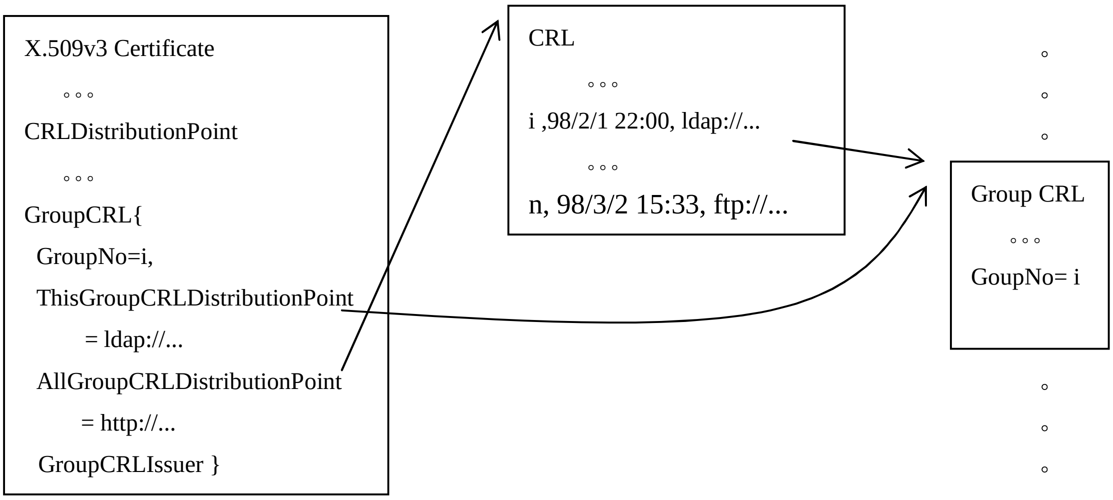

Fig. 2 The idea of CRL Grouping

By grouping CRL, the certificate user now can get the corresponded CRL by three ways:

- download the whole CRL according to the "CRLDistributionPoint" field;
- download the specified group CRL according to "GroupNo" and "ThisGroupCRLDistributionPoint" fields;
- fetch a short group-brief-CRL firstly and download the group CRL only when needed.

5. Improved CRL mechanism

When making use of CRL Cache Service and CRL Grouping, the procedure in which the end entity maintain and use CRL should be adjusted as following:

(1) After the end entity initializes, it should contact its CRL cache service provider, then download the group-brief-CRL and those group CRLs from the provider to form the PSE.

(2) The end entity scans the PSE periodically: download the group-brief-CRL from the CRL cache server when this CRL become invalid. For each CRL, check the group-brief-CRL to see whether the issue time of the group CRL differ from the one in PSE. If they are not the same, download that group CRL from the CRL cache server again and update the PSE.

(3) When the end entity need to validate a certificate (chain), it finds the position of the corresponded group CRL by checking the "GroupCRL" field. If that group CRL exists in PSE already, go to (6); otherwise, go to (4).

(4) Check the PSE, if the group-brief-CRL exists in it, go to (5); otherwise , download the group-brief-CRL from the CRL cache server and update the PSE.

(5) Download the corresponding group CRL from the CRL cache server and update the PSE.

(6) Check whether the group number in the certificate is the same as the one in the group CRL; check whether the certificate exists in the group CRL.

The following examples are going to show that how the CRL Caching Service and CRL Grouping can be used to enhance the validity mechanism of certificate revocation. Suppose that there is a CA hierarchy in education system of X province: the root CA:X.EDU is maintained by the administration center of X province; the secondary level CAs are maintained by six domain centers, ex., CA:XA.X.EDU; the bottom CAs are maintained by each university, which issue certificates to the students and teachers in their own school, ex., CA:XXU.X.EDU. The secondary level CA doesn't afford service for end entities, it only issues certificates for these bottom CAs in its domain. Suppose one secondary level CA works for 10 universities and each university have 4000 students and teachers in average. When teacher B visit library C, he will get various CRL which are issued by CA:X.EDU, CA:XA.X.EDU and CA:YYU.X.EDU (showed by the virtual line in figure 3) to check the whole certificate chain. With the CRL grouping, the teacher B may only need to check two CRL issued by CA:X.EDU and CA:YYU.X.EDU. If the ratio of certificate revocation is 3 percent, then the size of these two CRLs will be 2 (66 * 3%) and 120 (4000 * 3%).

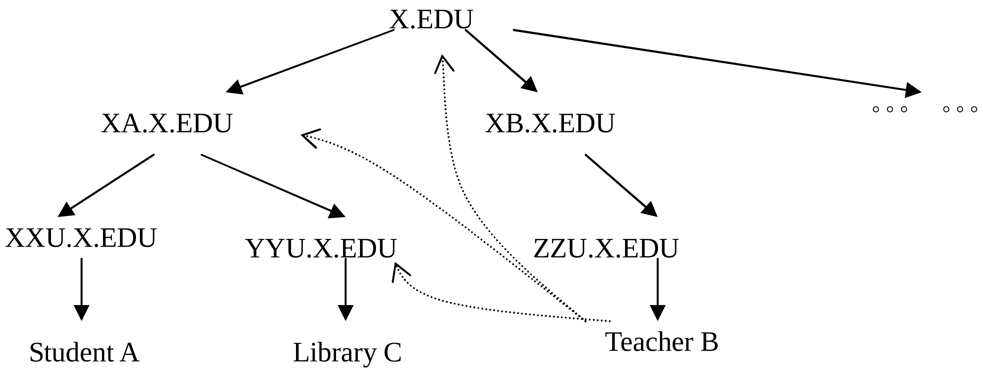

Fig. 3 A PKI example

When all the students and teachers in the XB domain begin to visit library C, the total length of CRLs to be download may increase to 4,880,000 (122 * 4000 * 10). By using CRL Caching Service, after the first time teacher B visits the library C, these two CRLs will be cached in the CRL caching server, so they can be download directly from the server when other user in XB domain make the same visit (see Fig.4). Therefore, the cross-boundary traffic will be reduced greatly. Especially when Library C become hot, this method will lessen its workload, and improve the service quality.

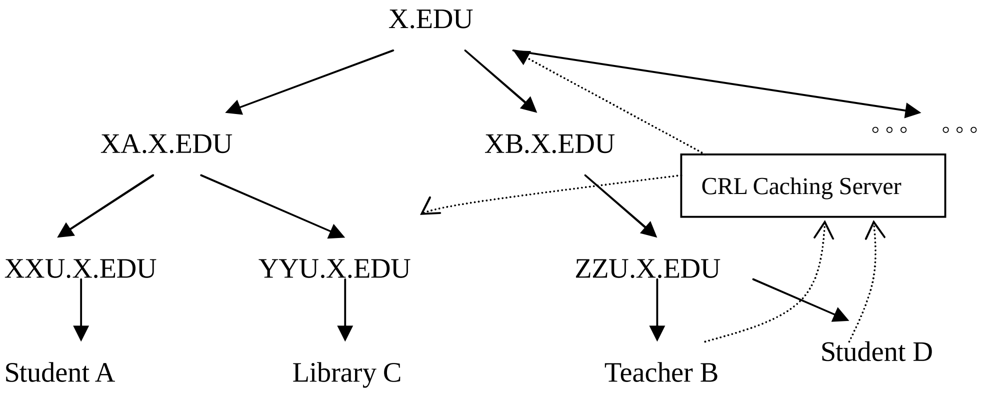

Fig. 4 An example of CRL caching service

Suppose Student A wants to visit Library C. If the scale of the XXU university become so large, for example 16000 students and teachers, the CRL issued by CA:XXU.X.EDU will have 480 items. Let the CRL be grouped into 10. For example the 1st group CRL will include all the revoked certificates belong to graduate students, and the 2nd group CRL will include the revoked certificates belong to all the teachers, now library C need only get the 1st group CRL whose size is much smaller.

Grouping is not only used to decrease the size the CRL. Besides those group CRLs, CA:XXU.X.EDU issues a group-brief-CRL to show the status of each group CRL. The size of this group-brief-CRL is only 10, so it can be issued more frequently, for example, one per hour, or even on demand. When C receives the request from A, it will fetch the group–brief-CRL from the CRL cache server maintained by the XXU University. If change is found, C will download the new group CRL, otherwise it will use the one at hand.

6. Conclusion

The reliability of PKI depends heavily on the certificate revocation mechanism, that is CRL today. Delay and scalability are two essential elements influencing the validity of CRL, especially in large-scale network environment with huge of users. The CRL mechanism has suffered long delay and become inefficient. With the CRL caching service and CRL grouping proposed in this paper, the global traffic can be reduced and workload of CRL issuer can distributed within the network. AT the same time, the synchronization between certificate revocation request and CRL issuing can also be improved because CRL caching server can inquire CRL issuer more frequently. All these advantages will make the application of certificate, e.g. distance learning, more feasible.

The GroupCRL defined in this paper is a non-critical extension to X.509v3 certificate. It will not violate the “Minimum Interoperability Specification for PKI Components, Version 1”suggested by NIST. As the certificate service getting more and more popular, the CRL mechanism should be more completed and efficient, just like domain name system today.

Reference

[1] IETF PKIX Working Group ,”draft-ietf-pkix-*.txt”, 1998

[2] Kohnfelder, Loren M., "Towards a Practical Public-key Cryptosystem", MIT S.B. Thesis, 1978.5

[3] WhitField Diffie and Martin Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, 1976.12

[4] Ronald L. Rivest and Bulter Lampson, “SDSI-A simple Distributed Security Infrastructure”, 1996.10

[5] Carl M. Ellson, “Certification Infrastructure Needs For Electronic Commerce And Personal Use”. 1997.7

[6] William Burrr, Donna Dodson, Noel nazario, W. Timothy Polk, “Minimum Interoperability Specification for PKI Components, Version 1”, NIST Paper, 1997.9